IBM WebSphere eXtreme Scale Version 7.1.1
Version 7.1

Guide d'administration

21 novembre 2011

Remarque
Certaines illustrations de ce manuel ne sont pas disponibles en français à la date d'édition.

décembre 2011

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- http://www.fr.ibm.com (serveur IBM en France)
- http://www.can.ibm.com (serveur IBM au Canada)
- http://www.ibm.com (serveur IBM aux Etats-Unis)

Compagnie IBM France
Direction Qualité
17, avenue de l'Europe
92275 Bois-Colombes Cedex

Table des matières

Figures

Tableaux

Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.

OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

France	Canada	Etats-Unis
(Pos1)		Home
Fin	Fin	End
(PgAr)		PgUp
(PgAv)		PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
(Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

Brevets

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

Assistance téléphonique

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.

A propos du *Guide d'administration*

La documentation de WebSphere eXtreme Scale inclut trois volumes qui fournissent les informations nécessaires pour utiliser, programmer et administrer le produit WebSphere eXtreme Scale.

Bibliothèque WebSphere eXtreme Scale

La bibliothèque WebSphere eXtreme Scale contient les documents suivants :

- La *Présentation du produit* contient une vue de haut niveau des concepts de WebSphere eXtreme Scale, avec des scénarios d'utilisation et des tutoriels.
- Le *Guide d'installation* explique comment installer les topologies communes de WebSphere eXtreme Scale.
- Le *Guide d'administration* contient les informations nécessaires pour les administrateurs système et explique notamment comment planifier les déploiements d'application, planifier la capacité, installer et configurer le produit, démarrer et arrêter des serveurs, surveiller l'environnement et le sécuriser.
- Le *Guide de programmation* contient des informations destinées aux développeurs d'applications, sur la manière de développer des applications pour WebSphere eXtreme Scale à l'aide des informations d'API incluses.

Pour télécharger les documents, accédez à la page de la bibliothèque de WebSphere eXtreme Scale.

Vous pouvez également accéder aux mêmes informations dans cette bibliothèque dans le centre de documentation de WebSphere eXtreme Scale version 7.1.1.

Utilisation hors ligne des manuels

Tous les manuels de la bibliothèque WebSphere eXtreme Scale contiennent des liens vers le centre de documentation, avec l'URL racine `http://publib.boulder.ibm.com/infocenter/wxsinfo/v7r1m1`. Ces liens vous permettent d'accéder directement aux informations associées. Toutefois, si vous travaillez hors ligne et rencontrez l'une de ces liens, vous pouvez rechercher le titre du lien dans les autres manuels dans la bibliothèque. La documentation d'API, le glossaire et les références des messages ne sont pas disponibles dans les manuels PDF.

A qui s'adresse ce document

Ce document est principalement destiné aux administrateurs système, administrateurs de la sécurité et opérateurs système.

Obtention des mises à jour de ce document

Vous pouvez obtenir les mises à jour de ce document en téléchargeant la version la plus récente à partir de la page de la bibliothèque de WebSphere eXtreme Scale.

Comment envoyer vos commentaires

Contactez l'équipe chargée de la documentation. Avez-vous trouvé ce que vous recherchiez ? Ces informations étaient-elles précises et complètes ? Envoyez vos commentaires sur cette documentation par courrier électronique, à l'adresse wasdoc@us.ibm.com.

Chapitre 1. Mise en route

Après avoir installé le produit, vous pouvez utiliser l'exemple de mise en route pour tester l'installation et utiliser le produit pour la première fois.

Tutoriel : Démarrer avec WebSphere eXtreme Scale

Après avoir installé WebSphere eXtreme Scale dans un environnement autonome, vous pouvez utiliser l'exemple d'application de démarrage qui présente clairement ses fonctions comme grille de données en mémoire.

Objectifs d'apprentissage

- Description des fichiers descripteur XML ObjectGrid de stratégie de déploiement et des fichiers descripteurs XML utilisés pour configurer l'environnement
- Démarrage des serveurs de catalogue et de conteneur à l'aide des fichiers de configuration
- En savoir plus sur le développement d'une application client
- Exécution de l'application client pour insérer des données dans la grille de données
- Surveillance des grilles de données à l'aide de la console Web

Durée

60 minutes

Leçon 1 du tutoriel d'initialisation : Définition de grilles de données avec des fichiers de configuration

Pour configurer des grilles de données simples, vous pouvez utiliser les fichiers `objectgrid.xml` et `deployment.xml` fournis dans l'exemple d'initialisation.

L'exemple utilise les fichiers `objectgrid.xml` et `deployment.xml` qui se trouvent dans le répertoire *racine_install_wxs*`/ObjectGrid/gettingstarted/xml`. Ces fichiers sont envoyés aux commandes de démarrage pour démarrer les serveurs de conteneur et un serveur de catalogue. Le fichier `objectgrid.xml` est le fichier XML descripteur d'ObjectGrid. Le fichier `deployment.xml` est le fichier XML descripteur de la stratégie de déploiement ObjectGrid. Ensemble, ces fichiers définissent une topologie répartie.

Fichier XML descripteur d'ObjectGrid

Un fichier XML de descripteur d'ObjectGrid permet de définir la structure de la grille d'objets utilisée par l'application. Il contient la liste des configurations de mappes de sauvegarde. Ces mappes de sauvegarde stockent les données en cache. L'exemple suivant présente un fichier d'exemple `objectgrid.xml`. Les premières lignes de ce fichier incluent l'en-tête requis de chaque fichier XML ObjectGrid. Cet exemple de fichier définit l'ObjectGrid `Grid` avec les mappes de sauvegarde `Map1` et `Map2`.

```
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
```

```
    <objectGrids>
        <objectGrid name="Grid">
            <backingMap name="Map1" />
            <backingMap name="Map2" />
        </objectGrid>
    </objectGrids>

</objectGridConfig>
```

Fichier XML du descripteur de la règle de déploiement

Un fichier XML de descripteur de stratégie de déploiement est transmis au serveur de conteneur lors du démarrage. La règle de déploiement doit être utilisée avec un fichier XML d'ObjectGrid et doit être compatible avec le fichier XML d'ObjectGrid qui lui est associé. Pour chaque élément objectgridDeployment dans la stratégie de déploiement, vous devez disposer d'un élément ObjectGrid correspondant dans le fichier XML d'ObjectGrid. Les éléments de la mappe de sauvegarde définis dans l'élément objectgridDeployment doivent être cohérents avec les mappes de sauvegarde contenues dans le fichier XML d'ObjectGrid. Chaque mappe de sauvegarde doit être référencée dans un seul et unique groupe de mappes.

Le fichier XML de descripteur de règle de déploiement est conçu pour être couplé avec le fichier XML d'ObjectGrid correspondant, le fichier `objectgrid.xml`. Dans l'exemple suivant, les premières lignes du fichier `deployment.xml` incluent l'en-tête requis de chaque fichier XML de règle de déploiement. Le fichier définit l'élément objectgridDeployment pour la grille ObjectGrid définie dans le fichier `objectgrid.xml`. Les mappes de sauvegarde Map1 et Map2 définies dans la grille ObjectGrid sont incluses dans le groupe de mappes mapSet pour lequel les attributs numberOfPartitions, minSyncReplicas et maxSyncReplicas sont configurés.

```
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy
 ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

    <objectgridDeployment objectgridName="Grid">
        <mapSet name="mapSet" numberOfPartitions="13" minSyncReplicas="0"
     maxSyncReplicas="1" >
            <map ref="Map1"/>
            <map ref="Map2"/>
        </mapSet>
    </objectgridDeployment>

</deploymentPolicy>
```

L'attribut numberOfPartitions de l'élément mapSet indique le nombre de partitions de l'élément mapSet. Il s'agit d'un attribut facultatif et sa valeur par défaut est 1. La valeur doit être adaptée à la capacité anticipée de la grille de données.

L'attribut minSyncReplicas de l'élément mapSet vise à indiquer le nombre minimal de fragments réplique synchrones de chaque partition du groupe de mappes. Il s'agit d'un attribut facultatif et sa valeur par défaut est égale à 0. Le fragment primaire et le fragment réplique ne sont pas positionnés tant que le domaine ne peut pas prendre en charge le nombre minimal de fragments réplique synchrones. Pour prendre en charge la valeur minSyncReplicas, vous avez besoin d'un nombre de conteneurs égal à la valeur de minSyncReplicas plus un. Si le nombre de fragments réplique synchrones est inférieur à la valeur de minSyncReplicas, les transactions d'écrire ne sont plus autorisées pour cette partition.

L'attribut maxSyncReplicas de l'élément mapSet vise à indiquer le nombre maximal de fragments réplique synchrones de chaque partition du groupe de mappes. Il s'agit d'un attribut facultatif et sa valeur par défaut est égale à 0. Aucune autre réplique synchrone n'est placée pour une partition une fois qu'un domaine a atteint ce nombre de fragments réplique synchrones pour cette partition spécifique. L'ajout de conteneurs prenant en charge cette grille d'objets peut entraîner un nombre croissant de fragments réplique synchrones si la valeur maxSyncReplicas n'a pas déjà été atteinte. L'exemple définit la valeur maxSyncReplicas sur 1, ce qui signifie que le domaine place au maximum une réplique synchrone. Si vous démarrez plusieurs instances de serveurs de conteneur, seule une réplique synchrone sera placée dans une des instances de serveurs de conteneur.

Point de contrôle de la leçon

Dans cette leçon, vous avez appris à :

- Définir des mappes qui stockent les données dans le fichier XML de descripteur d'ObjectGrid.
- Utiliser le fichier XML descripteur de déploiement pour définir le nombre de partitions et de répliques de la grille de données.

Leçon 2 du tutoriel d'initiation : Création d'une application client

Pour pouvoir insérer, supprimer, mettre à jour et extraire des données dans votre grille de données, vous devez écrire une application client. L'exemple d'initiation inclut une application client que vous pouvez utiliser pour en savoir plus sur la création de votre propre application client.

Le fichier `Client.java` dans le répertoire *racine_install_wxs*`/ObjectGrid/gettingstarted/client/src/` est le programme client qui montre comment se connecter à un serveur de catalogue, obtenir l'instance ObjectGrid et utiliser l'API ObjectMap. L'API ObjectMap stocke les données comme paires clé-valeur et elle est idéale pour la mise en cache d'objets qui n'ont aucune relation.

Si devez mettre en cache des objets qui ont des relations, utilisez l'API EntityManager.

1. Connectez-vous au service de catalogue en obtenant une instance ClientClusterContext.

   Pour établir la connexion au serveur de catalogues, utilisez la méthode connect de l'API ObjectGridManager. La méthode connect utilisée requiert seulement un noeud final de serveur de catalogue au format *`nom_hôte`*`:`*`port`*. Vous pouvez indiquer plusieurs noeuds finaux de serveur de catalogue en séparant les valeurs *`hostname`*`:`*`port`* par une virgule. Le fragment de code suivant montre comment se connecter à un serveur de catalogue et obtenir une instance ClientClusterContext :

   ```
   ClientClusterContext ccc = ObjectGridManagerFactory.getObjectGridManager().connect("localhost:2809", null, null);
   ```

   Si les connexions aux serveurs de catalogue aboutissent, la méthode connect retourne une instance ClientClusterContext. L'instance ClientClusterContext est requise pour obtenir l'ObjectGrid à partir de l'API ObjectGridManager.
2. Obtenez une instance ObjectGrid.

   Pour obtenir une instance ObjectGrid, utilisez la méthode getObjectGrid de l'API ObjectGridManager. La méthode getObjectGrid requiert l'instance ClientClusterContext et le nom de l'instance de grille de données. L'instance ClientClusterContext est obtenue pendant la connexion au serveur de

catalogue. Le nom de l'instance ObjectGrid est `Grid` ; ce nom est spécifié dans le fichier `objectgrid.xml`. Le fragment de code suivant montre comment obtenir la grille de données en appelant la méthode getObjectGrid de l'API ObjectGridManager.

```
ObjectGrid grid = ObjectGridManagerFactory.getObjectGridManager().getObjectGrid(ccc, "Grid");
```

3. Obtenez une instance Session.

   Vous pouvez obtenir une session de l'instance ObjectGrid obtenue. Une instance Session est indispensable pour obtenir l'instance ObjectMap et pour effectuer une démarcation de transaction. Le fragment de code suivant montre comment obtenir une instance Session en appelant la méthode getSession de l'API ObjectGrid.

   ```
   Session sess = grid.getSession();
   ```

4. Obtenez une instance ObjectMap.

   Après avoir obtenu une instance Session, vous pouvez obtenir une instance ObjectMap depuis une instance Session en appelant la méthode getMap de l'API Session. Vous devez transmettre le nom de la mappe comme paramètre à la méthode getMap pour obtenir l'instance ObjectMap. Le fragment de code suivant montre comment obtenir ObjectMap en appelant la méthode getMap de l'API Session.

   ```
   ObjectMap map1 = sess.getMap("Map1");
   ```

5. Utilisez les méthodes ObjectMap.

   Une fois une instance ObjectMap obtenue, vous pouvez utiliser l'API ObjectMap. N'oubliez pas que l'interface ObjectMap est une mappe transactionnelle et qu'elle requiert une démarcation de transaction à l'aide des méthodes begin et commit de l'API Session. Faute de démarcation de transaction explicite, les opérations ObjectMap s'exécutent avec des transactions de validation automatique.

   Le fragment de code suivant montre comment utiliser l'API ObjectMap avec une transaction de validation automatique.

   ```
   map1.insert(key1, value1);
   ```

   Le fragment de code suivant montre comment utiliser l'API ObjectMap avec une démarcation de transaction explicite.

   ```
   sess.begin();
   map1.insert(key1, value1);
   sess.commit();
   ```

Point de contrôle de la leçon

Dans cette leçon, vous avez appris à créer une application client simple pour effectuer des opérations de grille de données.

Leçon 3 du tutoriel d'initiation 3 : Exécution de l'exemple d'application client démarrée

Utilisez les étapes suivantes pour démarrer votre première grille de données et exécuter un client en vue d'interagir avec la grille de données.

le script `env.sh|bat` est appelé par les autres scripts pour la définition de variables d'environnement requises. Il n'est normalement pas nécessaire de modifier ce script.

- UNIX Linux `./env.sh`
- Windows `env.bat`

Pour exécuter l'application, vous devez d'abord démarrer le processus de service de catalogue. Le service de catalogue est le centre de contrôle de la grille de données. Il conserve la trace des emplacements de serveurs de conteneur et contrôle le placement des données sur les serveurs de conteneur hôtes. Une fois que le service de catalogue démarre, vous pouvez démarrer les serveurs de conteneur qui stockent les données d'application de la grille de données. Pour stocker plusieurs copies des données, vous pouvez démarrer plusieurs serveurs de conteneur. Lorsque tous les serveurs sont démarrés, vous pouvez exécuter l'application client pour insérer, mettre à jour, supprimer et extraire des données de la grille de données.

1. Ouvrez une session terminal ou une fenêtre de ligne de commande.
2. La commande suivante permet d'accéder au répertoire `gettingstarted` :

   `cd` *`racine_install_wxs`*`/ObjectGrid/gettingstarted`

   Remplacez *racine_install_wxs* par le chemin d'accès au répertoire racine d'installation d'eXtreme Scale ou le chemin d'accès de l'évaluation eXtreme Scale extraite *racine_install_wxs*.
3. Exécutez le script suivant pour démarrer un processus de service de catalogue sur le système hôte local :

   - UNIX Linux `./runcat.sh`
   - Windows `runcat.bat`

   Le processus du service de catalogue s'exécute dans la fenêtre du terminal en cours.

   Vous pouvez également démarrer le service de catalogue avec la commande **`startOgServer`**. Exécutez **`startOgServer`** depuis le répertoire *racine_install_wxs*`/ObjectGrid/bin` :

   - UNIX Linux `startOgServer.sh cs0 -catalogServiceEndPoints cs0:localhost:6600:6601 -listenerPort 2809`
   - Windows `startOgServer.bat cs0 -catalogServiceEndPoints cs0:localhost:6600:6601 -listenerPort 2809`
4. Ouvrez une autre session terminal ou fenêtre de ligne de commande et exécutez la commande suivante pour démarrer une instance de serveur de conteneur :

   - UNIX Linux `./runcontainer.sh server0`
   - Windows `runcontainer.bat server0`

   Le serveur de conteneur s'exécute dans la fenêtre du terminal en cours. Vous pouvez répéter cette étape avec un nom de serveur différent si vous voulez démarrer plus d'instances de serveurs de conteneur pour prendre en charge la réplication.

   Vous pouvez également démarrer les serveurs de conteneur avec la commande **`startOgServer`**. Exécutez **`startOgServer`** depuis le répertoire *racine_install_wxs*`/ObjectGrid/bin` :

   - UNIX Linux `startOgServer.sh c0 -catalogServiceEndPoints localhost:2809 -objectgridFile gettingstarted\xml\objectgrid.xml -deploymentPolicyFile gettingstarted\xml\deployment.xml`
   - Windows `startOgServer.bat c0 -catalogServiceEndPoints localhost:2809 -objectgridFile gettingstarted\xml\objectgrid.xml -deploymentPolicyFile gettingstarted\xml\deployment.xml`
5. Ouvrez une autre session terminal ou fenêtre de ligne de commande pour exécuter le client.

Le script `runclient.sh|bat` exécute le client CRUD et démarre l'opération voulue. Le script `runclient.sh|bat` est exécuté avec les paramètres suivants :

- UNIX Linux `./runclient.sh commande valeur1 valeur2`
- Windows `runclient.bat command value1 value2`

Pour *commande*, utilisez l'une des options suivantes :

- Définissez `i` pour insérer *value2* dans la grille de données avec la clé *value1*
- Spécifiez `u` pour mettre à jour l'objet indexé par *valeur1* avec *value2*
- Spécifiez `d` pour supprimer l'objet indexé par *valeur1*
- Spécifiez `g` pour extraire et afficher l'objet indexé par *valeur1*

a. Ajoutez des données à la grille de données :
   - UNIX Linux `./runclient.sh i key1 helloWorld`
   - Windows `runclient.bat i key1 helloWorld`

b. Recherchez et affichez la valeur :
   - UNIX Linux `./runclient.sh g key1`
   - Windows `runclient.bat g key1`

c. Mettez la valeur à jour :
   - UNIX Linux `./runclient.sh u key1 goodbyeWorld`
   - Windows `runclient.bat u key1 goodbyeWorld`

d. Supprimez la valeur :
   - UNIX Linux `./runclient.sh d key1`
   - Windows `runclient.bat d key1`

Point de contrôle de la leçon

Dans cette leçon, vous avez appris à :

- Démarrer les serveurs de catalogue et les serveurs de conteneur
- Exécuter l'exemple d'application client

Leçon 4 du tutoriel du guide de démarrage : Surveillance de l'environnement

Vous pouvez utiliser l'utilitaire **xscmd** et les outils de la console Web pour surveiller votre environnement de grille de données.

Surveillance à l'aide de la console Web

Avec la console Web, vous pouvez générer des graphiques des statistiques actuelles et historiques. Cette console fournit un certain nombre de graphiques préconfigurés pour des présentations générales et elle comporte une page de rapports personnalisés que vous pouvez utiliser pour élaborer des graphiques à partir des statistiques disponibles. Les fonctionnalités graphiques de la console de surveillance de WebSphere eXtreme Scale permettent de visualiser les performances globales des grilles des données présentes dans votre environnement.

Installez la console Web comme fonction facultative lorsque vous exécutez l'assistant d'installation.

1. Démarrez le serveur de la console. Le script **startConsoleServer.bat|sh** de démarrage du serveur de la console se trouve dans le répertoire *racine_install_wxs*`/ObjectGrid/bin` de votre installation.

2. Connectez-vous à la console.
   a. Dans votre navigateur Web, accédez à `https://your.console.host:7443`, en remplaçant `your.console.host` par le nom de l'hôte du serveur sur lequel vous avez installé la console.
   b. Connectez-vous à la console.
      - **ID utilisateur :** `admin`
      - **Mot de passe :** `admin`

      La page d'accueil de la console s'affiche.
3. Modifiez la configuration de la console. Cliquez sur **Paramètres** > **Configuration** pour afficher la configuration de la console. La configuration de la console comprend ce type d'informations :
   - la chaîne de trace pour le client WebSphere eXtreme Scale, comme `*=all=disabled`
   - le nom et le mot de passe de l'administrateur
   - son adresse e-mail
4. Créez et maintenez des connexions aux serveurs de catalogue que vous voulez surveiller. Répétez les étapes suivantes pour ajouter chaque serveur de catalogue à la configuration.
   a. Cliquez sur **Paramètres** > **Serveurs de catalogue eXtreme Scale**.
   b. Ajoutez un nouveau serveur de catalogue.

      1) Cliquez sur l'icône Ajouter ( ) pour enregistrer un serveur de catalogue existant.
      2) Fournissez des informations, telles que le nom d'hôte et le port d'écoute. Voir «Planification des ports réseau», à la page 64 pour plus d'informations sur la configuration des ports et les valeurs par défaut.
      3) Cliquez sur **OK**.
      4) Vérifiez que le serveur de catalogue a bien été ajouté à l'arborescence de navigation.
5. Visualisez le statut de la connexion La zone **Domaine en cours** indique le nom du domaine de services de catalogue qui est actuellement utilisé pour afficher des informations dans la console Web. L'état de la connexion s'affiche en regard du nom du domaine de services de catalogue.
6. Affichez les statistiques des grilles de données et des serveurs ou créez un rapport personnalisé.

Surveillance avec l'utilitaire xscmd

1. Ouvrez une fenêtre de ligne de commande. Sur la ligne de commande, définissez les variables d'environnement appropriées.
   a. Définissez la variable d'environnement *CLIENT_AUTH_LIB* :
      - Windows `set CLIENT_AUTH_LIB=<path_to_security_JAR_or_classes>`
      - UNIX `set CLIENT_AUTH_LIB=<path_to_security_JAR_or_classes>`
        `export CLIENT_AUTH_LIB`
2. Accédez au répertoire *rép_base_wxs*`/bin`.

   `cd rép_base_wxs/bin`
3. Exécutez plusieurs commandes pour afficher des informations sur votre environnement.

- Afficher tous les serveurs de conteneur en ligne pour la grille de données de la grille et le groupe de mappes mapSet :

  ```
  xscmd -c showPlacement -g Grid -ms mapSet
  ```

- Afficher les informations de routage de la grille de données.

  ```
  xscmd -c routetable -g Grid
  ```

- Afficher le nombre d'entrées de mappe dans la grille de données.

  ```
  xscmd -c showMapSizes -g Grid -ms mapSet
  ```

Arrêt des serveurs

Une fois que vous avez fini d'utiliser l'application client et de surveiller l'exemple d'environnement du guide de démarrage, vous pouvez arrêter les serveurs.

- Si vous avez utilisé les fichiers script pour démarrer les serveurs, utilisez <ctrl+c> pour arrêter le processus de service de catalogue et les serveurs de conteneur dans les fenêtres correspondantes.
- Si vous avez utilisé la commande **startOgServer** pour démarrer vos serveurs, utilisez la commande **stopOgServer** pour les arrêter.

  Arrêtez le serveur de conteneur :

  - UNIX Linux `stopOgServer.sh c0 -catalogServiceEndPoints localhost:2809`
  - Windows `stopOgServer.bat c0 -catalogServiceEndPoints localhost:2809`

  Arrêtez le serveur de conteneur :

  - UNIX Linux `stopOgServer.sh cs1 -catalogServiceEndPoints localhost:2809`
  - Windows `stopOgServer.bat cs1 -catalogServiceEndPoints localhost:2809`

Point de contrôle de la leçon

Dans cette leçon, vous avez appris à :

- démarrer la console Web et la connecter au serveur de catalogue ;
- surveiller les statistiques de la grille et des serveurs ;
- arrêter les serveurs.

Chapitre 2. Planification

Avant d'installer WebSphere eXtreme Scale et de déployer vos applications de grille de données, vous devez choisir votre topologie de mise en cache, planifier la capacité, vérifier les configurations matérielle et logicielle requises et les paramètres de réseau et d'optimisation, etc. Vous pouvez également utiliser la liste de contrôle opérationnelle pour vérifier que votre environnement est prêt pour le déploiement d'applications.

Vous trouverez une discussion des pratiques recommandées pour la conception d'applications WebSphere eXtreme Scale dans l'article suivant de developerWorks : Principles and best practices for building high performing and highly resilient WebSphere eXtreme Scale applications.

Présentation de la planification

Avant de commencer à utiliser WebSphere eXtreme Scale dans un environnement de production, les points suivants sont à prendre en considération afin d'optimiser le déploiement.

Considérations relatives à l'installation

Vous pouvez installer WebSphere eXtreme Scale dans un environnement autonome, ou vous pouvez intégrer l'installation à WebSphere Application Server. Pour pouvoir mettre à niveau de manière transparente les serveurs, vous devez planifier l'environnement en conséquence. Pour optimiser les performances, les serveurs de catalogue doivent s'exécuter sur des machines différentes de celles des serveurs de conteneur. Si vous devez exécuter vos serveurs de catalogue et serveurs de conteneurs sur la même machine, utilisez des installations distinctes de WebSphere eXtreme Scale pour les serveurs de catalogue et les serveurs de conteneur. En utilisant deux installations, vous pouvez mettre à niveau l'installation qui exécute le serveur de catalogue en premier. Voir les

Remarques relatives à la topologie de mise en cache

Votre architecture peut utiliser la mise en cache des données locales en mémoire ou la mise en cache des données client-serveur réparties. Chaque type de topologie du cache présente des avantages et des inconvénients. La topologie de mise en cache que vous implémentez dépend de la configuration de votre environnement et de l'application. Pour plus d'informations sur les différentes topologies de cache, voir «Planification de la topologie», à la page 10.

Remarques sur la capacité de données

Points à prendre en considération :

- **Nombre de systèmes et de processeurs** : combien de machines et de processeurs physiques sont nécessaires dans l'environnement ?
- **Nombre de serveurs** : combien de serveurs eXtreme Scale pour héberger les mappes eXtreme Scale ?
- **Nombre de partitions** : la quantité de données stockées dans les mappes est l'un des facteurs déterminant le nombre de partitions nécessaires.

- **Nombre de répliques** : combien de répliques sont requises pour chacun des fragments primaires du domaine ?
- **Réplication synchrone ou asynchrone** : les données sont-elles si vitales pour nécessiter une réplication synchrone ? Ou bien, est-ce que les performances sont une priorité plus importante ? Dans ce cas, la réplication asynchrone s'impose
- **Tailles de pile** : quel volume sera stocké sur chaque serveur ?

Pour une discussion détaillée de chacune de ces considérations, voir les «Planification de la capacité de l'environnement», à la page 55..

Planification de la topologie

Avec WebSphere eXtreme Scale, l'architecture de votre système peut utiliser la mise en cache des données locales en mémoire ou la mise en cache des données client-serveur réparties. L'architecture peut avoir des relations différentes avec vos bases de données. Vous pouvez également configurer la topologie pour l'étendre à plusieurs centres de données.

WebSphere eXtreme Scale requiert une infrastructure supplémentaire minimale pour pouvoir fonctionner. Cette infrastructure consiste en des scripts permettant d'installer, de démarrer et d'arrêter une application Java Platform, Enterprise Edition sur un serveur. Les données mises en cache sont stockées dans les serveurs de conteneur et les clients se connectent à distance au serveur.

Environnements internes

Lors du déploiement dans un environnement interne, WebSphere eXtreme Scale s'exécute dans une seule machine virtuelle Java et il n'est pas répliqué. Pour configurer un environnement local, vous pouvez utiliser un fichier XML ObjectGrid ou les API ObjectGrid.

Environnement réparti

Lorsque effectuez le déploiement dans un environnement réparti, WebSphere eXtreme Scale s'exécute dans un ensemble de machines virtuelles Java, ce qui améliore les performances, la disponibilité et l'évolutivité. Dans cette configuration, vous pouvez utiliser les fonctions de réplication et de partitionnement des données. Vous pouvez également ajouter d'autres serveurs sans redémarrer les serveurs eXtreme Scale existants. Comme dans le cas d'un environnement local, un fichier XML ObjectGrid ou une configuration par programmation équivalente est nécessaire dans un environnement réparti. Vous devez également fournir un fichier XML de stratégie de déploiement contenant les détails de la configuration.

Il est possible de créer des déploiements simples ou des déploiements plus vastes se chiffrant en téraoctets et comptant plusieurs milliers de serveurs.

Cache interne local

Dans le cas le plus simple, WebSphere eXtreme Scale peut être utilisé comme cache de grille de données locale (non répartie) en mémoire. Cette mise en cache locale peut s'avérer particulièrement utile pour les applications au nombre d'accès simultanés élevé où plusieurs unités d'exécution doivent accéder aux données temporaires et les modifier. Les données conservées dans une grille de données locale peuvent être indexées et extraites à l'aide de requêtes. Les requêtes

permettent d'utiliser des jeux de données volumineux en mémoire. Le support fourni avec machine virtuelle Java (JVM), qui est prêt à être utilisé, dispose d'une structure de données limitées.

La topologie de cache local en mémoire de WebSphere eXtreme Scale permet d'octroyer un accès cohérent et transactionnel aux données temporaires dans une machine virtuelle Java unique.

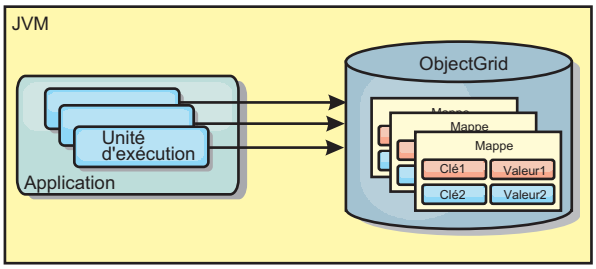


Figure 1. Scénario de cache local en mémoire

Avantages

- Configuration simple : une ObjectGrid peut être créée à l'aide d'un programme ou de manière déclarative avec le fichier XML du descripteur de déploiement ObjectGrid ou à l'aide d'une autre structure telle que Spring.
- Rapide : chaque mappe de sauvegarde peut être ajustée de façon indépendante pour optimiser l'utilisation de la mémoire et des accès simultanés.
- Configuration idéale pour les topologies de machine virtuelle Java dotées de petits jeux de données ou pour la mise en cache de données fréquemment consultées.
- Transactionnelle. Les mises à jour de mappe de sauvegarde peuvent être regroupées dans la même unité d'oeuvre et peuvent être intégrées en dernier lieu aux transactions constituées de deux phases telles que les transactions JTA (Java Transaction Architecture).

Inconvénients

- Aucune tolérance de panne.
- Les données ne sont pas répliquées. Les mémoires cache internes se prêtent aux données de référence en lecture seule.
- Non évolutive. La quantité de mémoire requise par la base de données peut dépasser la capacité de la machine virtuelle Java.
- Problèmes survenant lors de l'ajout de machines virtuelles Java :
  - Les données ne peuvent pas être facilement partitionnées ;
  - Nécessité de répliquer manuellement l'état entre les machines virtuelles Java ou chaque instance de cache peut présenter différentes versions des mêmes données.
  - L'invalidation est coûteuse.
  - Chaque cache doit être préchauffé indépendamment. Le préchauffage est la période de chargement d'un jeu de données permettant de remplir le cache avec des données valides.

Utilisation

La topologie de déploiement de la mémoire cache interne locale ne doit être utilisée que lorsque la quantité de données à mettre en cache est limitée (peut être abritée par une seule machine virtuelle Java) et est relativement stable. Cette approche doit tolérer les données obsolètes. L'utilisation d'expulseurs pour conserver les données les plus fréquemment ou récemment utilisées dans le cache peut contribuer à réduire la taille du cache et à accroître la pertinence des données.

Cache local répliqué sur des homologues

Vous devez vous assurer que le cache est synchronisé si plusieurs processus avec des instances indépendantes de cache existent. Pour vérifier que les instances de cache sont synchronisées, activez un cache répliqué sur des homologues avec JMS (Java Message Service).

WebSphere eXtreme Scale comprend deux plug-in qui propagent automatiquement les modifications de transactions entre les instances ObjectGrid homologues. Le plug-in JMSObjectGridEventListener propage automatiquement les modifications eXtreme Scale à l'aide de JMS.

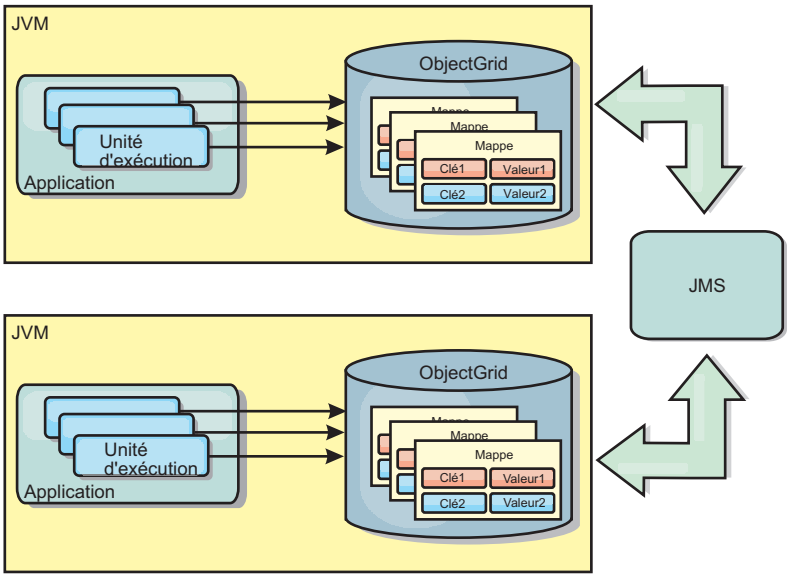


Figure 2. Cache répliqué sur des homologues avec des modifications qui sont propagées à l'aide de JMS

Si vous exécutez un environnement WebSphere Application Server, le plug-in TranPropListener est aussi disponible. Il utilise le gestion HA (high availability) pour propager les modifications à chaque instance de cache homologue.

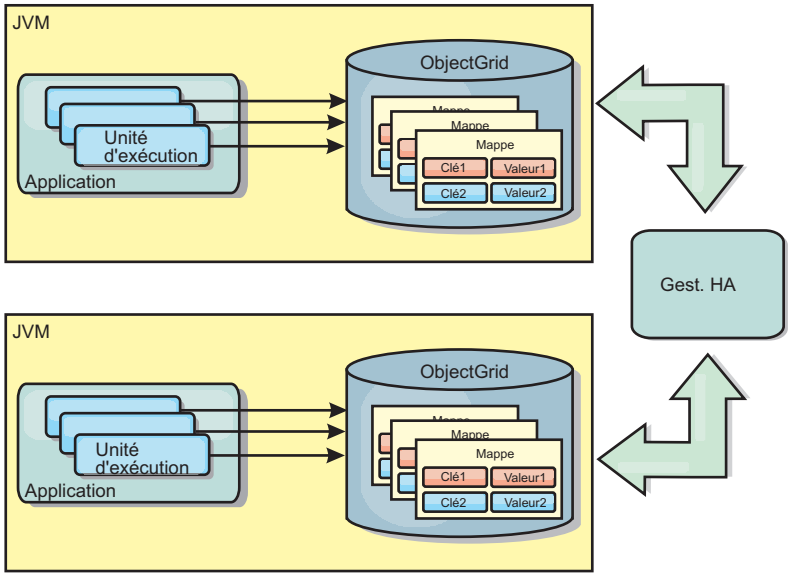


Figure 3. Cache répliqué sur des homologues avec des modifications qui sont propagées à l'aide du gestionnaire de haute disponibilité

Avantages

- Plus grande validité des données car celles-ci sont actualisées plus souvent.
- Avec le plug-in TranPropListener, tout comme avec l'environnement local, il est possible de créer la grille de données eXtreme Scale par programmation ou de manière déclarative avec le fichier XML du descripteur de déploiement d'eXtreme Scale ou avec d'autres structures de travail comme Spring. L'intégration au gestionnaire de haute disponibilité s'effectue automatiquement.
- Chaque mappe de sauvegarde peut être optimisée indépendamment en termes d'utilisation de la mémoire et de simultanéité des accès.
- Il est possible de regrouper en une seule unité d'oeuvre les mises à jour des mappes de sauvegarde qui peuvent être intégrées comme derniers participants de transactions en deux phases comme le sont les transactions Java Transaction Architecture (JTA).
- Idéal pour les topologies comprenant un nombre restreint de machines virtuelles Java avec un dataset de taille raisonnablement réduite ou pour la mise en cache des données à accès fréquent.
- Les modifications de la grille de données eXtreme Scale sont répliquées à toutes les instances eXtreme Scale homologues. Les modifications sont cohérentes tant qu'un abonnement durable est utilisé.

Inconvénients

- La configuration et la maintenance du plug-in JMSObjectGridEventListener peut s'avérer une tâche complexe. Il est possible de créer la grille de données eXtreme Scale par programmation ou de manière déclarative avec le fichier XML du descripteur de déploiement d'eXtreme Scale ou avec d'autres structures de travail comme Spring.
- Pas d'extensibilité : la quantité de mémoire requise par la base de données risque de submerger la machine virtuelle Java.
- Fonctionne de manière incorrecte lorsqu'on ajoute des machines virtuelles Java :
  - les données ne sont pas facilement partitionnées
  - l'invalidation est onéreuse

– chaque cache doit être prérempli de manière indépendante

Quand l'utiliser

Utilisez la topologie de déploiement uniquement lorsque la quantité de données à mettre en cache est faible, peut tenir sur une seule machine virtuelle Java, et relativement stable.

Cache imbriqué

Les grilles WebSphere eXtreme Scale peuvent s'exécuter dans des processus existants, tels que des serveurs eXtreme Scale intégrés ou vous pouvez les gérer comme des processus externes.

Les grilles imbriquées sont utiles lorsque l'exécution se fait dans un serveur d'applications tel que WebSphere Application Server. Vous pouvez démarrer les serveurs eXtreme Scale non imbriqués à l'aide de scripts de ligne de commande et les exécuter dans un processus Java.

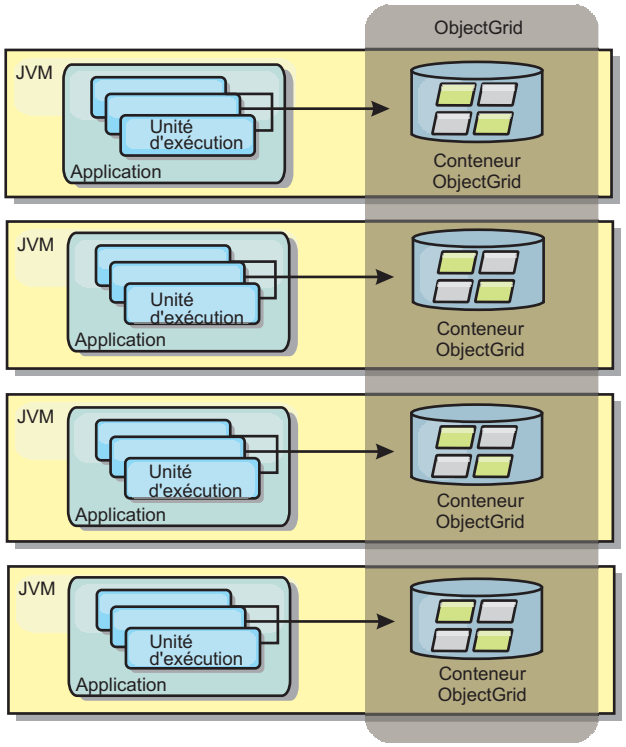


Figure 4. Cache imbriqué

Avantages

- simplification de l'administration en raison du nombre inférieur de processus à gérer
- simplification du déploiement d'application car la grille utilise le chargeur de classe de l'application client
- prise en charge du partitionnement et de la haute disponibilité.

Inconvénients

- augmentation de l'encombrement mémoire dans le processus client car toutes les données sont regroupées dans le processus
- augmentation de l'utilisation de l'unité centrale en vue de la gestion des demandes des clients
- plus grande difficulté à gérer les mises à niveau des applications car les clients utilisent les mêmes fichiers d'archive Java que les serveurs
- moindre flexibilité. Les clients et les serveurs de grille ne peuvent évoluer au même rythme. Lorsque des serveurs sont définis en externe, la gestion du nombre de processus devient plus flexible

Utilisation

Utilisez les grilles imbriquées lorsqu'une grande quantité de mémoire est disponible dans le processus client pour les données de la grille et pour les données de basculement.

Plus d'informations, voir la rubrique relative à l'activation du mécanisme d'invalidation de client dans *Guide d'administration*.

Cache réparti

La plupart du temps, WebSphere eXtreme Scale est utilisé en tant que cache partagé permettant un accès transactionnel aux données de plusieurs composants là où une base de données classique aurait été nécessaire. Avec le cache partagé, il n'est plus nécessaire de configurer une base de données.

Cohérence de la mémoire cache

Le cache est cohérent car tous les clients y voient les mêmes données. Chaque donnée est stockée dans le cache sur un seul serveur ce qui permet d'éviter la coexistence de plusieurs copies d'enregistrements risquant de contenir des versions différentes des données. Un cache cohérent contient un nombre croissant de données au fur et à mesure que l'on ajoute des serveurs à la grille et le cache évolue de manière linéaire au fur et à mesure que la taille de la grille augmente. Comme les clients accèdent aux données de cette grille de données avec des appels de procédure distante, cette mémoire est également appelée cache distant ou éloigné. Grâce au partitionnement des données, chaque processus contient un sous-ensemble unique de données. Les grandes grilles peuvent contenir davantage de données et traiter plus de demandes pour ces données. Par ailleurs la cohérence évite d'avoir à envoyer les données d'invalidation autour de la grille de données, car aucune donnée périmée n'existe. Le cache cohérent contient uniquement la copie la plus récente de chaque donnée.

Si vous exécutez un environnement WebSphere Application Server, le plug-in TranPropListener est aussi disponible. Il utilise le composant de haute disponibilité (gestionnaire HA) de WebSphere Application Server pour propager les modifications à chaque instance de cache ObjectGrid homologue.

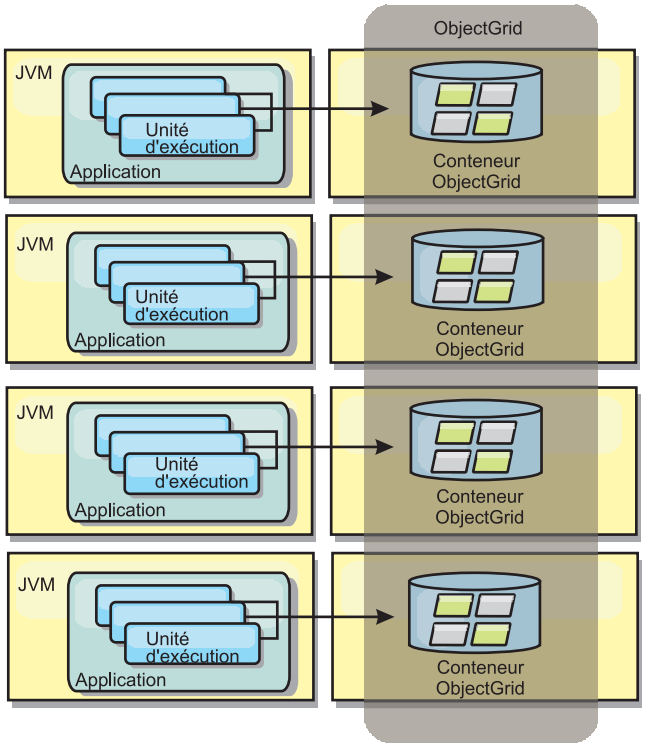


Figure 5. Cache réparti

Cache local

Lorsqu'eXtreme Scale est utilisé dans le cadre d'une topologie répartie, les clients peuvent éventuellement disposer d'un cache local en ligne. L'on appelle cache local ce cache facultatif. Il s'agit d'un ObjectGrid indépendant, présent sur chaque client et faisant office de cache du cache distant côté serveur. Il est activé par défaut lorsque le verrouillage est configuré sur OPTIMISTIC ou sur NONE. Son utilisation est impossible lorsque le verrouillage est configuré sur PESSIMISTIC.

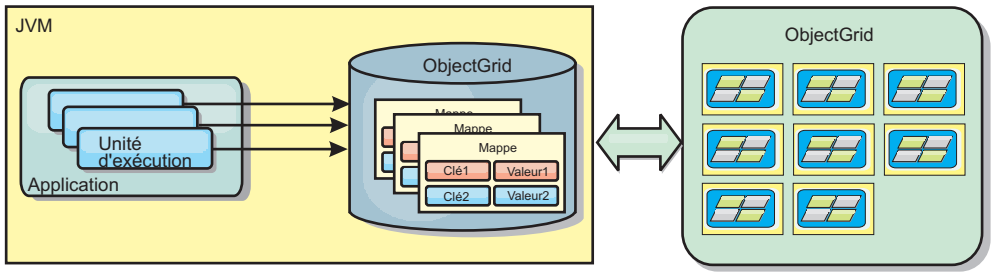


Figure 6. Cache local

Le cache local est très rapide car il offre un accès en mémoire à un sous-ensemble des données stockées à distance sur les serveurs eXtreme Scale. Il n'est pas partitionné et contient des données provenant de n'importe quelle partition eXtreme Scale distante. Jusqu'à trois groupes de caches peuvent exister dans WebSphere eXtreme Scale :

1. Le cache du groupe des transactions contient toutes les modifications apportées à une même transaction. Il contient une copie de travail des données jusqu'à ce que la transaction soit validée. Lorsqu'une transaction client demande des données à une ObjectMap, la transaction est vérifiée en priorité.

2. Le cache local du groupe des clients contient un sous-ensemble des données du groupe des serveurs. Lorsque le groupe des transactions ne contient pas les données, les données sont extraites du niveau client, si elles sont disponibles et insérées dans le cache des transactions.
3. La grille de données dans le groupe des serveurs contient la majorité des données et elle est partagée entre tous les clients. Le groupe des serveurs peut être partitionné, ce qui permet la mise en cache d'un grand nombre de données. Lorsque le cache local ne contient pas de données, celles-ci sont extraites du groupe des serveurs et insérées dans le cache du client. Le groupe des serveurs peut aussi avoir un plug-in Loader. Lorsque la grille ne contient pas les données demandées, le chargeur est appelé et les données résultantes sont insérées dans la grille à partir du magasin de données dorsal.

Pour désactiver le cache local, donnez la valeur 0 à l'attribut numberOfBuckets dans la configuration du descripteur eXtreme Scale des remplacements par le client. Pour plus d'informations sur les stratégies de verrouillage dans eXtreme Scale, consultez la rubrique relative au verrouillage des entrées de mappe. Le cache local peut également être configuré de façon à utiliser d'autres règles d'expulsion et des plug-in différents qui utilisent une configuration de descripteur eXtreme Scale des remplacements par les clients.

Avantage

- Rapidité du temps de réponse, car tous les accès aux données se font localement La recherche de données dans le cache local évite de consulter la grille des serveurs et rend les données distantes accessibles localement.

Inconvénients

- Augmentation de la durée des données obsolètes, car le cache local à chaque niveau est peut-être désynchronisé avec les données en cours dans la grille de données.
- Basé sur un expulseur pour invalider les données afin d'éviter de manquer de mémoire.

Utilisation

A utiliser lorsque le temps de réponse est élevé et que la présence de données périmées est tolérée.

Intégration de la base de données : caches avec écriture différée, caches en ligne et caches secondaires

WebSphere eXtreme Scale est utilisé pour servir de frontal à une base de données classiques et ainsi éliminer l'activité de lecture qui est normalement envoyée vers la base de données. Un cache cohérent peut être utilisé avec une application soit directement, soit indirectement en passant alors par un associateur relationnel d'objets (ORM). Le cache cohérent peut décharger des tâches de lecture la base de données ou le dorsal. Dans un scénario un tout petit peu plus complexe, comme celui d'un accès transactionnel à un dataset dans lequel seules certaines données requièrent des garanties de persistance classique, il est possible d'utiliser le filtrage pour décharger même les transactions d'écriture.

Vous pouvez configurer WebSphere eXtreme Scale pour qu'il fonctionne en tant qu'espace extrêmement flexible de traitement de base de données interne. Cela dit, WebSphere eXtreme Scale n'est pas un associateur relationnel d'objets. Il ne sait pas d'où les données de la grille de données proviennent. Une application ou un

associateur relationnel d'objets peuvent placer des données sur un serveur eXtreme Scale. C'est à la source de données qu'il incombe de vérifier la cohérence des données avec leur base de données d'origine. En d'autres termes, eXtreme Scale ne peut pas invalider les données qu'il a extraites automatiquement d'une base de données. C'est à l'application ou à l'associateur de fournir cette fonction et de gérer les données stockées dans eXtreme Scale.

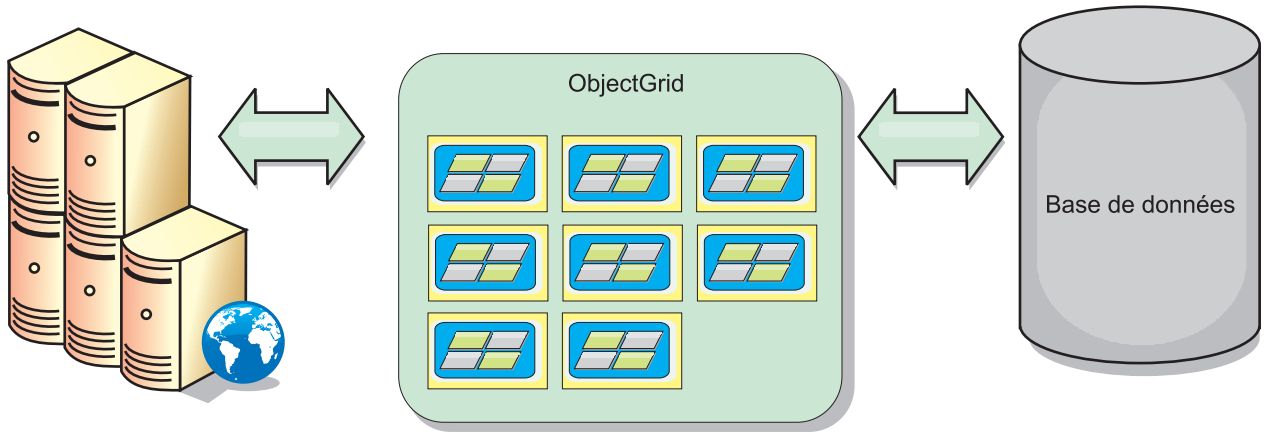


Figure 7. ObjectGrid en tant que mémoire tampon de base de données

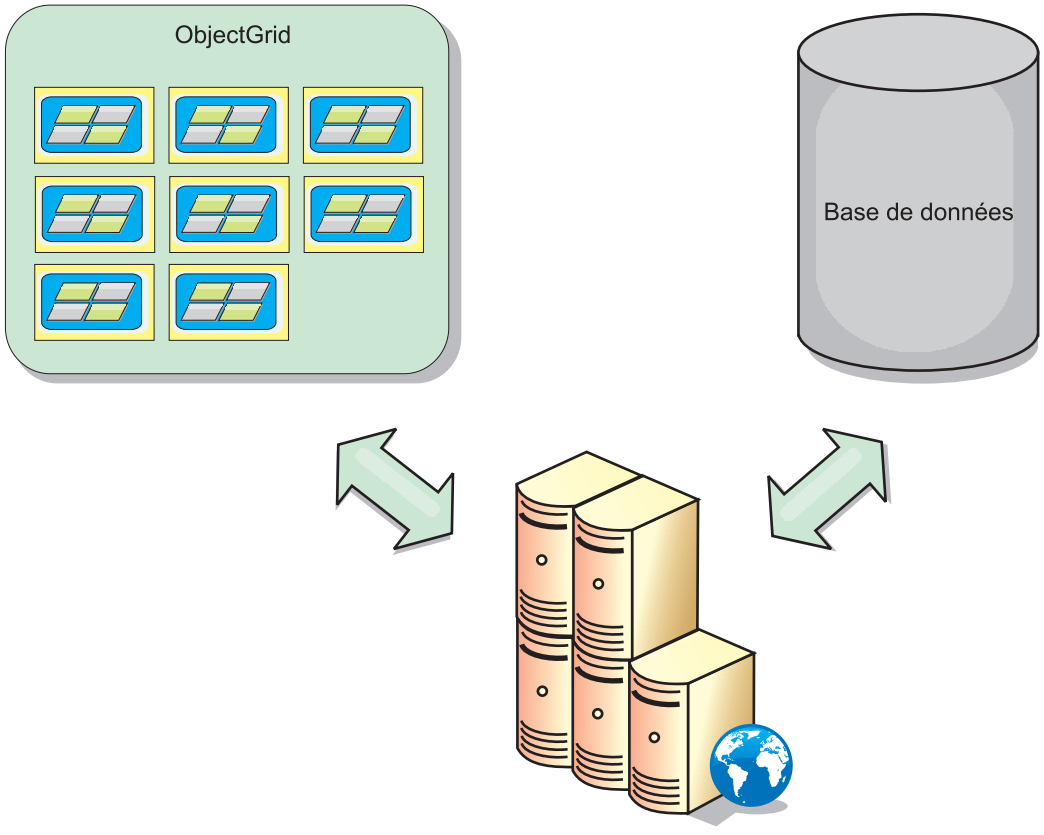


Figure 8. ObjectGrid en tant que cache secondaire

Cache partiel et cache complet

WebSphere eXtreme Scale peut s'utiliser en tant que cache partiel ou que cache complet. Un cache partiel ne conserve qu'un sous-ensemble des données totales,

alors qu'un cache complet conserve toutes les données et peut être rempli en différé en fonction des besoins en données. Les caches partiels sont normalement accessibles à l'aide de clés (et non pas d'index ou de requêtes), car les données sont partiellement disponibles uniquement.

Cache partiel

Si une clé est absente dans un cache partiel ou que les données ne sont pas disponibles et qu'un échec de cache se produit, le niveau suivant est appelé. Les données sont extraites d'une base de données, par exemple, et elles sont insérées au groupe de caches de grille de données. Si vous utilisez une requête ou un index, seules les valeurs actuellement chargées sont accessibles et les requêtes ne sont pas transférées aux autres groupes.

Cache complet

Un cache complet comporte toutes les données requises et il est possible d'y accéder à l'aide d'attributs non-clés avec des index ou des requêtes. Un cache complet est préchargé avec des données de la base de données avant que l'application tente d'accéder aux données. Un cache complet peut fonctionner sous la forme d'un remplacement de base de données une fois que les données sont chargées. Etant donné que toutes les données sont disponibles, les requêtes et les index peuvent être utilisés pour rechercher et agréger les données.

Cache secondaire

Lorsque WebSphere eXtreme Scale est utilisé en tant que cache secondaire, le système dorsal est utilisé avec la grille de données.

Cache secondaire

Vous pouvez configurer le produit en tant que cache secondaire pour la couche d'accès aux données d'une application. Dans ce scénario, WebSphere eXtreme Scale permet de stocker temporairement des objets qui seraient normalement extraits d'une base de données dorsale. Les applications vérifient si la grille de données contient les données. Si les données se trouvent dans la grille de données, ces données sont renvoyées à l'appelant. Si elles n'existent pas, elles sont extraites de la base de données dorsale. Elles sont ensuite insérées dans la grille de données afin que la demande suivante puisse utiliser la copie mise en cache. Le diagramme suivant montre comment WebSphere eXtreme Scale peut être utilisé en tant que cache secondaire à l'aide d'une couche d'accès aux données arbitraire, telle qu'OpenJPA ou Hibernate.

Plug-in de cache secondaire pour Hibernate et OpenJPA

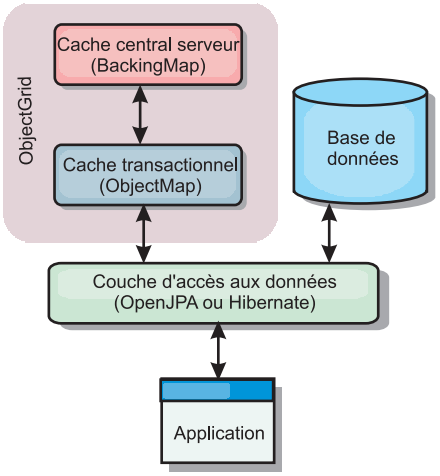


Figure 9. Cache secondaire

Les plug-in de cache pour OpenJPA et Hibernate sont inclus dans WebSphere eXtreme Scale pour que vous puissiez utiliser le produit comme cache secondaire automatique. L'utilisation d'WebSphere eXtreme Scale en tant que fournisseur de cache améliore les performances lors de la lecture et de l'interrogation des données et réduit la charge pesant sur la base de données. WebSphere eXtreme Scale présente plusieurs avantages par rapport à des implémentations de cache pré-intégrées car le cache est automatiquement répliqué entre tous les processus. Lorsqu'un client met une valeur en mémoire cache, tous les autres clients peuvent l'utiliser.

Cache en ligne

Vous pouvez configurer la mise en cache en ligne pour un système dorsal de base de données ou en tant que cache secondaire pour une base de données. La mise en cache en ligne utilise eXtreme Scale comme moyen principal pour interagir avec les données. Lorsque eXtreme Scale est utilisé en tant que cache en ligne, l'application interagit avec le système dorsal à l'aide d'un plug-in Loader.

Cache en ligne

Lorsque WebSphere eXtreme Scale est utilisé en tant que cache en ligne, il interagit avec le système dorsal à l'aide d'un plug-in Loader. Ce scénario permet de simplifier l'accès aux données car les applications peuvent accéder aux API eXtreme Scale directement. Plusieurs scénarios de cache sont pris en charge dans eXtreme Scale pour assurer la synchronisation des données dans le cache et des données dans le système dorsal. Le diagramme suivant illustre l'interaction entre le cache en ligne, l'application et le système dorsal.

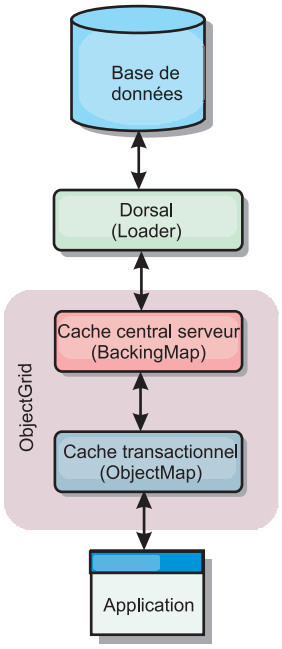


Figure 10. Cache en ligne

L'option de mise en cache en ligne simplifie l'accès aux données en permettant aux applications d'accéder directement aux API eXtreme Scale. WebSphere eXtreme Scale prend en charge plusieurs scénarios de mise en cache en ligne, comme suit.

- Sans interruption
- Ecriture immédiate
- Post-écriture

Scénario de mise en cache sans interruption

Un cache sans interruption est un cache partiel chargeant en lazy loading à partir d'une clé les entrées de données au fur et à mesure que ces entrées sont demandées. Cette opération peut se dérouler sans que l'appelant sache comment sont renseignées les entrées. Si les données sont introuvables dans le cache eXtreme Scale, eXtreme Scale récupère les données manquantes auprès du plug-in Loader qui charge les données provenant de la base de données d'arrière plan et les insère dans le cache. Les requêtes suivantes pour la même clé de données se trouveront dans le cache, jusqu'à ce qu'elles soient supprimées, invalidées ou expulsées.

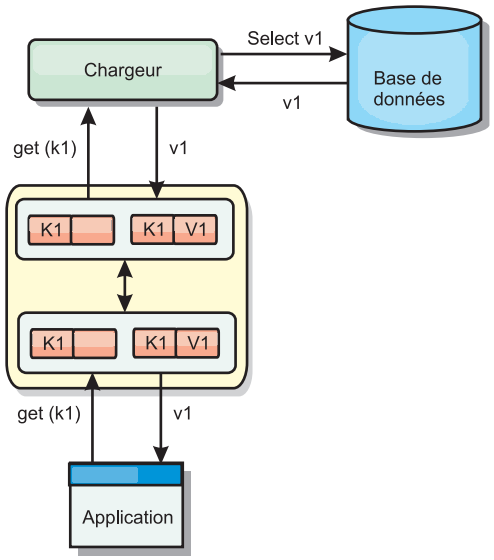


Figure 11. Mise en cache sans interruption

Scénario de mise en cache à écriture immédiate

Dans un cache à écriture immédiate, chaque écriture dans le cache est inscrite de manière synchrone dans la base de données à l'aide du chargeur. Cette méthode permet la cohérence avec le système dorsal, mais réduit les performances d'écriture étant donné que l'opération de base de données est synchrone. Le cache et la base de données étant tous deux mis à jour, les lectures suivantes à la recherche des mêmes données auront lieu dans le cache, évitant ainsi de faire appel à la base de données. Un cache à écriture immédiate est souvent utilisé conjointement à un cache sans interruption.

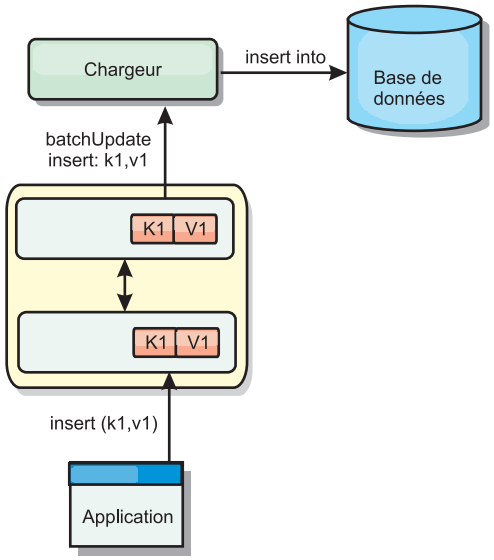


Figure 12. Mise en cache à écriture immédiate

Scénario de mise en cache en écriture différée

La synchronisation de la base de données peut être améliorée en écrivant les modifications de manière asynchrone. Cette opération est appelée mise en cache en

écriture différée. Les modifications, normalement écrites de manière synchrone dans le chargeur, sont mises en mémoire tampon dans eXtreme Scale et écrites dans la base de données à l'aide d'une unité d'exécution en arrière-plan. Les performances d'écriture sont considérablement améliorées, car l'opération de base de données est supprimée de la transaction client et les écritures de la base de données peuvent être comprimées.

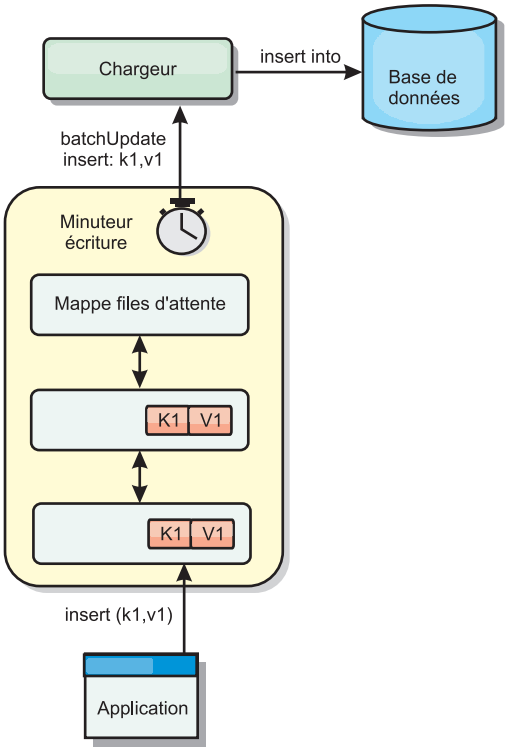


Figure 13. Mise en cache en écriture différée

Mise en cache en écriture différée

Vous pouvez utiliser la mise en cache en écriture différée pour réduire le temps système supplémentaire nécessaire lors de la mise à jour d'une base de données utilisée en tant que base de données dorsale.

Présentation de la mise en cache en écriture différée

La mise en cache en écriture différée met en file d'attente de manière asynchrone les mises à jour du plug-in Loader. Vous pouvez améliorer les performances en déconnectant les mises à jour, les insertions et les suppressions au sein d'une mappe, le temps système pour la mise à jour de la base de données dorsale. La mise à jour asynchrone est effectuée après un retard (de cinq minutes, par exemple) ou après un certain nombre d'entrées (1 000 entrées).

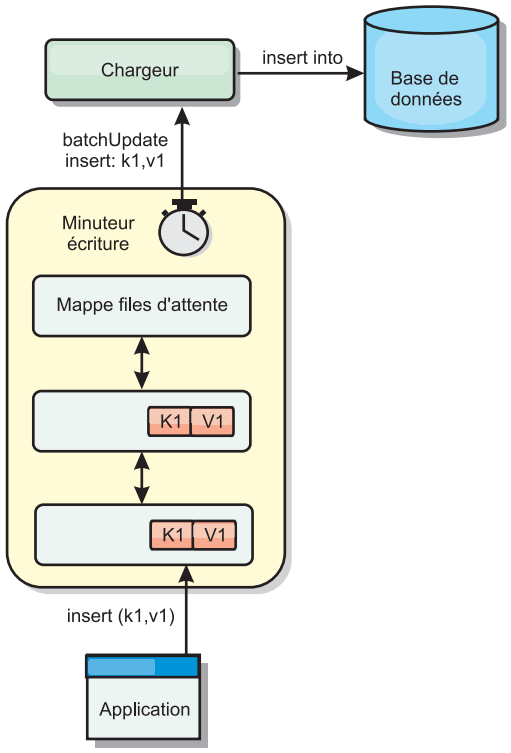


Figure 14. Mise en cache en écriture différée

La configuration à écriture différée sur une mappe de sauvegarde crée une unité d'exécution entre le Loader et la mappe. Le Loader délègue alors les demandes de données via l'unité d'exécution en fonction des paramètres de configuration de la méthode BackingMap.setWriteBehind. Lorsqu'une transaction eXtreme Scale insère, met à jour ou supprime une entrée dans une mappe, un objet LogElement est créé pour chacun de ces enregistrements. Ces éléments sont envoyés au Loader à écriture différée et mis en file d'attente dans une ObjectMap spéciale appelée mappe de files d'attente. Chaque mappe de sauvegarde pour laquelle le paramètre d'écriture différée est activé a ses propres mappes de files d'attente. L'unité d'exécution à écriture différée supprime périodiquement les données mises en file d'attente des mappes correspondantes et les insère dans le Loader dorsal.

Le chargeur à écriture différée envoie uniquement les types insertion, mise à jour et suppression des objets LogElement au chargeur réel. Tous les autres types, par exemple le type EVICT, sont ignorés.

La prise en charge de l'écriture différée *est* une extension du plug-in Loader, qui vous permet d'intégrer eXtreme Scale à la base de données. A ce sujet, vous pouvez consulter avec profit les explications «Configuration des chargeurs JPA», à la page 353 sur la configuration d 'un chargeur JPA.

Avantages

L'activation de l'écriture différée présente les avantages suivants :

- **Isolement en cas d'arrêt anormal de la base de données dorsale :** la mise en cache à écriture différée propose une couche d'isolement en cas d'arrêt anormal de la base de données dorsale. Les mises à jour sont alors placées dans la mappe de files d'attente. Les applications peuvent continuer à envoyer des transactions

vers eXtreme Scale. Lors de la reprise du système dorsal, les données contenues dans la mappe de files d'attente sont insérées dans celui-ci.

- **Réduction de la charge du système dorsal :** le chargeur à écriture différée fusionne les mises à jour en fonction des clés de façon qu'une seule mise à jour fusionnée par clé existe dans la mappe de files d'attente. Cette fusion diminue le nombre de mises à jour dans la base de données dorsale.
- **Amélioration des performances de la transaction :** la durée de chaque transaction eXtreme Scale est réduite car la transaction n'a plus à attendre que les données soient synchronisées avec le système dorsal.

Considérations liées à la conception d'applications

L'activation de l'écriture différée est une opération simple, mais la création d'une application devant utiliser l'écriture différée requiert une attention particulière. Sans écriture différée, la transaction ObjectGrid encadre la transaction dorsale. La transaction ObjectGrid démarre avant la transaction dorsale et se termine après celle-ci.

Lorsque la prise en charge de l'écriture différée est activée, la transaction ObjectGrid se termine avant le début de la transaction dorsale. La transaction ObjectGrid et la transaction dorsale sont dissociées.

Contraintes d'intégrité référentielle

Chaque mappe de sauvegarde configurée avec écriture différée dispose de sa propre unité d'exécution d'écriture différée pour envoyer les données vers le système dorsal. Les données mises à jour dans les différentes mappes d'une transaction ObjectGrid sont mises à jour dans le système dorsal via différentes transactions dorsales. Par exemple, la transaction T1 met à jour la clé key1 dans la mappe Map1 et la clé key2 dans la mappe Map2. La clé key1 mise à jour dans la mappe Map1 est actualisée vers le système dorsal dans une transaction expéditrice et la clé key2 actualisée dans la mappe Map2 l'est dans une autre transaction expéditrice ; cette mise à jour vers le dorsal est effectuée dans deux unités d'exécution différentes, toutes deux en écriture différée. Si les données stockées dans Map1 et Map2 ont des liens, tels que des contraintes de clé externe dans le système dorsal, les mises à jour sont susceptibles d'échouer.

Lors de la conception de contraintes d'intégrité référentielle dans votre base de données dorsale, vérifiez que les mises à jour désordonnées sont autorisées.

Verrouillage d'une mappe de files d'attente

Le comportement des transactions en matière de verrouillage constitue une autre différence notable. La grille d'objets prend en charge trois stratégies de verrouillage différentes : PESSIMISTIC, OPTIMISITIC et NONE. Les mappes de files d'attente à écriture différée utilisent la stratégie de verrouillage pessimiste, quelle que soit la stratégie de verrouillage configurée pour leur mappe de sauvegarde. Deux types différents d'opérations permettant d'acquérir un verrou sur la mappe de files d'attente existent :

- Lorsqu'une transaction ObjectGrid est validée ou qu'un vidage (de mappe ou de session) se produit, la transaction lit la clé de la mappe de files d'attente et place un verrou S sur la clé.
- Lorsqu'une transaction ObjectGrid est validée, elle tente de mettre à niveau le verrou S vers un verrou X sur la clé.

Ce comportement de mappe de files d'attente supplémentaire vous permet de voir quelques différences dans le comportement de verrouillage.

- Si la mappe de l'utilisateur est configurée comme stratégie de verrouillage pessimiste, la différence dans le comportement de verrouillage n'est pas grande. Chaque fois qu'un vidage ou qu'une validation est appelée, un verrou S est placé sur la même clé dans la mappe de files d'attente. Au moment de la validation, un verrou X est acquis pour la clé dans la mappe de l'utilisateur, mais également pour la clé dans la mappe de files d'attente.
- Si la mappe de l'utilisateur est configurée comme stratégie de verrouillage optimiste ou inexistante, la transaction utilisateur suit le modèle de la stratégie pessimiste. Chaque fois qu'un vidage ou qu'une validation est appelée, un verrou S est acquis sur la même clé dans la mappe de files d'attente. Au moment de la validation, un verrou X est acquis pour la clé dans la mappe de files d'attente utilisant la même transaction.

Nouvelles tentatives de transaction du chargeur

ObjectGrid ne prend pas en charge les transactions à 2 phases ou transactions XA. L'unité d'exécution à écriture différée supprime les enregistrements de la mappe de files d'attente et met à jour les enregistrements dans le système dorsal. En cas d'échec du serveur au milieu de la transaction, certaines mises à jour dorsales risquent d'être perdues.

Le chargeur à écriture différée tente automatiquement d'écrire à nouveau les transactions ayant échoué et envoie une LogSequence en attente de validation au système dorsal pour éviter toute perte de données. Pour que l'exécution de cette action soit possible, le chargeur doit être idempotent, ce qui signifie que lorsque Loader.batchUpdate(TxId, LogSequence) est appelé deux fois avec la même valeur, le résultat est le même que s'il était appelé une fois. Les implémentations du chargeur doivent implémenter l'interface RetryableLoader pour activer cette fonction. Pour plus de détails, consultez la documentation relative à l'API.

Echecs du chargeur

Le plug-in du chargeur (Loader) risque d'échouer lorsqu'il ne parvient pas à communiquer avec le dorsal de base de données. Cela peut se produire si le serveur de base de données ou la connexion réseau est arrêté(e). Le chargeur en écriture différée met en file d'attente les mises à jour et tente d'envoyer régulièrement les données modifiées au chargeur. Ce dernier doit signaler le problème de connectivité à l'environnement d'exécution ObjectGrid en générant une exception LoaderNotAvailableException.

L'implémentation du chargeur doit donc pouvoir distinguer un échec lié aux données d'une défaillance physique du chargeur. En cas d'échec lié aux données, une exception LoaderException ou OptimisticCollisionException doit être générée, alors qu'en cas de défaillance physique du chargeur, une exception LoaderNotAvailableException doit être générée. ObjectGrid gère ces deux exceptions de manière différente :

- Si une exception LoaderException est interceptée par le chargeur en écriture différée, celui-ci considère que l'échec est dû à une défaillance de données, telle qu'une erreur de clé en double. Le chargeur dégroupe la mise à jour et tente de ne mettre à jour qu'un enregistrement à la fois, afin d'isoler la défaillance de données. Si une exception {{LoaderException}} est à nouveau détectée lors de la mise à jour de l'enregistrement concerné, un enregistrement d'échec de la mise à jour est créé et consigné dans la mappe des mises à jour ayant échoué.

- Si une exception LoaderNotAvailableException est interceptée par le chargeur en écriture différée, celui-ci considère que l'échec est dû à l'impossibilité de se connecter à la base de données, par exemple, lorsque la base de données dorsale est inactive, lorsque la connexion à une base de données est indisponible ou lorsque le réseau est inactif. Le chargeur attend 15 secondes, puis tente à nouveau la mise à jour par lots de la base de données.

L'erreur courante est d'émettre une exception LoaderException à la place d'une exception LoaderNotAvailableException. Tous les enregistrements du chargeur à écriture différée deviennent alors des enregistrements d'échec de la mise à jour, ce qui réduit à néant l'objectif de l'isolement en cas d'arrêt anormal du système dorsal.

Remarques sur les performances

En supprimant de la transaction la mise à jour du chargeur, la mise en cache en écriture différée augmente le temps de réponse. Elle augmente également la capacité de traitement de la base de données, car les mises à jour de base de données sont combinées. Il est important de comprendre le temps système supplémentaire généré par l'unité d'exécution à écriture différée, qui permet de retirer les données de la mappe de files d'attente et de les insérer dans le chargeur.

Le temps de mise à jour maximal et le nombre de mises à jour maximal doivent être ajustés en fonction de l'environnement et des types d'utilisation prévus. Si la valeur du temps ou du nombre de mises à jour maximal est trop petite, le temps système de l'unité d'exécution d'écriture différée peut dépasser les avantages tirés. Définir une valeur élevée pour ces deux paramètres peut également augmenter l'utilisation de la mémoire pour la mise en file d'attente des données et retarder le moment de péremption des enregistrements de bases de données.

Pour des performances optimales, réglez les paramètres d'écriture différée en fonction des facteurs suivants :

- ratio des transactions de lecture et d'écriture
- fréquence de mise à jour d'enregistrements identiques
- temps d'attente pour la mise à jour de la base de données

Chargeurs

Avec un plug-in Loader, une mappe de grille de données peut se comporter comme un cache pour les données généralement conservées dans un magasin persistant sur le même système ou un autre système. Généralement, une base de données ou un système de fichiers est utilisé comme stockage de persistance. Une machine virtuelle Java (JVM) peut également être utilisée comme source des données, ce qui permet de créer des caches basés sur un concentrateur à l'aide d'eXtreme Scale. Un chargeur peut lire et écrire des données vers un stockage persistant ou à partir de celui-ci.

Présentation

Les chargeurs sont des plug-in de mappe de sauvegarde appelés lorsque des modifications sont apportées à la mappe de sauvegarde ou lorsque cette dernière est dans l'impossibilité de répondre à une demande de données (absence dans le cache). Le chargeur est appelé lorsque le cache ne peut pas satisfaire une demande de clé, offrant ainsi une fonction de lecture et un remplissage laborieux du cache. Un chargeur permet également les mises à jour de la base de données lorsque les valeurs du cache viennent à changer. Toutes les modifications dans une transaction

sont regroupées pour réduire le nombre d'interactions de base de données. Un plug-in TransactionCallback est utilisé conjointement avec le chargeur pour déclencher la démarcation de la transaction principale. L'utilisation de ce plug-in est importante lorsque plusieurs mappes sont incluses dans une seule transaction ou lorsque les données de transaction sont vidées dans le cache sans validation.

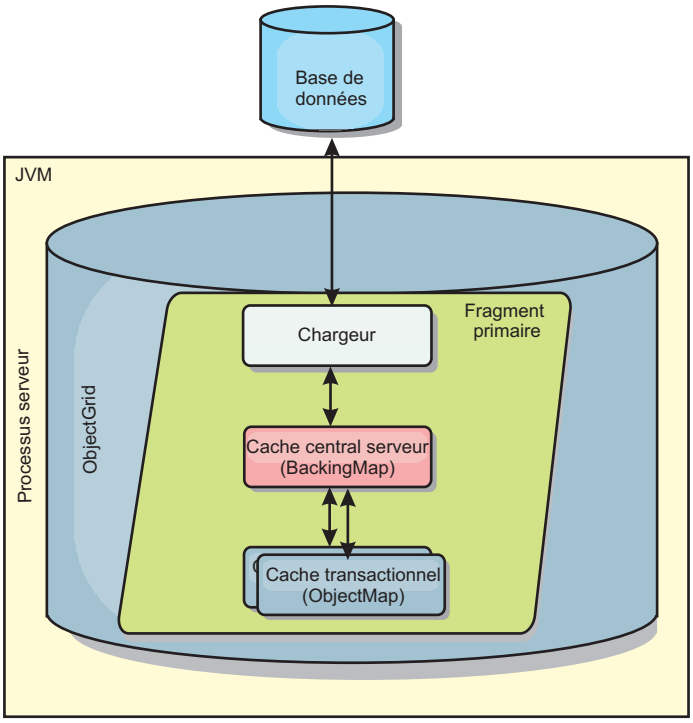


Figure 15. Chargeur

Le chargeur peut donc utiliser les mises à jour sur-qualifiées pour éviter le verrouillage intempestif de la base de données. En stockant un attribut de version dans la valeur du cache, le chargeur peut distinguer l'image de la valeur avant et après la mise à jour dans le cache. Cette valeur peut ensuite être utilisée lors de la mise à jour de la base de données ou du programme d'arrière plan pour vérifier que les données n'ont pas été mises à jour. Un chargeur peut également être configuré pour précharger la grille de données lorsqu'elle démarre. Lorsqu'elle est partitionnée, une instance de chargeur est associée à chaque partition. Si la mappe de la société comporte dix partitions, il existe dix instances de chargeur, une pour chaque partition principale. Lorsque le fragment primaire de la mappe est activé, la méthode preloadMap du chargeur est appelée de manière synchrone ou asynchrone, ce qui déclenche le chargement automatique de la partition de la mappe avec les données du programme d'arrière plan. Lorsqu'il est appelé de manière synchrone, toutes les transactions client sont bloquées, ce qui empêche tout accès incohérent à la grille de données. Sinon, un préchargeur client peut être utilisé pour charger l'intégralité de la grille de données.

Deux chargeurs pré-intégrés peuvent simplifier considérablement l'intégration aux dorsaux de bases de données relationnelles. Les chargeurs JPA utilisent les fonctions du mappage objet-relationnel(ORM) des implémentations OpenJPA et Hibernate des spécifications JPA (Java Persistence API). Pour plus d'informations, voir Chargeurs JPA.

Si vous utilisez des chargeurs dans une configuration à plusieurs centre de données, vous devez étudier la façon dont les données de révision et la cohérence de la mémoire cache est conservée entre les grilles de données. Pour plus d'informations, voir «Remarques sur les chargeurs dans une topologie multimaître» , à la page 41.

Configuration de chargeur

Pour ajouter un chargeur à la configuration BackingMap, vous pouvez utiliser la configuration à l'aide d'un programme ou la configuration XML. Un chargeur a la relation suivante avec une mappe de sauvegarde.

- Une mappe de sauvegarde peut avoir un seul chargeur.
- Une mappe de sauvegarde client (cache local) ne peut pas avoir de chargeur.
- Une définition de chargeur peut être appliquée à plusieurs mappes de sauvegarde, mais chaque mappe de sauvegarde dispose de sa propre instance de chargeur.

Préchargement et préremplissage des données

Dans la plupart des scénarios qui utilise un chargeur, vous pouvez préparer la grille de données en y préchargeant ses données.

Lorsque vous utilisez la grille de données comme un cache complet, elle doit contenir toutes les données et elle doit être chargée pour que les clients puissent s'y connecter. Lorsque vous utilisez un cache partiel, vous pouvez préparer le cache avec des données pour que les clients puissent avoir accès immédiatement à ces données dès qu'ils se connectent.

Il existe deux approches pour pré-charger des données dans la grille de données ; vous pouvez utiliser un plug-in Loader ou un chargeur client, comme décrit dans les sections suivantes.

Plug-in Loader

Le plug-in Loader est associé à chaque mappe et chargé de synchroniser un fragment primaire de partition avec la base de données. La méthode preloadMap du plug-in Loader est invoquée automatiquement lors de l'activation d'un fragment, Par exemple, vous disposez de 100 partitions, il existe 100 instances Loader, chacune chargeant les données de sa partition. En cas d'exécution synchrone, tous les clients sont bloqués jusqu'à la fin du préchargement.

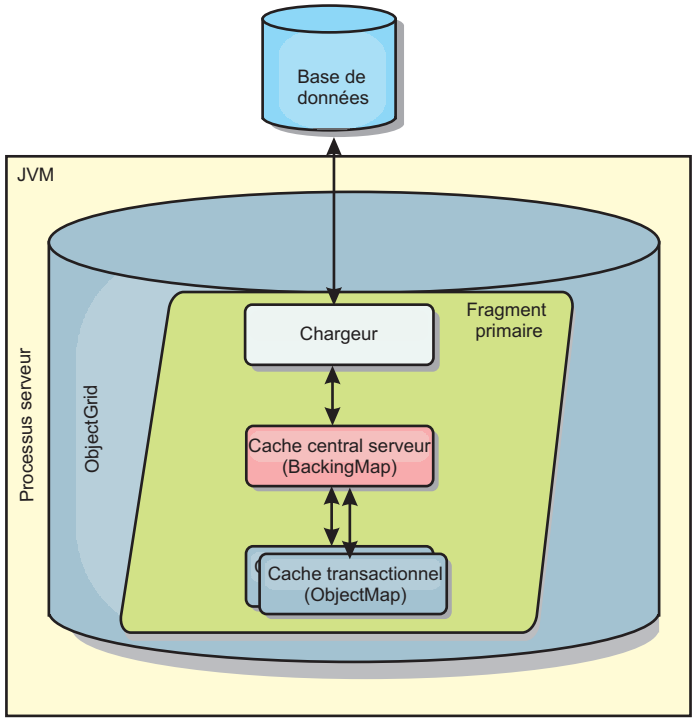


Figure 16. Plug-in Loader

Loader client

Un loader client est un pattern d'utilisation d'un ou plusieurs clients pour charger les données dans la grille. L'utilisation de plusieurs clients pour charger les données de la grille peut s'avérer efficace lorsque le schéma de partition n'est pas stocké dans la base de données. Vous pouvez appeler des chargeurs de client manuellement ou automatiquement lorsque la grille de données démarre. Ces chargeurs peuvent éventuellement utiliser StateManager pour faire passer la grille de données en mode de préchargement pour que les clients ne puissent pas accéder à la grille lorsqu'elle précharge les données. WebSphere eXtreme Scale contient un chargeur JPA (Java Persistence API) que vous pouvez utiliser pour charger automatiquement la grille de données avec le fournisseur JPA OpenJPA ou Hibernate. Pour plus d'informations sur les fournisseurs de cache, voir «Plug-in de cache niveau 2 (L2) JPA», à la page 331.

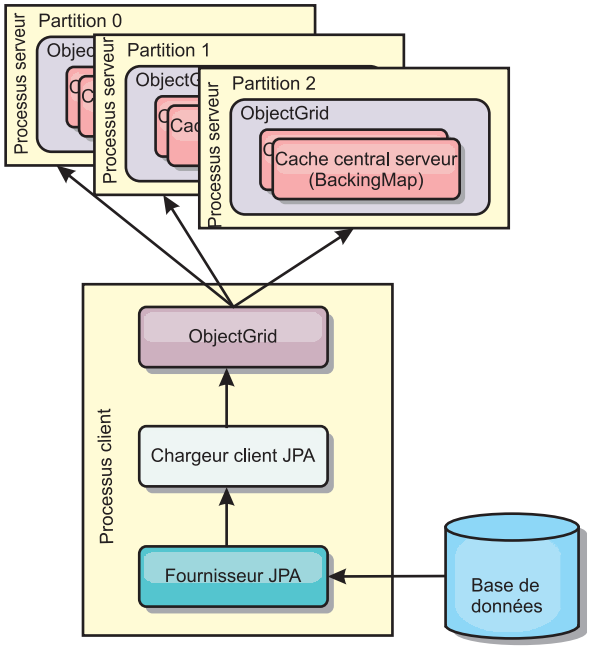


Figure 17. Loader client

Méthodes de synchronisation de base de données

Lorsque WebSphere eXtreme Scale est utilisé en tant que cache, les applications doivent être écrites de sorte qu'elles tolèrent les données périmées si la base de données peut être mise à jour de manière indépendante par rapport à une transaction eXtreme Scale. En tant qu'espace de traitement de base de données en mémoire synchronisé, eXtreme Scale permet d'assurer la mise à jour du cache de plusieurs manières.

Méthodes de synchronisation de base de données

Actualisation régulière

Le cache peut être régulièrement invalidé ou mis à jour de manière automatique à l'aide du programme de mise à jour temporelle de base de données JPA (Java Persistence API).Le programme de mise à jour interroge régulièrement la base de données à l'aide d'un fournisseur JPA, afin de rechercher des mises à jour ou des insertions survenues depuis la mise à jour précédente. Tous les changements détectés sont automatiquement invalidés ou mis à jour lorsqu'ils sont utilisés avec un cache incomplet. S'ils sont utilisés avec un cache complet, les entrées peuvent être détectées et insérées dans le cache. Les entrées ne sont jamais supprimées du cache.

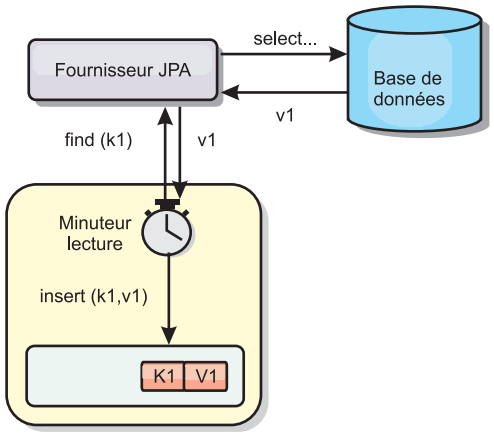


Figure 18. Actualisation régulière

Suppression

Les caches incomplets peuvent utiliser les stratégies de suppression pour supprimer automatiquement les données du cache sans que cela n'affecte la base de données. eXtreme Scale inclut trois stratégies : durée de vie, utilisation la moins récente et utilisation la moins fréquente. Si l'option de suppression en fonction de la mémoire est activée, ces trois stratégies suppriment les données de manière plus agressive à mesure que la mémoire est limitée.

Invalidation en fonction d'événements

Il est possible d'invalider les caches partiels et complets à l'aide d'un générateur d'événements comme JMS (Java Message Service). L'invalidation par le biais de JMS peut être associée de manière manuelle à tout processus qui met à jour le dorsal à l'aide d'un déclencheur de base de données. eXtreme Scale contient un plug-in JMS ObjectGridEventListener qui informe les clients des éventuelles modifications du cache du serveur. Cette procédure peut réduire la durée d'accès du client aux données périmées.

Invalidation par programme

Les API eXtreme Scale permettent l'interaction manuelle du cache local et du cache serveur à l'aide des méthodes des API Session.beginNoWriteThrough(), ObjectMap.invalidate() et EntityManager.invalidate(). Si un processus client ou serveur n'a plus besoin d'une partie des données, les méthodes d'invalidation peuvent être utilisées pour supprimer les données du serveur local ou du serveur cache. La méthode beginNoWriteThrough applique une opération ObjectMap ou EntityManager au cache local sans appeler le programme de chargement. Si l'opération est appelée à partir d'un client, elle s'applique uniquement au cache local (le programme de chargement distant n'est pas appelé). Si elle est appelée sur le serveur, l'opération s'applique uniquement au cache central du serveur sans appeler le programme de chargement.

L'invalidation des données

Pour supprimer les données de la mémoire cache d'évolution, vous pouvez utiliser un mécanisme d'invalidation basé sur les événements ou à l'aide d'un programme.

Invalidation basée sur les événements

Il est possible d'invalider les caches incomplets et complets à l'aide d'un générateur d'événements tel que Java Message Service (JMS). L'invalidation par le biais de JMS peut être associée de manière manuelle à tout processus qui met à jour le dorsal à l'aide d'un déclencheur de base de données. Un plug-in JMS ObjectGridEventListener est fourni dans eXtreme Scale pour permettre aux clients d'être informés des modifications dans le cache du serveur. Ce type de notification peut réduire la durée d'accès du client aux données obsolètes.

L'invalidation basée sur les événements est composée normalement des trois composants suivants.

- **File d'attente des événements :** une file d'attente d'événements stocke les événements de modification des données. Il peut s'agir d'une file d'attente JMS, d'une base de données, d'une file d'attente interne premier entré premier sorti ou de tout autre événement dans la mesure où elle peut gérer les événements de modification des données.
- **Publicateur d'événements :** un publicateur d'événements publie les événements de modification de données dans la file d'attente d'événements. Une publicateur d'événements est généralement une application que vous créez ou une implémentation de plug-in eXtreme Scale. Il sait quand les données ont été modifiées ou il modifie les données lui-même. Lorsqu'une transaction est validée, les événements sont générés pour les données modifiées et le publicateur d'événements publie ces événements dans la file d'attente d'événements.
- **Consommateur d'événements :** un consommateur d'événements consomme les événements de modification de données. Le consommateur d'événements est généralement une application permettant de vérifier la mise à jour des données de la grille cible avec les dernières modifications apportées aux autres grilles. Il interagit avec la file d'attente d'événements pour récupérer les dernières données et applique les modifications apportées aux données dans la grille cible. Les consommateurs d'événements peuvent utiliser les API eXtreme Scale pour invalider les données obsolètes ou mettre à jour la grille avec les dernières données.

Par exemple, JMSObjectGridEventListener comporte une option pour un modèle client-serveur dans lequel la file d'attente d'événements est une destination JMS désignée. Tous les processus serveur sont des publicateurs d'événements. Lorsqu'une transaction est validée, le serveur récupère les modifications apportées aux données et les publie à la destination JMS désignée. Tous les processus client sont des consommateurs d'événements. Ils reçoivent les modifications apportées aux données de la destination JMS désignée et appliquent les modifications au cache local du client.

Pour plus d'informations, consultez la rubrique relative au mécanisme d'invalidation de client dans le *Guide de l'administration* .

Invalidation par programme

Les API WebSphere eXtreme Scale autorise l'interaction manuelle du cache local et du cache serveur à l'aide des méthodes Session.beginNoWriteThrough(), ObjectMap.invalidate() et EntityManager.invalidate(). Si un processus client ou serveur n'a plus besoin d'une partie des données, les méthodes invalidate permettent de supprimer des données d'un cache local ou de serveur. La méthode beginNoWriteThrough applique toutes les opérations ObjectMap ou EntityManager

au cache local sans appeler le chargeur. Si l'opération est appelée à partir d'un client, elle s'applique uniquement au cache local (le programme de chargement distant n'est pas appelé). Si elle est appelée sur le serveur, l'opération s'applique uniquement au cache central du serveur sans appeler le programme de chargement.

Vous pouvez utiliser l'invalidation par programme à l'aide d'autres techniques pour déterminer quand il convient d'invalider les données. Par exemple cette méthode d'invalidation utilise des mécanismes d'invalidation basée sur les événements pour recevoir les événements de modification de données, puis utilise les API pour invalider les données obsolètes.

Indexation

Utilisez le plug-in MapIndexPlugin pour générer un ou plusieurs index dans une mappe BackingMap pour prendre en charge l'accès aux données ne correspondant pas à une clé.

Types d'indexation et configuration d'index

L'indexation est représentée par le plug-in MapIndexPlugin ou Index, en bref. Index est un plug-in BackingMap. Une mappe de sauvegarde peut avoir plusieurs index configurés, dès lors que chacun d'entre eux respecte les règles de configuration d'index.

Vous pouvez utiliser l'indexations pour générer une ou plusieurs index dans une mappe BackingMap. Un index se construit à partir d'un attribut ou d'une liste des attributs d'un objet de la mappe. L'indexation permet aux applications de trouver plus rapidement certains objets. Grâce à elle, en effet, les applications peuvent trouver les objets dont les attributs indexés ont une certaine valeur ou se situent dans une plage de valeurs.

Deux types d'indexation sont possibles : statiques et dynamiques. L'indexation statique oblige à configurer le plug-in d'indexation index dans la mappe de sauvegarde avant d'initialiser l'instance ObjectGrid. Comme pour la mappe de sauvegarde, cela peut se faire par programmation ou via XML. L'indexation statique commence à générer l'index pendant l'initialisation de la grille d'objets. L'index est synchrone en permanence avec la mappe de sauvegarde et il est prêt à être utilisé. Après que l'indexation statique a démarré, la maintenance de l'index fait partie de la gestion des transactions par eXtreme Scale. Lorsque les transactions valident leurs modifications, ces dernières actualisent également l'index statique et les modifications apportées à l'index sont annulées en cas d'annulation de la transaction.

L'indexation dynamique permet de créer un index dans une mappe de sauvegarde avant ou après l'initialisation de l'instance ObjectGrid qui contient cette mappe. Les applications contrôlent le cycle de vie de l'indexation dynamique, ce qui permet de supprimer un index dynamique devenu inutile. Lorsqu'une application crée un index dynamique, cet index n'est pas forcément utilisable immédiatement en raison du temps que met à s'effectuer la génération complète de l'index. Comme la durée dépend de la quantité de données indexées, l'interface DynamicIndexCallback est fournie pour les applications qui souhaitent recevoir des notifications lorsque se produisent certains événements l'indexation, à savoir les événements ready, error et destroy. Les applications peuvent implémenter cette interface de rappel et s'enregistrer auprès de l'indexation dynamique.

Si un plug-in d'indexation est configuré pour une mappe de sauvegarde, il est possible d'obtenir de la mappe d'objet correspondante l'objet proxy de l'index. L'appel de la méthode getIndex dans la mappe et la transmission du nom du plug-in Index renvoie l'objet proxy de l'index. L'objet proxy doit être transtypé vers l'interface d'indexation de l'application utilisée, MapIndex, MapRangeIndex ou une interface d'indexation personnalisée, par exemple. Une fois l'objet proxy obtenu, l'on peut utiliser les méthodes définies dans l'interface d'indexation de l'application afin de trouver des objets mis en cache.

La liste qui suit récapitule la procédure à appliquer pour procéder à l'indexation :

- ajout d'index statiques ou dynamiques dans la mappe de sauvegarde
- obtention d'un objet proxy d'index grâce à la méthode getIndex de la mappe d'objet
- transtypage de l'objet proxy vers l'interface d'indexation de l'application utilisée (MapIndex, MapRangeIndex ou une interface d'indexation personnalisée, par exemple)
- utilisation des méthodes qui sont définies dans l'interface d'indexation de l'application pour rechercher les objets mis en cache

La classe HashIndex est l'implémentation du plug-in d'indexation pré-intégré capable de prendre en charge les deux interfaces pré-intégrées d'API d'indexation : MapIndex et MapRangeIndex. Vous pouvez également créer vos propres index. Vous pouvez ajouter HashIndex à la BackingMap en tant qu'index statique ou dynamique, obtenir un objet proxy d'index MapIndex ou MapRangeIndex et utiliser cet objet proxy pour chercher des objets mis en cache.

Index par défaut

Si vous souhaitez effectuer une itération dans les clés d'une mappe locale, vous pouvez utiliser l'index par défaut. Cet index ne requiert pas de configuration, mais elle doit être utilisée sur le fragment en utilisant un agent ou une instance ObjectGrid extraite de la méthode ShardEvents.shardActivated(ObjectGrid shard).

Indexation et qualité des données obtenues par une requête d'index

Il faut bien avoir présent à l'esprit que les méthodes de requêtes sur les index ne représentent qu'un cliché des données à un instant t. Les entrées de données ne sont pas verrouillées après l'envoi à l'application des résultats de la requête. L'application doit être consciente que les données peuvent très bien être actualisées après lui avoir été retournées. Supposons, par exemple, que l'application obtienne la clé d'un objet mis en cache grâce à la méthode findAll de MapIndex. Cet objet key retourné est associé dans le cache à une entrée de données. L'application doit être capable d'exécuter la méthode get sur la mappe d'objet pour trouver un objet à partir de l'objet key. Si une autre transaction supprime du cache l'objet données juste avant l'appel à la méthode get, le résultat qui sera retourné sera null.

Points à prendre en considération à propos des performances de l'indexation

L'un des objectifs primordiaux de l'indexation est d'améliorer les performances globales de la mappe de sauvegarde. Une utilisation incorrecte de l'indexation peut compromettre les performances de l'application. Avant d'utiliser l'indexation, les facteurs suivants sont à prendre en considération :

- **Le nombre de transactions simultanées en écriture :** l'indexation peut se produire chaque fois qu'une transaction écrit des données dans une mappe de

sauvegarde. Les performances se dégradent si un grand nombre de transactions écrivent en même temps des données dans la mappe au moment où une application lance des requêtes sur l'index.

- **La taille des résultats retournés par une requête :** les performances de la requête déclinent d'autant plus que la taille de ses résultats augmente. Les performances tendent à se dégrader lorsque la taille des résultats atteint 15 % ou plus de la mappe de sauvegarde.
- **Le nombre d'index générés sur la même mappe de sauvegarde :** chaque index consomme des ressources système. Les performances diminuent au fur et à mesure que le nombre d'index augmente sur la mappe de sauvegarde.

Cela dit, l'indexation peut augmenter considérablement les performances des mappes de sauvegarde. C'est particulièrement vrai lorsque la mappe de sauvegarde comporte surtout des opérations de lecture. Les résultats des requêtes représentent alors un faible pourcentage des entrées de la mappe et seul un petit nombre d'index sont générés sur la mappe.

Planification de plusieurs topologies de centre de données

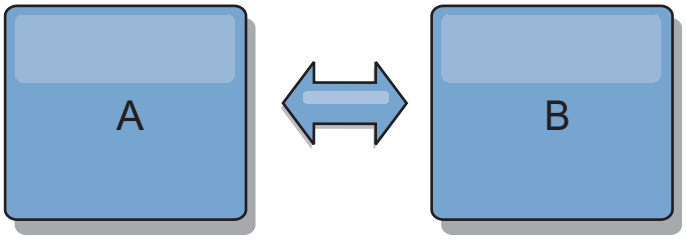
En utilisant la réplication asynchrone multimaître, au moins deux grilles de données peuvent devenir des copies exactes de l'une de l'autre. Chaque grille de données est hébergée dans un domaine de services de catalogue indépendant, avec ses propres de service de catalogue, serveurs de conteneur et un nom unique. Avec la réplication asynchrone multimaître, vous pouvez utiliser des liaisons pour connecter un ensemble de domaine de services de catalogue. Les domaine de services de catalogue sont ensuite synchronisés en utilisant la réplication via ces liaisons. Vous pouvez construire quasiment n'importe quelle topologie via la définition de liaisons entre les domaine de services de catalogue.

Topologies pour la réplication multimaître

Vous disposez de plusieurs options pour choisir la topologie de votre déploiement qui intègre la réplication multimaître.

Liaisons connectant des domaine de services de catalogue

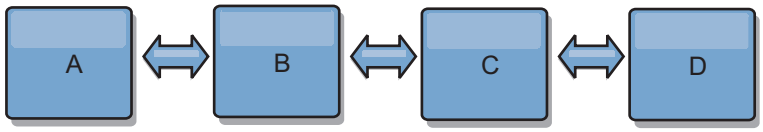
Une infrastructure de grilles de données de réplication est un graphique de domaine de services de catalogue interconnectés avec des liaisons bidirectionnelles. Avec une liaison, deux domaine de services de catalogue peuvent communiquer les modifications de données. Par exemple, la topologie la plus simple est une paire de domaine de services de catalogue avec une liaison unique entre eux. Les domaine de services de catalogue sont nommés par ordre alphabétique: A, B, C, etc., à partir de la gauche. Une liaison peut traverser un réseau WAN (wide area network) pour couvrir une grande distance. Même si la liaison est interrompue, vous pouvez toujours modifier les données dans l'un des domaine de services de catalogue. La topologie rapproche les modifications quand la liaison reconnecte les domaine de services de catalogue. Les liens tentent automatiquement de se reconnecter si la connexion réseau est interrompue.



Après avoir établi les liaisons, eXtreme Scale tente d'abord de rendre chaque domaine de services de catalogue identique. Ensuite, eXtreme Scale tente de maintenir identiques les conditions à mesure que des modifications se produisent dans un domaine de services de catalogue. L'objectif vise à faire de chaque domaine de services de catalogue le miroir exact d'un autre domaine de services de catalogue connecté par les liaisons. Les liaisons de réplication entre les domaine de services de catalogue permettent copier une modification effectuée dans un domaine vers les autres domaines.

Topologies linéaires

Même s'il s'agit d'un déploiement simple, une topologie linéaire montre certaines qualités des liaisons. Tout d'abord, il n'est pas nécessaire qu'un domaine de services de catalogue soit connecté directement à tous les autres domaine de services de catalogue pour pouvoir recevoir les modifications. Le domaine B extrait les modifications du domaine A. Le domaine C reçoit les modifications du domaine A via le domaine B, lequel connecte les domaines A et C. De même, le domaine D reçoit les modifications des autres domaines via le domaine C. De ce fait, la charge de la répartition des modifications est distribuée et elle n'incombe plus à la seule source de ces modifications.



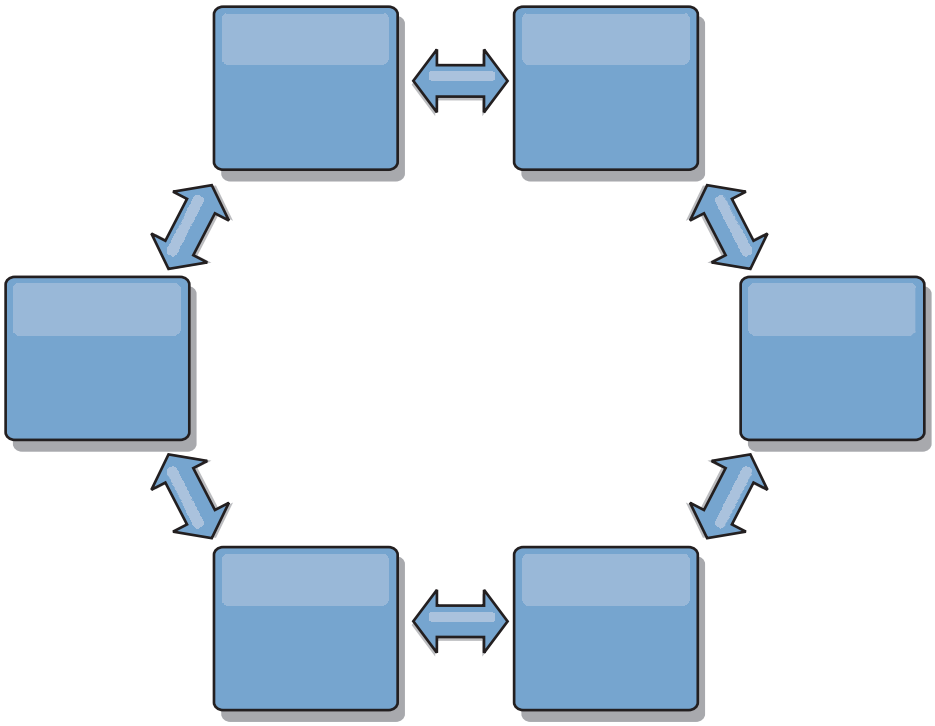
Notez que si le domaine C est défaillant, les actions suivantes se produisent :

1. Le domaine D serait orphelin jusqu'au redémarrage du domaine C.
2. Le domaine C doit se synchroniser avec le domaine B, lequel est une copie du domaine A.
3. Le domaine D utilise le domaine C pour se synchroniser avec les modifications des domaines A et B. Ces modifications se sont produites initialement lorsque le domaine D étaient orphelin (lorsque le domaine C était arrêté).

Enfin, les domaines A, B, C et D, sont de nouveau tous identiques.

Topologies en anneau

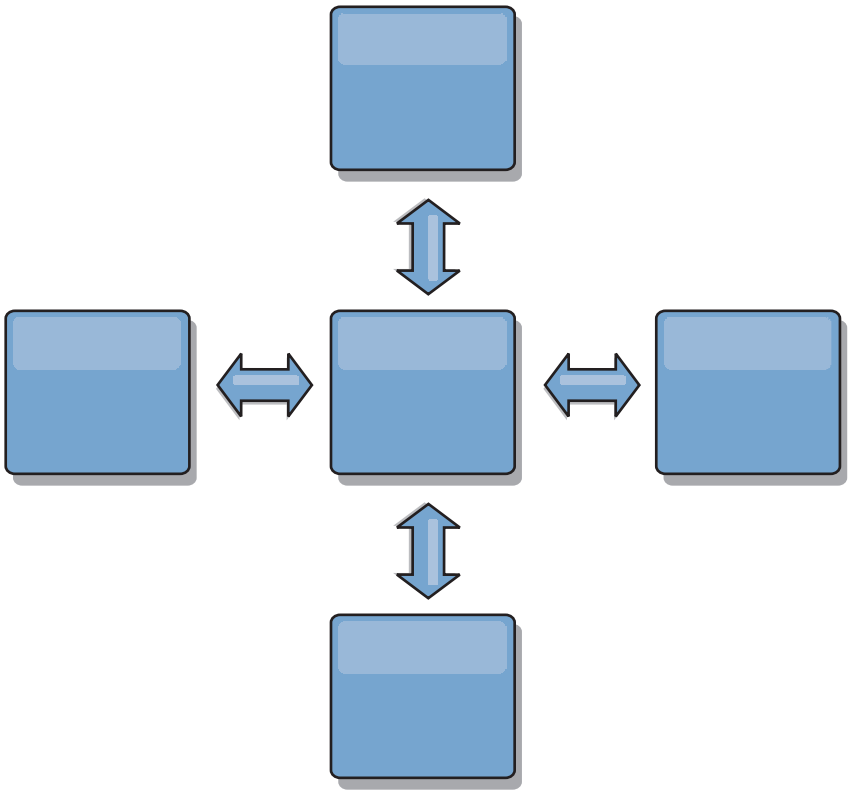
Les topologies en anneau sont un exemple de topologie encore plus résilientes. Lorsqu'un domaine de services de catalogue ou une liaison unique tombe en panne, les domaine de services de catalogue restants peuvent encore obtenir des modifications. Les domaine de services de catalogue parcourent l'anneau en s'éloignant de la défaillance. Chaque domaine de serveur de catalogue a au maximum deux liens vers d'autres domaine de services de catalogue, quelle que soit la taille de la topologie en anneau. Le délai de propagation des modifications peut être important. Les modifications d'un domaine de services de catalogue peuvent devoir traverser plusieurs liaisons pour que tous les domaine de services de catalogue aient les modifications. Une topologie linéaire a la même caractéristique.



Vous pouvez également déployer une topologie en anneau plus sophistiquée, avec un domaine de services de catalogue racine au centre de l'anneau. Le domaine de services de catalogue racine fait office de point central de réconciliation. Les autres domaine de services de catalogue font office de points distants de réconciliation pour les modifications se produisant dans le domaine de services de catalogue racine. Le domaine de services de catalogue racine peut arbitrer les modifications entre les domaine de services de catalogue. Si une topologie en anneau contient plusieurs anneaux autour d'un domaine de services de catalogue racine, le domaine ne peut pas arbitrer les modifications dans la partie interne de l'anneau. Toutefois, les résultats de l'arbitrage sont propagés dans les domaine de services de catalogue des autres anneaux.

Topologies en étoile

Avec une topologie en étoile, les modifications parcourent un domaine de services de catalogue en étoile. Etant donné que le concentrateur est le seul domaine de services de catalogue intermédiaire spécifié, les topologies en étoile ont une latence inférieure. Le domaine du concentrateur est connecté à chaque branche de domaine via une liaison. Le concentrateur distribue les modifications entre les domaine de services de catalogue. Il fait office de point de rapprochement pour les collisions. Dans un environnement soumis à une fréquence élevée de modifications, le concentrateur peut avoir besoin de s'exécuter sur plus de matériels que les branches pour rester synchronisé. WebSphere eXtreme Scale est conçu pour évoluer de manière linéaire, ce qui signifie que l'on peut, si nécessaire, étoffer le concentrateur sans difficultés. Toutefois, si le concentrateur tombe en panne, les modifications ne sont pas distribuées jusqu'à ce qu'il redémarre. Toutes les modifications sur les branches du sous-domaine de services de catalogue seront réparties après la reconnexion du concentrateur.



Vous pouvez également utiliser une stratégie avec les clients intégralement répliqués, une variante de la topologie qui utilise une paire de serveurs eXtreme Scale s'exécutant comme concentrateur. Chaque client crée une grille de données à conteneur unique autonome avec un catalogue dans la machine virtuelle Java client. Un client utilise sa grille de données pour se connecter au catalogue du concentrateur. Cette connexion provoque la synchronisation du client avec le concentrateur dès que le client obtient une connexion au concentrateur.

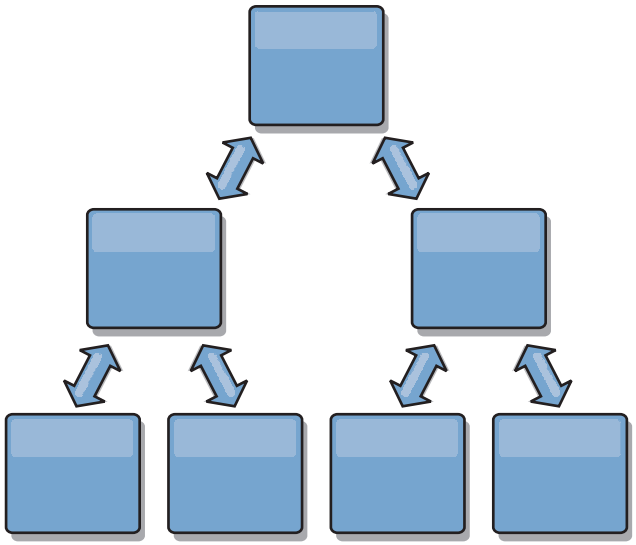
Toutes les modifications effectuées par le client sont locales pour le client et elles sont répliquées vers le concentrateur de manière asynchrone. Le concentrateur joue le rôle de domaine d'arbitrage, répartissant les modifications à tous les clients connectés. La topologie de clients intégralement répliqués fournit un cache L2 fiable pour un associateur relationnel d'objets comme OpenJPA. Les modifications sont réparties rapidement via le concentrateur entre les machines virtuelles client. Si la taille du cache peut être contenue dans le segment de mémoire disponible, la topologie est une architecture fiable pour ce style de cache L2.

Si nécessaire, utilisez plusieurs partitions pour échelonner le domaine concentrateur sur plusieurs machines virtuelles Java. Etant donné que toutes les données doivent toujours tenir sur une seule machine virtuelle Java client, plusieurs partitions augmentent la capacité du concentrateur à répartir et à arbitrer les modifications. Cependant, plusieurs partitions ne changent pas la capacité d'un domaine unique.

Topologies en arbre

Vous pouvez également utiliser un arbre dirigé acyclique. Un arbre acyclique n'a pas de cycles ou de boucles, et une configuration dirigée limite les liaisons aux parents et enfants existants uniquement. Cette configuration peut être utile pour les topologies disposant d'un grand nombre de domaine de services de catalogue, et il

n'est pas pratique d'avoir un concentrateur central connecté à chaque branche. Ce type de topologie peut également être utile lorsque vous devez ajouter des domaine de services de catalogue enfant sans mettre à jour le domaine de services de catalogue racine.



Une topologie en arbre peut toujours avoir un point central de rapprochement dans le domaine de services de catalogue racine. Le deuxième niveau peut toujours fonctionner en tant que point de rapprochement distant pour les modifications se produisant dans le domaine de services de catalogue en dessous. Le domaine de services de catalogue racine peut arbitrer les modifications entre les domaines de services de catalogue sur le deuxième niveau uniquement. Vous pouvez également utiliser des arbres n-aire ayant chacun n enfants à chaque niveau. Chaque domaine de services de catalogue se connecte à *n* liaisons.

Clients intégralement répliqués

Cette variante de la topologie implique une paire de serveurs eXtreme Scale s'exécutant comme concentrateur. Chaque client crée une grille de données à conteneur unique autonome avec un catalogue dans la machine virtuelle Java client. Un client utilise sa grille de données pour se connecter au catalogue du concentrateur, ce qui provoque la synchronisation du client avec le concentrateur dès que le client obtient une connexion au concentrateur.

Toutes les modifications effectuées par le client sont locales pour le client et elles sont répliquées vers le concentrateur de manière asynchrone. Le concentrateur joue le rôle de domaine d'arbitrage, répartissant les modifications à tous les clients connectés. La topologie de clients intégralement répliqués fournit un bon cache de niveau 2 pour un associateur relationnel d'objets comme OpenJPA. Les modifications sont réparties rapidement via le concentrateur entre les machines virtuelles client. Tant que la taille du cache peut être contenue dans l'espace de segment mémoire disponible des clients, cette topologie est une architecture tout à fait indiquée pour ce style de cache de niveau 2.

Si nécessaire, utilisez plusieurs partitions pour échelonner le domaine concentrateur sur plusieurs machines virtuelles Java. Toutes les données devant tenir sur une seule machine virtuelle Java, l'utilisation de partitions multiples augmente la capacité du concentrateur à répartir et à arbitrer les modifications, mais elle ne change pas la capacité d'un domaine unique.

Considérations de configuration pour les topologies multimaîtres

Tenez compte des points suivants lorsque vous déterminez l'opportunité et la manière d'utiliser des topologies de réplication multimaîtres.

- **Exigences de groupe de mappes**

  Les groupes de mappes doivent avoir les caractéristiques suivantes pour pouvoir répliquer les modifications dans les liaisons d'un domaine de services de catalogue :

  – Le nom ObjectGrid et le nom de groupe de mappes dans un domaine de services de catalogue doivent correspondre au nom ObjectGrid et au nom de groupe de mappes d'autres domaine de services de catalogue. Par exemple, ObjectGrid "og1" et le groupe de mappes "ms1" doivent être configurés dans les domaine de services de catalogue A et B pour pouvoir répliquer les données dans la mappe entre les domaine de services de catalogue.

  – Est une grille de données `FIXED_PARTITION`. Les grilles de données `PER_CONTAINER` ne peuvent pas être répliquées.

  – A le même nombre de partitions dans chaque domaine de services de catalogue. Le groupe de mappes peut ou peut ne pas avoir le même nombre et le même type de répliques.

  – A les mêmes types de données répliquées dans chaque domaine de services de catalogue.

  – Contient les mêmes mappes et modèles de mappes dynamiques dans chaque domaine de services de catalogue.

  – N'utilise pas le gestionnaire d'entités. Un groupe de mappes contenant une mappe d'entités n'est pas répliqué entre les domaine de services de catalogue.

  – N'utilise pas la mise en cache en écriture différée. Un groupe de mappes contenant une mappe qui est configurée avec la prise en charge de l'écriture différée n'est pas répliqué entre les domaine de services de catalogue.

  Tous les ensembles de mappes ayant les caractéristiques ci-dessus commencent à répliquer après que les domaine de services de catalogue dans la topologie ont été démarrés.

- **Chargeurs de classe avec plusieurs domaine de services de catalogue**

  Les domaine de services de catalogue doivent avoir accès à toutes les classes qui sont utilisées comme clés et valeurs Toutes les dépendances doivent être reflétées dans tous les chemins d'accès aux classes des machines virtuelles Java (JVM) de conteneur de la grille de données de tous les domaines. Si un plug-in CollisionArbiter extrait la valeur d'une entrée de cache, les classes correspondant aux valeurs doivent être présentes pour le domaine qui démarre l'arbitre.

Remarques sur les chargeurs dans une topologie multimaître

Lorsque vous utilisez des chargeurs dans une topologie multimaître, vous devez envisager les problèmes éventuels de collision et de maintenance des informations de révision. La grille de données conserve les informations de révision sur les éléments de façon à ce que les collisions puissent être détectées lorsque d'autres fragments primaires dans la configuration y écrivent des entrées. Lorsque des entrées sont ajoutées à partir d'un chargeur, ces informations de révision ne sont pas incluses et l'entrée prend une nouvelle révision. Etant donné que la révision de l'entrée semble être une nouvelle insertion, une fausse collision peut se produire si un autre fragment primaire modifie également cet état ou insère les mêmes informations à partir d'un chargeur.

Les modifications de réplication appellent la méthode get sur le chargeur avec la liste des clés qui ne sont pas déjà dans la grille de données, mais qui vont être modifiées lors de la transaction de réplication. Lorsque la réplication se produit,

ces entrées sont des entrées de collision. Lorsque les collisions sont arbitrées et que la révision est appliquée, une mise à jour par lots est appelée sur le chargeur pour appliquer les modifications à la base de données. Toutes les mappes qui ont été modifiées dans la fenêtre de révision sont mises à jour dans la même transaction.

L'énigme de préchargement

Supposons une topologie avec les deux centres de données A et B qui ont des bases de données indépendantes, mais seul le centre de données A a une grille active. Lorsque vous établissez une liaison entre les centres de données pour une configuration multimaître, les grilles de données dans le centre de données A commencent à envoyer les données aux nouvelles grilles dans le centre de données B, ce qui crée une collision avec chaque entrée. Un autre problème est l'existence de données dans la base de données du centre de données B, mais qui ne figurent pas dans la base de données du centre de données A. Ces lignes ne sont pas remplies et arbitrées, ce qui génère des incohérences qui ne sont pas résolues.

Solution de l'énigme de préchargement

Etant donné que les données qui se trouvent uniquement dans la base de données ne peuvent pas comporter des révisions, vous devez toujours précharger complètement la grille de données à partir de la base de données locale pour établir la liaison multimaître. Ensuite, les deux grilles de données peuvent réviser et arbitrer les données, pour atteindre finalement un état cohérent.

L'énigme du cache partiel

Avec un cache partiel, la première application tente de trouver des données dans la grille de données. Si les données ne sont pas dans la grille de données, elles sont recherchées dans la base de données à l'aide du chargeur. Les entrées sont supprimées de la grille de données régulièrement pour maintenir une mémoire cache de petite taille.

Ce type de mémoire cache peut être problématique dans un scénario de configuration multimaître, car les entrées dans la grille de données ont des métadonnées de révision qui permettent de détecter quand des collisions se produisent et de déterminer qui a effectué les modifications. Lorsque des liaisons entre les centres de données ne fonctionnent pas, un centre de données peut mettre à jour une entrée et ensuite éventuellement mettre à jour la base de données et invalider l'entrée dans la grille de données. Lorsque la liaison est rétablie, les centres de données tentent de synchroniser les révisions les unes par rapport aux autres. Toutefois, étant donné que la base de données a été mise à jour et que l'entrée de la grille de données a été invalidée, la modification est perdue du point de vue du centre de données qui s'est arrêté. En conséquence, les deux côtés de la grille de données sont désynchronisés et ne sont pas cohérents.

Solution de l'énigme de cache partiel

Topologie en étoile :

Vous pouvez exécuter le chargeur uniquement dans la topologie en étoile pour maintenir la cohérence des données lors de l'extension de la grille de données. Toutefois, si vous envisagez ce déploiement, notez que les chargeurs peuvent permettre à la grille de données d'être partiellement chargée, ce qui implique qu'un expulseur a été configuré. Si les rayons de la configuration sont des caches partiels, les échecs en mémoire cache n'ont aucun moyen d'extraire des données de la base

de données. En raison de cette restriction, vous devez utiliser une topologie de cache complètement remplie avec une configuration en étoile.

Invalidations et expulsion

L'invalidation crée des incohérences entre la grille de données et la base de données. Les données peuvent être supprimées de la grille de données, à l'aide d'un programme ou par l'expulsion. Lorsque vous développez votre application, sachez que le traitement des révisions ne réplique pas les modifications invalidées, ce qui provoque des incohérences entre les fragments primaires.

Les événements d'invalidation ne sont pas des modifications de l'état du cache et n'entraînent pas de réplication. Tous les expulseurs configurés s'exécutent indépendamment des autres expulseurs dans la configuration. Par exemple, vous pouvez avoir un expulseur configuré pour un seuil de mémoire dans un domaine de services de catalogue, mais un type d'expulseur différent moins agressif dans l'autre domaine de services de catalogue lié. Lorsque des entrées de grille de données sont supprimées en raison de la règle de seuil de mémoire, les entrées dans l'autre domaine de services de catalogue ne sont pas affectées.

Mises à jour de la base de données et invalidation de la grille de données

Des problèmes se produisent lorsque vous mettez à jour la base de données directement en arrière-plan lors de l'appel de l'invalidation dans la grille de données pour les entrées mises à jour dans une configuration multimaître. Ce problème se produit, car la grille de données ne peut pas répliquer la modifications dans les autres fragments primaires jusqu'à ce qu'un accès de cache transfère l'entrée vers la grille de données.

Plusieurs programmes d'écriture dans une seule base de données logique

Lorsque vous utilisez une seule base de données avec plusieurs fragments primaires qui sont connectés par l'intermédiaire d'un chargeur, des conflits transactionnels se produisent. Votre implémentation de chargeur doit gérer ces types de scénarios.

Mise en miroir des données à l'aide de la réplication multimaître

Vous pouvez configurer des bases de données indépendantes qui sont connectées à des domaine de services de catalogue indépendants. Dans cette configuration, le chargeur peut envoyer les modifications d'un centre de données vers un autre.

Considérations de conception pour la réplication multimaître

Lors de l'implémentation de la réplication multimaître, vous devez tenir compte de divers éléments dans votre conception, tels que l'arbitrage, les liaisons et les performances.

Points concernant l'arbitrage à prendre en considération dans la conception des topologies

Des collisions entre des modifications peuvent se produire s'il est possible à des enregistrements identiques d'être modifiés simultanément en deux endroits différents. Configurez chaque domaine de services de catalogue pour que les domaines aient le même nombre de processeurs, la même quantité de mémoire et le même nombre de ressources réseau. Vous remarquerez sans doute que des

domaine de services de catalogue d'exécution gérant les collisions de modifications (arbitrage) utilisent plus de ressources que d'autres domaine de services de catalogue. Les collisions sont détectées de manière automatique. Elles sont traitées avec l'un des deux mécanismes suivants :

- **Arbitre par défaut** : le protocole par défaut doit utiliser les modifications du domaine de services de catalogue occupant la position la moins basse alphabétiquement. Par exemple, si les domaine de services de catalogue A et B génèrent un conflit pour un enregistrement, la modification du domaine de services de catalogue B est ignorée. Le domaine de services de catalogue A conserve sa version et l'enregistrement dans le domaine de services de catalogue B est modifié pour qu'il corresponde à l'enregistrement du domaine de services de catalogue A. Ce comportement s'applique également aux applications où les utilisateurs ou les sessions sont normalement liés ou ont une affinité à l'une des grilles de données.
- **Arbitre personnalisé** : les applications peuvent fournir un arbitre personnalisé. Lorsqu'un domaine de services de catalogue détecte une collision, il démarre l'arbitre. Pour plus d'informations sur le développement d'un arbitre personnalisé utile, voir Développement d'arbitres personnalisés pour la réplication multi-maître.

Pour les topologies dans lesquelles les collisions sont possibles, songez à implémenter une topologie en étoile ou en arbre. Les deux topologies sont propices à éviter les collisions constantes, ce qui peut se produire dans les scénarios suivants :

1. Plusieurs domaine de services de catalogue sont affectés par une collision.
2. Chaque domaine de services de catalogue gère la collision en local, ce qui produit des révisions.
3. Les révisions entrent en collision, d'où des révisions de révisions.

Pour éviter les collisions, choisissez un domaine de services de catalogue spécifique, appelé *domaine de services de catalogue d'arbitrage* comme arbitre des collisions d'un sous-ensemble de domaine de services de catalogue. Par exemple, une topologie en étoile pourra utiliser le concentrateur comme gestionnaire de collisions. Le gestionnaire de collisions ignore toutes les collisions qui sont détectées par les sous-domaine de services de catalogue. Le domaine de services de catalogue du concentrateur crée des révisions, empêchant les révisions de collisions inattendues. Le domaine de services de catalogue qui est affecté à la gestion des collisions doit se lier à tous les domaines dont il est chargé de traiter les collisions. Dans une topologie en arbre, tous les domaines parent internes traitent les collisions pour leurs enfants immédiats. En revanche, si vous utilisez une topologie en anneau, vous ne pouvez pas désigner un domaine de services de catalogue dans le fichier comme arbitre.

Le tableau qui suit récapitule les approches en matière d'arbitrage qui sont les plus compatibles avec les diverses topologies.

Tableau 1. Approches en matière d'arbitrage. Ce tableau énonce si l'arbitrage entre applications est compatible avec les diverses topologies.

Topologie	Arbitrage d'application	Notes
Ligne de deux domaine de services de catalogue	Oui	Choisissez un domaine de services de catalogue comme arbitre.

Tableau 1. Approches en matière d'arbitrage (suite). Ce tableau énonce si l'arbitrage entre applications est compatible avec les diverses topologies.

Topologie	Arbitrage d'application	Notes
Ligne de trois domaine de services de catalogue	Oui	Le domaine de services de catalogue du milieu doit être l'arbitre. Assimilez ce domaine de services de catalogue au concentrateur dans une topologie en étoile simple.
Ligne de plus de trois domaine de services de catalogue	Non	L'arbitrage d'application n'est pas pris en charge.
Concentrateur avec n "rayons"	Oui	Le concentrateur avec des liens vers toutes les branches doit être le domaine de services de catalogue d'arbitrage.
Anneau de N domaine de services de catalogue	Non	L'arbitrage d'application n'est pas pris en charge.
Arbre dirigé acyclique (arbre n-aire)	Oui	Tous les noeuds racine doivent évaluer leurs descendants directs uniquement.

Points concernant les liens à prendre en considération dans la conception des topologies

Dans l'idéal, une topologie comprend le minimum de liens tout en optimisant les compromis entre les temps d'attente des modifications, la tolérance aux pannes et les caractéristiques de performances.

- **Temps d'attente des modifications**

  Le temps d'attente de modification est déterminé par le nombre de domaine de services de catalogue intermédiaires par lequel un changement doit passer avant d'arriver à un domaine de services de catalogue spécifique.

  Une topologie a le meilleur temps d'attente lorsqu'elle élimine les domaine de services de catalogue intermédiaires en liant chacun des domaine de services de catalogue à chacun des autres domaine de services de catalogue. Toutefois, un domaine de services de catalogue doit effectuer la réplication par rapport à son nombre de liens. Pour les topologies de grande taille, le nombre de liens à définir peut entraîner une charge administrative.

  La vitesse à laquelle une modification est copiée vers les autres domaine de services de catalogue dépend de facteurs supplémentaires, tels que :

  – Bande passante du processeur et du réseau dans le domaine de services de catalogue source
  – Nombre de domaine de services de catalogue intermédiaire et de liens entre la source et la cible du domaine de services de catalogue source et cible
  – Ressources en processeur et en réseau disponibles pour les domaine de services de catalogue source, cible et intermédiaires

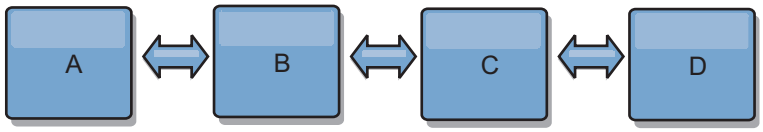
- **Tolérance aux pannes**

  La tolérance aux pannes est déterminée par le nombre de chemins existant entre deux domaine de services de catalogue pour la réplication des modifications.

  Si vous ne disposez que d'un seul lien entre une paire de domaine de services de catalogue, une défaillance de lien empêche la propagation des modifications. De même, les modifications ne sont pas propagées entre les domaine de services de catalogue si un incident de liaison se produit sur les domaines intermédiaires.

Votre topologie pourrait avoir un lien unique d'un domaine de services de catalogue vers un autre de sorte que le lien passe par des domaines intermédiaires. Dans ce cas, les modifications ne sont pas propagées si l'un des domaine de services de catalogue intermédiaires est défaillant.

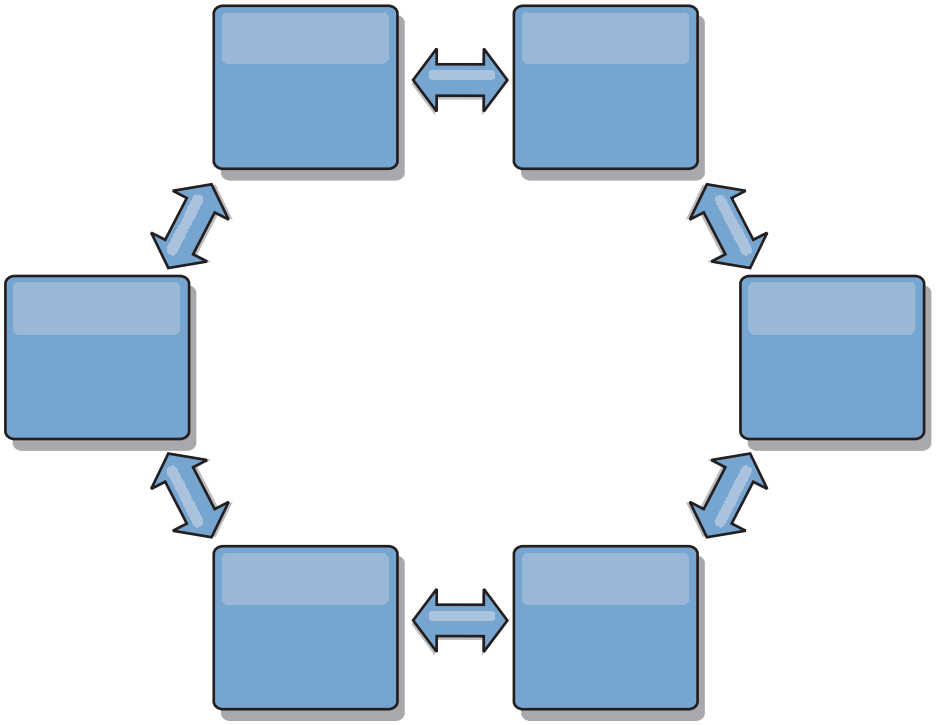
Supposons la topologie linéaire à quatre domaine de services de catalogue A, B, C et D :



Si l'une de ces conditions existe, le domaine D ne voit pas les modification de A :

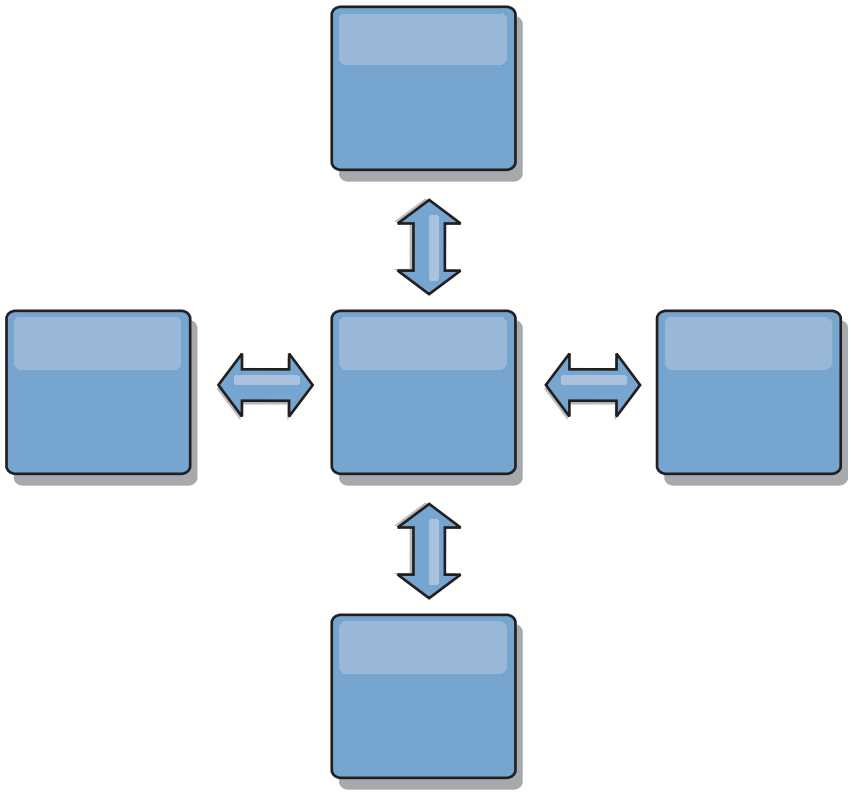
- Le domaine A est actif et B est arrêté.
- Les domaines A et B sont actifs et C est arrêté.
- Le lien entre A et B ne fonctionne pas.
- Le lien entre B et C ne fonctionne pas.
- Le lien entre C et D est arrêté.

En revanche, avec une topologie en anneau, chaque domaine de services de catalogue peut recevoir les modifications dans un sens ou dans l'autre.



Par exemple, si un service de catalogue donné de votre topologie en anneau est arrêté, les deux domaines contigus peuvent toujours extraire les modifications directement de l'autre.

Toutes les modifications sont propagées via le concentrateur. Par conséquent, contrairement aux topologies linéaires et en anneau, la conception en étoile peut tomber en panne si le concentrateur est défaillant.



Un domaine de services de catalogue unique est résilient à un certain degré de perte de service. Cependant, les incidents les plus importants, tels que les indisponibilités de réseau étendu ou les pertes de liaisons entre les centres de données physiques peuvent perturber les domaine de services de catalogue.

- **Liaison et performances**

  Le nombre de liaisons définies sur un domaine le service de catalogue affecte les performances. Un plus grand nombre de liaisons utilisent davantage de ressources et les performances de réplication peuvent baisser. La possibilité d'extraire les modifications pour un domaine A via d'autres domaines empêche le domaine A de répliquer ses transactions partout. La charge de la répartition des modifications dans un domaine est limitée par le nombre de liaisons qu'il utilise et non pas par le nombre de domaines dans la topologie. Cette propriété est synonyme d'évolutivité et les domaines de la topologie peuvent partager la charge de la répartition des modifications.

  Un domaine de services de catalogue peut extraire les modifications indirectement via d'autres domaine de services de catalogue. Supposons une topologie linéaire avec cinq domaine de services de catalogue.

  ```
  A <=> B <=> C <=> D <=> E
  ```

  – A extrait les modifications de B, C, D, et E via B
  – B extrait les modifications directement de A et de C et les modifications de D et de E via C
  – C extrait les modifications directement de B et de D et les modifications de A via B et de E via D
  – D extrait les modifications directement de C et de E et les modifications de A et de B via C
  – E extrait les modifications directement de D et les modifications de A, B et C via D

La charge de la répartition dans les domaine de services de catalogue A et E est la plus faible, car ils ont chacun une seule liaison à un domaine de services de catalogue unique. Les domaines B, C et D ont chacun une liaison avec deux domaines. Par conséquent, la charge de la répartition dans les domaines B, C, et D est le double de celle des domaines A et E. La charge de travail dépend du nombre de liaisons dans chaque domaine et non pas du nombre total de domaines dans la topologie. Par conséquent, la répartition de charge décrite demeurerait constante, même si la ligne contenait 1 000 domaines.

Considérations relatives aux performances de réplication multimaître

Tenez compte des limitations suivantes lorsque vous utilisez des topologies de réplication multimaître :

- **Optimisation de la répartition des modifications**, comme expliquée dans la section précédente.
- **Performances des liens de réplication** WebSphere eXtreme Scale crée un seul socket TCP/IP entre n'importe quelle paire de machines virtuelles Java. Tout le trafic entre les machines virtuelles Java passe par le socket unique, y compris le trafic de la réplication multimaître. Les domaine de services de catalogue sont hébergés dans au moins *n* machines virtuelles Java pour fournir au minimum *n* liaisons TCP aux domaine de services homologues. Ainsi, les domaine de services de catalogue avec un plus grand nombre de conteneurs offrent de meilleures performances de réplication. Un plus grand nombre de conteneurs requiert davantage de processeurs et de ressources réseau.
- **Le support de l'optimisation de la fenêtre dynamique TCP et RFC 1323** RFC 1323 à chaque extrémité d'une liaison renvoie plus de données pour un aller-retour. Ce support augmente le débit en développant la capacité de la fenêtre d'un facteur d'environ 16 000.

  Notez que les sockets TCP utilisent un mécanisme de fenêtre dynamique pour contrôler le flux des données en vrac. Ce mécanisme limite généralement le socket à 64 Ko pour un intervalle d'aller-retour. Si l'intervalle aller-retour est de 100 ms, la bande passante est limitée à 640 Ko/s sans optimisation supplémentaire. L'utilisation intégrale de la bande passante disponible sur un lien peut nécessiter une optimisation qui est spécifique au système d'exploitation. La plupart des systèmes d'exploitation comportent des paramètres d'optimisation, y compris des options RFC 1323, permettant d'améliorer le débit sur les liaisons à forte latence.

  Plusieurs facteurs peuvent affecter les performances de la réplication :

  – Vitesse d'extraction des modifications par eXtreme Scale.
  – Vitesse à laquelle eXtreme Scale peut traiter les demandes de réplication d'extraction.
  – Capacité de la fenêtre dynamique.
  – Avec l'optimisation de la mémoire tampon réseau aux deux extrémités d'une liaison, eXtreme Scale extrait les modification sur le socket de manière efficace.
- **Sérialisation des objets** Toutes les données doivent être sérialisables. Si un domaine de services de catalogue n'utilise pas `COPY_TO_BYTES`, il doit utiliser la sérialisation Java ou ObjectTransformers pour optimiser les performances de sérialisation.
- Par défaut **la compression** WebSphere eXtreme Scale compresse toutes les données envoyées entre les domaines de services de catalogue. Vous ne pouvez désactiver actuellement la compression.

- **Optimisation de la mémoire** L'utilisation de la mémoire pour une topologie de réplication multimaître est largement indépendante du nombre de domaine de services de catalogue dans la topologie.

  La réplication multimaître ajoute un temps de traitement fixe par entrée de mappe pour la gestion des versions. Chaque conteneur suit également une quantité fixe de données pour chaque domaine de services de catalogue dans la topologie. Une topologie à deux domaines de services de catalogue utilise approximativement la même quantité de mémoire qu'une topologie à 50 domaine de services de catalogue. WebSphere eXtreme Scale n'utilise pas de journaux de relecture ou de files d'attente similaires dans son implémentation. Ainsi, aucune structure de récupération n'est prête si la liaison de réplication n'est pas disponible pendant un certain temps et redémarre ensuite.

Interopérabilité avec d'autres produits WebSphere

WebSphere eXtreme Scale peut être intégré à d'autres produits de serveurs, comme WebSphere Application Server et WebSphere Application Server Community Edition.

WebSphere Application Server

Vous pouvez intégrer WebSphere Application Server à divers éléments de votre configuration WebSphere eXtreme Scale. Vous pouvez déployer des applications de grille de données et utiliser WebSphere Application Server pour héberger les serveurs de conteneur et de catalogue. Vous pouvez également utiliser la sécurité WebSphere Application Server dans votre environnement WebSphere eXtreme Scale.

WebSphere Portal

Vous pouvez rendre persistantes des sessions HTTP depuis WebSphere Portal dans une grille de données dans WebSphere eXtreme Scale.

WebSphere Application Server Community Edition

WebSphere Application Server Community Edition peut partager l'état des sessions, mais d'une manière peu efficace et non évolutive. WebSphere eXtreme Scale fournit une couche de persistance répartie à hautes performances qui peut servir à répliquer l'état mais sans s'intégrer facilement aux autres serveurs d'applications extérieurs à WebSphere Application Server. Vous pouvez intégrer ces deux produits pour offrir une solution de gestion de session évolutive.

WebSphere Real Time

Avec le support pour WebSphere Real Time, l'offre Java temps réel la plus efficace, WebSphere eXtreme Scale permet aux applications Extreme Transaction Processing (XTP) d'avoir des temps de réponse plus cohérents et plus prévisibles.

Planification pour l'installation

Avant d'installer le produit, vous devez tenir compte de votre environnement.

Configurations matérielle et logicielle requises

Vue d'ensemble des conditions requises en termes de matériels et de systèmes d'exploitation. Bien que vous ne soyez pas tenu d'utiliser un niveau spécifique de

matériel ou de système d'exploitation pour WebSphere eXtreme Scale, nous n'en fournissons pas moins sur le site de support du produit (page Configuration requise) une liste détaillée des matériels et logiciels officiellement pris en charge. En cas de conflit entre les informations présentées par le Centre de documentation et celles figurant sur cette page, les informations fournies par le site Web prévalent. Les conditions préalables répertoriées par le Centre de documentation sont fournies à titre informatif uniquement.

Voir la page System Configurations requises pour connaître les configurations matérielles et logicielles officielles.

Vous n'êtes pas obligé d'installer et de déployer eXtreme Scale sur un niveau de système d'exploitation spécifique. Chaque installation Java Platform, Standard Edition (Java SE) et Java Platform, Enterprise Edition (Java EE) requiert des niveaux de système d'exploitation ou des correctifs différents.

Vous pouvez installer et déployer le produit dans les environnements Java EE et Java SE. Vous pouvez également regrouper le composant client avec les applications Java EE directement sans les intégrer à WebSphere Application Server. WebSphere eXtreme Scale prend en charge Java SE 5 et les versions suivantes et WebSphere Application Server Version 6.1 et les versions suivantes.

Configuration matérielle

WebSphere eXtreme Scale ne requiert pas la présence d'un niveau spécifique de matériel. La configuration matérielle requise dépend du matériel pris en charge pour l'installation de Java Platform, Standard Edition que vous utilisez pour exécuter WebSphere eXtreme Scale. Si vous utilisez eXtreme Scale avec WebSphere Application Server ou une autre implémentation Java Platform, Enterprise Edition, la configuration matérielle requise par ces plateformes est suffisante pour WebSphere eXtreme Scale.

Configuration requise en matière de système d'exploitation

- **Sans la console Web**

  eXtreme Scale ne requiert pas la présence d'un système d'exploitation d'un niveau donné. Chaque implémentation Java SE et Java EE requiert un niveau différent du système d'exploitation ou des correctifs pour les problèmes identifiés lors du test de l'implémentation Java. Les niveaux nécessaires à ces implémentations sont suffisants pour eXtreme Scale.
- **Avec la console Web**

  Les conditions suivantes s'appliquent pour chaque système d'exploitation si vous utilisez la console :
  - Linux : JMV 32 bits ou 64 bits
  - Linux PPC : JVM 32 bits uniquement
  - Windows : JVM 32 bits uniquement
  - AIX : JVM 32 bits uniquement

Navigateurs Web requis

La console Web prend en charge les navigateurs Web suivants :

- Mozilla Firefox, version 3.5.x et versions ultérieures
- Mozilla Firefox, version 3.6.x et versions ultérieures
- Microsoft Internet Explorer version 7 ou 8

Configuration requise pour WebSphere Application Server

- WebSphere Application Server Version 6.1.0.39 ou version suivante
- WebSphere Application Server Version 7.0.0.19 ou version suivante
- WebSphere Application Server Version 8.0.0.1 ou version suivante

Pour plus d'informations, consultez la section Recommended fixes for WebSphere Application Server.

Autres conditions requises par le serveur d'applications

Les autres implémentations Java EE peuvent utiliser la phase d'exécution d'eXtreme Scale en tant qu'instance locale ou client pour les serveurs eXtreme Scale. Pour implémenter Java SE, vous devez utiliser la version 5 ou une version suivante.

Java SE : points à prendre en considération

WebSphere eXtreme Scale nécessite Java SE 5 ou une version suivante. En règle générale, les nouvelles versions de Java SE ont des fonctions plus efficaces et sont plus performantes.

Versions prises en charge

Vous pouvez utiliser WebSphere eXtreme Scale avec Java SE 5 et les versions suivantes. La version que vous utilisez doit être actuellement pris en charge par le fournisseur de l'environnement JRE (Java Runtime Environmen).

Un environnement JRE entièrement pris en charge est installé avec les installations autonomes WebSphere eXtreme Scale et WebSphere eXtreme Scale Client dans le répertoire *racine_install_wxs*/java et peut être utilisé par les clients et les serveurs. Si vous installez WebSphere eXtreme Scale dans WebSphere Application Server, vous pouvez utiliser l'environnement JRE inclus dans l'installation WebSphere Application Server.

WebSphere eXtreme Scale tire parti de la fonctionnalité Java Development Kit (JDK) 5 ou d'une version suivante lorsqu'elle devient disponible. Généralement, les nouvelles versions Java Development Kit (JDK) et Java SE sont plus performantes et ont une fonctionnalité plus efficace.

Voir Logiciels pris en charge pour plus d'informations.

Fonctions WebSphere eXtreme Scale dépendantes de Java

Tableau 2. Fonctions nécessitant Java SE 5 ou Java SE 6.

WebSphere eXtreme Scale utilise la fonctionnalité introduite dans Java SE 5 ou Java SE 6 pour fournir les fonctions suivantes du produit.

Caractéristique	Pris en charge dans Java SE 5 et les versions suivantes	Pris en charge dans Java SE 6 et les versions suivantes
Annotations d'API EntityManager (facultatif : vous pouvez également utiliser des fichiers XML)	X	X

Tableau 2. Fonctions nécessitant Java SE 5 ou Java SE 6 (suite).

WebSphere eXtreme Scale utilise la fonctionnalité introduite dans Java SE 5 ou Java SE 6 pour fournir les fonctions suivantes du produit.

Caractéristique	Pris en charge dans Java SE 5 et les versions suivantes	Pris en charge dans Java SE 6 et les versions suivantes
Java Persistence API (JPA) : programme de chargement JPA, programme de chargement de client JPA et programme de mise à jour en fonction du temps JPA	X	X
L'expulsion basée sur la mémoire (utilise MemoryPoolMXBean)	X	X
Agents d'instrumentation : • `wxssizeagent.jar` : augmente la précision des mesures de mappe d'octets utilisées. • `ogagent.jar` : augmente la performance des entités d'accès aux zones.	X	X
Console Web de surveillance		X

Java EE : points à prendre en considération

Lors de la préparation de l'intégration WebSphere eXtreme Scale dans un environnement Java Platform, Enterprise Edition, tenez compte de certains éléments, tels que les versions, les options de configuration, les conditions requises et les limitations, le déploiement et la gestion des applications.

Exécuter des applications eXtreme Scale en environnement Java EE

Une application Java EE peut se connecter à une application eXtreme Scale distante. En outre, l'environnement WebSphere Application Server permet le démarrage d'un serveur eXtreme Scale lorsqu'une application démarre dans le serveur d'applications.

Si vous utilisez un fichier XML pour créer une instance ObjectGrid et que ce fichier XML se trouve dans le module du fichier EAR, accédez à ce fichier à l'aide de la méthode getClass().getClassLoader().getResource("META-INF/objGrid.xml") afin d'obtenir un objet URL permettant de créer une instance ObjectGrid. Dans l'appel à la méthode, remplacez le nom du fichier XML utilisé.

Vous pouvez utiliser des beans de démarrage pour que, à son démarrage, une application amorce une instance ObjectGrid et supprime cette instance lorsqu'elle s'arrête. Un bean de démarrage est un bean de session sans état avec un emplacement distant com.ibm.websphere.startupservice.AppStartUpHome et une interface distante com.ibm.websphere.startupservice.AppStartUp. L'interface distante possède deux méthodes : la méthode start et la méthode stop. Utilisez la méthode start pour amorcer l'instance et la méthode stop pour détruire l'instance. L'application utilise la méthode ObjectGridManager.getObjectGrid pour maintenir la référence à cette instance. Voir les informations relatives à l'accès à un objet

ObjectGrid avec ObjectGridManager dans *Guide de programmation* pour plus d'informations.

Utiliser des loaders de classes

Lorsque les modules d'application qui utilisent des chargeurs de classe différents partagent une instance ObjectGrid unique dans une application Java EE, vérifiez que les objets qui sont stockés dans eXtreme Scale et que les plug-ins du produit se trouvent dans un chargeur commun dans l'application.

Gérer dans un servlet le cycle de vie des instances ObjectGrid

Pour gérer le cycle de vie d'une instance ObjectGrid dans un servlet, vous pouvez utiliser la méthode init pour créer l'instance et la méthode destroy pour supprimer l'instance. Si l'instance est mise en cache, elle est extraite et manipulée dans le code du servlet. Voir les informations relatives à l'accès à un objet ObjectGrid avec l'interface ObjectGridManager dans *Guide de programmation* pour plus d'informations.

Conventions relatives aux répertoires

Les conventions de répertoire suivantes sont utilisées dans toute la documentation pour faire référence à des répertoires spéciaux, tels que *wxs_install_root* et *wxs_home*. Vous pouvez accéder à ces répertoires pendant plusieurs scénarios différents, y compris lors de l'installation et de l'utilisation des outils de ligne de commande.

racine_install_wxs
Le répertoire *wxs_install_root* est le répertoire racine où sont installés les fichiers du produit WebSphere eXtreme Scale. Le répertoire *wxs_install_root* peut être le répertoire dans lequel l'archive d'évaluation est extraite ou depuis lequel le produit est installé WebSphere eXtreme Scale.

- Exemple où la version d'essai a été extraite :

  Exemple : `/opt/IBM/WebSphere/eXtremeScale`
- Exemple où WebSphere eXtreme Scale est installé dans un répertoire autonome :

  Exemple : `/opt/IBM/eXtremeScale`
- Exemple lorsque WebSphere eXtreme Scale est intégré à WebSphere Application Server :

  Exemple : `/opt/IBM/WebSphere/AppServer`

wxs_home
Le répertoire *wxs_home* est le répertoire racine du produit, des bibliothèques, des exemples et des composants WebSphere eXtreme Scale. Ce répertoire est identique au répertoire *wxs_install_root* lorsque l'archive d'évaluation est extraite. Pour les installations autonomes, le répertoire *wxs_home* est le sous-répertoire `ObjectGrid` du répertoire *wxs_install_root*. Pour les installations qui sont intégrées à WebSphere Application Server, ce répertoire est le répertoire `optionalLibraries/ObjectGrid` du répertoire *wxs_install_root*.

- Exemple lorsque la version d'essai a été extraite :

  Exemple : `/opt/IBM/WebSphere/eXtremeScale`
- Exemple lorsque WebSphere eXtreme Scale est installé dans un répertoire autonome :

  Exemple : `/opt/IBM/eXtremeScale/ObjectGrid`

- Exemple lorsque WebSphere eXtreme Scale est intégré à WebSphere Application Server :

  Exemple : `/opt/IBM/WebSphere/AppServer/optionalLibraries/ObjectGrid`

was_root
Le répertoire *was_root* est le répertoire racine d'une installation WebSphere Application Server :

Exemple : `/opt/IBM/WebSphere/AppServer`

restservice_home
Le répertoire *restservice_home* est le répertoire dans lequel se trouvent les bibliothèques et les exemples du service de données REST d'WebSphere eXtreme Scale. Ce répertoire s'appelle `restservice` et il est le sous-répertoire de *wxs_home*.

- Exemple pour les déploiements autonomes :

  Exemple : `/opt/IBM/WebSphere/eXtremeScale/ObjectGrid/restservice`

- Exemple pour les déploiements intégrés à WebSphere Application Server :

  Exemple : `/opt/IBM/WebSphere/AppServer/optionalLibraries/ObjectGrid/restservice`

tomcat_root
Le répertoire *home_tomcat* est le répertoire racine de l'installation d'Apache Tomcat.

Exemple : `/opt/tomcat5.5`

wasce_root
wasce_root est le répertoire racine de l'installation WebSphere Application Server Community Edition.

Exemple : `/opt/IBM/WebSphere/AppServerCE`

java_home
Le répertoire *java_home* est le répertoire racine d'une installation de Java Runtime Environment Kit (JRE).

Exemple : `/opt/IBM/WebSphere/eXtremeScale/java`

samples_home
samples_home est le répertoire dans lequel vous extrayez les exemples de fichiers qui sont utilisés pour les tutoriels.

Exemple : `/wxs-samples/`

dvd_root
dvd_root est le répertoire racine du DVD qui contient le produit.

Exemple : `dvd_root/docs/`

equinox_root
Le répertoire *equinox_root* est le répertoire racine de l'installation de l'infrastructure OSGi Eclipse Equinox.

Exemple :`/opt/equinox`

user_home
Le répertoire *user_home* est l'emplacement de stockage des fichiers utilisateur, tels que les profils de sécurité.

Windows `c:\Documents and Settings\user_name`

UNIX `/home/user_name`

Planification de la capacité de l'environnement

Si la taille initiale et la taille projetée des données ont été définies, vous pouvez planifier la capacité dont vous avez besoin pour exécuter WebSphere eXtreme Scale. En utilisant ces exercices de planification, vous pouvez déployer WebSphere eXtreme Scale de manière efficace pour les modifications futures et optimiser l'élasticité de la grille de données, ce que vous ne pourriez pas faire dans un autre scénario, par exemple avec une base de données interne ou un autre type de base de données.

Définition de la taille de la mémoire et calcul du nombre de partitions

Vous pouvez calculer la quantité de mémoire et le nombre de partitions nécessaires pour votre configuration.

Avertissement : Cette rubrique s'applique lorsque vous n'utilisez pas le mode de copie COPY_TO_BYTE. Si vous utilisez le mode COPY_TO_BYTES, la taille de la mémoire est beaucoup plus petite et la procédure de calcul est différente.

WebSphere eXtreme Scale stocke les données dans l'espace adresse de machines virtuelles Java (JVM). Chaque JVM fournit un espace processeur pour traiter la création, la récupération, la mise à jour et la suppression d'appels pour les données stockées dans la JVM. En outre, chaque JVM fournit de l'espace mémoire pour les serveurs secondaires et les entrées de données. Les objets Java varient en taille. Par conséquent, vous devez effectuer une mesure afin d'estimer la quantité de mémoire nécessaire.

Pour adapter la taille de la mémoire à vos besoins, chargez les données d'application dans une seule JVM. Lorsque l'utilisation de segment de mémoire atteint 60 %, notez le nombre d'objets utilisés. Ce nombre correspond au nombre d'objets maximal recommandé pour chaque machines virtuelles Java. Pour obtenir la définition de taille la mieux adaptée, utilisez des données réalistes et introduisez tout index défini, car les index occupent également de la mémoire. La meilleure méthode pour dimensionner l'utilisation de la mémoire consiste à exécuter une sortie **verbosegc** de récupération de place, car cette sortie vous donne les valeurs après la récupération de place. Vous pouvez interroger l'utilisation du segment de mémoire à tout moment via des beans gérés ou à l'aide d'une programme, mais ces requêtes ne vous donnent qu'un cliché instantané du segment de mémoire. Ce cliché peut contenir de l'espace non récupéré. Par conséquent, cette méthode n'indique pas précisément la mémoire utilisée.

Mise à l'échelle de la configuration

Nombre de fragments par partition (valeur numShardsPerPartition)

Pour calculer le nombre de fragments par partition, ou valeur numShardsPerPartition, ajoutez 1 pour le fragment primaire plus le nombre total de fragments réplique souhaité.

```
numShardsPerPartition = 1 + total_number_of_replicas
```

Nombre de machines virtuelles Java (valeur minNumJVMs)

Pour mettre à l'échelle votre configuration, décidez d'abord du nombre total maximal d'objets à stocker. Pour déterminer le nombre de machines virtuelles Java nécessaire, utilisez la formule suivante :

```
minNumJVMS=(numShardsPerPartition * numObjs) / numObjsPerJVM
```

Arrondissez cette valeur à l'entier le plus près.

Nombre de fragments (valeur numShards)

Lorsque la taille finale est atteinte, utilisez 10 fragments pour chaque JVM. Comme indiqué précédemment, chaque JVM dispose d'un fragment primaire et (N-1) fragments de réplique, soit dans ce cas, neuf répliques. Etant donné que vous disposez déjà du nombre de machines virtuelles Java pour le stockage de données, vous pouvez multiplier le nombre de machines virtuelles Java par 10 pour obtenir le nombre de fragments :

```
numShards = minNumJVMs * 10 shards/JVM
```

Nombre de partitions

Si une partition dispose déjà d'un fragment primaire et d'un fragment réplique, cette partition a donc deux fragments (le primaire et le réplique). Le nombre de partitions correspond au nombre de fragments divisé par 2 et arrondi au nombre premier le plus proche. Si la partition présente un fragment primaire et deux secondaires, le nombre de partitions correspond au nombre de fragments divisé par 3 et arrondi au nombre premier le plus proche.

```
numPartitions = numShards / numShardsPerPartition
```

Exemple de mise à l'échelle

Dans cet exemple, le nombre d'entrées commence à 250 millions. Chaque année, le nombre d'entrées croît d'environ 14 %. Après sept ans, le nombre total d'entrées atteint 500 millions et vous devez donc planifier la capacité en conséquence. Pour une haute disponibilité, un serveur secondaire est nécessaire. Avec un serveur secondaire, le nombre d'entrées double, soit 1 000 000 000 entrées. Dans le cadre d'un test, deux millions d'entrées peuvent être stockées dans chaque JVM. L'utilisation de calculs pour ce scénario montre le besoin de la configuration suivante :

- 500 machines virtuelles Java pour stocker le nombre final d'entrées.
- 5 000 fragments, obtenus en multipliant 500 machines virtuelles Java par 10.
- 2 500 partitions, arrondies à 2503 (nombre premier supérieur le plus proche), calculées en prenant 5 000 fragments, divisés par deux pour les fragments primaires et secondaires.

Début de la configuration

A partir des calculs précédents, démarrez avec 250 machines virtuelles Java pour atteindre 500 machines virtuelles Java en cinq ans. Avec cette configuration, vous pouvez gérer la croissance incrémentielle jusqu'à ce que vous accédiez au nombre d'entrées final.

Dans cette configuration, environ 200 000 entrées sont stockées par partition (500 millions d'entrées divisées par 2 503 partitions). Dans la mappe qui contient les entrées, affectez au paramètre **numberOfBuckets** le nombre premier le plus proche, soit 70 887 dans l'exemple, ce qui maintient le rapport autour de trois.

Le nombre maximal de machines virtuelles Java est atteint

Lorsque vous atteignez le nombre maximal de 500 machines virtuelles Java, vous pouvez toujours agrandir votre grille de données. Lorsque le nombre de machines virtuelles Java dépasse le nombre maximal de 500, le nombre de fragments commence à tomber en dessous de 10 pour chaque JVM, ce qui est inférieur au nombre recommandé. La taille des fragments augmente et risque d'entraîner des problèmes. Répétez le processus de dimensionnement en tenant compte de la croissance future de nouveau et redéfinissez le nombre de partitions. Cela requiert un redémarrage de la grille de données complète ou une indisponibilité de votre grille de données.

Nombre de serveurs

Avertissement : N'utilisez la pagination sur un serveur sous aucune circonstance.

Une seule JVM utilise plus de mémoire que la taille d'un segment de mémoire. Par exemple, avec 1 Go de segment de mémoire, une JVM utilise en fait 1,4 Go de mémoire réelle. Déterminez la mémoire vive disponible sur le serveur. Divisez la quantité de mémoire vive par la quantité de mémoire pour chaque JVM pour obtenir le nombre maximal de machines virtuelles Java sur le serveur.

Définition du nombre d'unités centrales par partition

Bien que l'une des fonctions principales d'eXtreme Scale soit sa capacité d'évolutivité, il est également important d'évaluer et d'adapter le nombre idéal d'unités centrales en vue d'une montée en charge.

Les coûts liés au processeur comprennent :

- Opérations de création, d'extraction, de mise à jour et de suppression depuis les clients
- Coût de la réplication à partir d'autres machines virtuelles Java
- Coût de l'invalidation
- Coût de la politique d'éviction
- Coût de la récupération de place
- Coût de la logique d'application
- Coût de la sérialisation

machines virtuelles Java par serveur

Utilisez deux serveurs et démarrez le nombre maximal de JVM par serveur. Utilisez le nombre de partitions calculé à la section précédente. Ensuite, préchargez dans ces machines virtuelles Java une quantité de données ne dépassant pas la capacité des deux ordinateurs. Utilisez un serveur distinct en tant que client. Exécutez une simulation de transaction réaliste sur cette grille de données de deux serveurs.

Pour calculer la valeur de référence, essayez de saturer l'utilisation du processeur. Si vous n'y parvenez pas, c'est probablement parce que le réseau est saturé. Dans ce cas, ajoutez des cartes réseau et procédez à une permutation circulaire de ces machines virtuelles Java.

Exécutez les ordinateurs à 60% d'utilisation du processeur et mesurez le taux de transactions de création, d'extraction, de mise à jour et de suppression. Cette mesure indique la capacité de traitement des deux serveurs. Ce nombre double avec quatre serveurs, double encore avec huit serveurs, etc. Cette progression

suppose que la capacité du réseau et la capacité du client peuvent également progresser.

Les temps de réponse d'eXtreme Scale doivent donc rester stable au fur et à mesure que le nombre de serveurs évolue. Le débit des transactions doit progresser de manière linéaire à mesure que des ordinateurs sont ajoutés à la grille de données.

Définition de la taille d'unités centrales pour des transactions parallèles

Les transactions à partition unique présentent une évolution de débit linéaire à mesure que la taille de la grille augmente. Les transactions parallèles diffèrent des transactions à partition unique, car elles affectent un ensemble de serveurs (cet ensemble peut comprendre tous les serveurs).

Si une transaction affecte tous les serveurs, le débit est limité au débit du client ayant initié la transaction ou au serveur affecté le plus lent. Les grilles de données de grande taille répartissent davantage les données et fournissent plus d'espace processeur, de mémoire, de réseau, etc. Toutefois, le client doit attendre la réponse du serveur le plus lent et doit utiliser les résultats de la transaction.

Lorsqu'une transaction affecte un sous-ensemble de serveurs, M sur N serveurs reçoivent une requête. Le débit est alors "N divisé par M" fois plus vite que le débit du serveur le plus lent. Par exemple, si vous disposez de 20 serveurs et d'une transaction qui affecte 5 serveurs, le débit est 4 fois supérieur au débit du serveur le plus lent de la grille de données.

Lorsqu'une transaction parallèle se termine, les résultats sont envoyés à l'unité d'exécution du client ayant commencé la transaction. Ce client doit ensuite procéder à l'agrégation des résultats en unité d'exécution simple. Le temps d'agrégation augmente avec l'augmentation du nombre de serveurs affectés par la transaction. Toutefois, cette durée dépend de l'application, car il se peut que chaque serveur renvoie un résultat plus petit à mesure que la taille de la grille de données augmente.

Généralement, les transactions parallèles affectent tous les serveurs dans la grille de données, car les partitions y sont réparties uniformément. Dans ce cas, le débit se limite à la première hypothèse.

Récapitulatif

Avec cette définition de taille, vous disposez de trois métriques, comme suit.

- Nombre de partitions.
- Nombre de serveurs nécessaires pour la mémoire requise.
- Nombre de serveurs nécessaires pour le débit requis.

Si vous avez besoin de 10 serveurs pour la quantité de mémoire nécessaire, mais que vous obtenez uniquement 50 % du débit requis en raison d'une saturation du processeur, vous devait avoir deux fois plus de serveurs.

Pour une stabilité optimale, vous devez exécuter les serveurs à 60 % du chargement de processeur et les segments de mémoire de JVM à 60 % du chargement des segments de mémoire. Les renforts peuvent ensuite pousser l'utilisation du processeur à 80-90 %, mais n'exécutent pas régulièrement vos

serveurs à un niveau supérieur à ces niveaux.

Planification de la capacité de la mémoire cache dynamique

L'API de cache dynamique est disponible pour les applications Java EE qui sont déployées dans WebSphere Application Server. Ce cache dynamique peut être optimisé pour mettre en cache les données métier et les fichiers HTML générés ou pour synchroniser les données en cache de la cellule à l'aide du service DRS de réplication des données.

Présentation

Toutes les instances du cache dynamique créées avec le fournisseur de cache dynamique de WebSphere eXtreme Scale sont par défaut à disponibilité élevée. Le niveau et le coût de la mémoire de la haute disponibilité dépendent de la topologie utilisée.

Si vous utilisez une topologie imbriquée, la taille du cache est limitée à la quantité de mémoire disponible dans un processus serveur unique et chaque processus serveur stocke une copie complète du cache. Tant que l'exécution de ce processus serveur unique se poursuit, le cache survit. Les données de cette mémoire sont perdues uniquement en cas d'arrêt de tous les serveurs qui y accèdent.

Dans le cas d'une topologie partitionnée imbriquée, la taille du cache est limitée à l'ensemble de l'espace disponible dans tous les processus serveur. Le fournisseur de cache dynamique eXtreme Scale par défaut utilise une réplique pour chaque fragment primaire, de sorte que chaque donnée mise en cache est stockée deux fois.

Utilisez la formule A pour déterminer la capacité du cache partitionné imbriqué.

Formule A

D * C / (1 + S) = M

Où :

- D = Mémoire disponible par processus conteneur
- C = nombre de conteneurs
- S = nombre de fragments réplique
- M = taille totale du cache

Pour une grille de données WebSphere Application Server Network Deployment disposant d'un espace disponible de 256 Mo dans chaque processus, avec 4 processus serveur, une instance du cache sur tous ces serveurs peut stocker jusqu'à 512 mégaoctets de données. Dans ce mode, le cache peut survivre à une panne de serveur sans perte de données. De la même manière, deux serveurs au maximum peuvent être arrêtés séquentiellement sans perte de données. Pour l'exemple suivant, la formule est la suivante :

256 Mo * 4 conteneurs/ (1 primaire + 1 réplique) = 512 Mo.

Les caractéristiques de taille des mémoires cache utilisant une topologie distante sont similaires à celles qui utilisent une topologie partitionnée imbriquée, mais les premières sont limitées à la quantité d'espace disponible dans tous les processus conteneur eXtreme Scale.

Dans une topologie distante, il est possible d'augmenter le nombre de fragments réplique pour atteindre un niveau de disponibilité plus élevé, au prix d'une augmentation de la quantité de mémoire. Dans la plupart des applications de cache dynamique, cela est inutile, mais vous pouvez modifier le fichier `dynacache-remote-deployment.xml` pour augmenter le nombre de répliques.

Utilisez les formules suivantes, B et C, afin de déterminer l'impact de l'ajout de fragments de réplique sur la haute disponibilité du cache.

Formule B

N = Minimum(T -1, S)

Où :

- N = nombre d'échecs simultanés de processus
- T = nombre total de conteneurs
- S = nombre total de fragments réplique

Formule C

Ceiling(T/ (1+N)) = m

Où :

- T = nombre total de conteneurs
- N = nombre total de fragments réplique
- m = nombre minimal de conteneurs nécessaires pour prendre en charge les données en cache.

Pour l'optimisation des performances avec le fournisseur de cache dynamique, voir «Optimisation du fournisseur de cache dynamique», à la page 505.

Définition de la taille du cache

Pour qu'une application utilisant le fournisseur de cache dynamique WebSphere eXtreme Scale puisse être déployée, les principes généraux décrits dans la section précédente doivent être combinés avec les données environnementales des systèmes de production. Les premiers chiffres à identifier sont le nombre total de processus conteneur et la quantité de mémoire disponible dans chaque processus pouvant contenir les données du cache. Dans une topologie imbriquée, les conteneurs du cache sont aussi localisés dans les processus serveur WebSphere Application, de sorte qu'il existe un conteneur par serveur partageant le cache. Le meilleur moyen de définir l'espace disponible dans le processus est d'identifier la quantité de mémoire supplémentaire de l'application sans activer la mise en cache et WebSphere Application Server. Pour cela, vous pouvez analyser les données détaillées de récupération de place. Lorsque la topologie utilisée est distante, vous trouvez ces informations dans la sortie détaillée de la récupération de place d'un conteneur autonome ayant démarré récemment et dans lequel aucune donnée de cache n'a encore été entrée. Dernier élément auquel vous devez penser : vous devez réserver des segments de la mémoire pour la récupération de place. La somme de l'espace restant dans le conteneur WebSphere Application Server ou dans le conteneur autonome et de la taille réservée pour le cache ne doit pas dépasser 70 % du total des segments de mémoire.

Une fois ces informations collectées, les valeurs peuvent être entrées dans la formule A décrite précédemment, pour déterminer la taille maximale du cache partitionné. Une fois cette taille connue, l'étape suivante consiste à déterminer le nombre total d'entrées du cache pouvant être prises en charge, ce qui suppose d'identifier la taille moyenne de chaque entrée. Il suffit pour cela d'ajouter 10% à la taille de l'objet client. Voir le guide sur l'optimisation du cache dynamique et du service de réplication des données pour plus d'informations sur l'utilisation du cache dynamique.

Lorsque la compression est activée, elle affecte la taille de l'objet client et non l'espace restant dans le système de mise en cache. Utilisez la formule suivante pour déterminer la taille d'un objet mis en cache lorsque la compression est utilisée :

T = O * C + O * 0.10

Où :

- T = Taille moyenne de l'objet mis en cache
- O = Taille moyenne de l'objet client non compressé
- C = Rapport de compression exprimé sous la forme d'une fraction.

Un rapport de compression de 2 à 1 est égal à 1/2 = 0.50. Une valeur moins élevée est recommandée. Si l'objet stocké est un objet Java simple normal principalement rempli de types primitifs, vous pouvez supposer que le rapport de compression est de l'ordre de 0,60 à 0,70. Si l'objet mis en cache est un objet servlet, JSP ou WebServices, la meilleure méthode pour déterminer le rapport de compression consiste à compresser un exemple représentatif avec un utilitaire de compression ZIP. Si cette opération est impossible, considérez qu'un rapport de compression compris entre 0,2 et 0,35 est fréquent pour ce type de données.

Ensuite, utilisez ces informations pour déterminer le nombre total d'entrées du cache qui peuvent être prises en charge. Utilisez la formule D suivante :

Formule D

T = S / M

Où :

- T = Nombre total d'entrées du cache
- S = Taille totale disponible pour les données du cache, calculée à l'aide de la formule A
- M = Taille moyenne de chaque entrée du cache

Enfin, vous devez définir la taille de l'instance de cache dynamique pour appliquer cette limite. A cet égard, le fournisseur de cache dynamique WebSphere eXtreme Scale est différent du fournisseur de cache dynamique par défaut. Utilisez la formule suivante pour déterminer la valeur de la taille de l'instance de cache dynamique. Cette formule est la suivante :

Formule E

Ct = Tt / Np

Où :

- Tt = Taille cible totale du cache

- Ct = Paramètre de la taille du cache à définir dans l'instance de cache dynamique
- Np = Nombre de partitions. La valeur par défaut est 47.

Associez la taille de l'instance de cache dynamique à une valeur calculée par la formule E pour chaque serveur partageant l'instance du cache.

Planification de la configuration

Avant de configurer le matériel ou logiciel, vous devez tenir compte des points suivants.

Liste de contrôle opérationnelle

Utilisez la liste de contrôle opérationnelle afin de préparer votre environnement pour le déploiement de WebSphere eXtreme Scale.

Tableau 3. Liste de contrôle opérationnelle

Elément de la liste de contrôle	Pour plus d'informations

Si vous utilisez AIX, optimisez les paramètres suivants du système d'exploitation :
 	
TCP_KEEPINTVL
Le paramètre TCP_KEEPINTVL fait partie d'un protocole de maintien de connexion du socket qui permet la détection des indisponibilités du réseau. La propriété spécifie l'intervalle entre les paquets envoyés pour valider la connexion. Si vous utilisez WebSphere eXtreme Scale, spécifiez la valeur `10`. Pour vérifier le paramètre actuel, exécutez la commande suivante :
`# no –o tcp_keepintvl`
Pour modifier le paramètre actuel, exécutez la commande suivante :
`# no –o tcp_keepintvl=10`
Le paramètre TCP_KEEPINTVL est exprimé en demi secondes.
 	
TCP_KEEPINIT
Le paramètre TCP_KEEPINIT fait partie d'un protocole de maintien de connexion du socket qui permet la détection des indisponibilités du réseau. La propriété spécifie le délai d'attente initial de la connexion TCP. Si vous utilisez WebSphere eXtreme Scale, spécifiez la valeur `40`. Pour vérifier le paramètre actuel, exécutez les commandes suivantes :
`# no –o tcp_keepinit`
Pour modifier le paramètre actuel, exécutez la commande suivante :
`# no –o tcp_keepinit=40`
Le paramètre TCP_KEEPINIT est exprimé en demi secondes.
	• Pour des informations sur l'optimisation d'AIX, voir la rubrique Optimisation des systèmes AIX.

Mettez à jour le fichier `orb.properties` pour modifier le comportement de transport de la grille. Le fichier `orb.properties` se trouve dans le répertoire `java/jre/lib`.	«Propriétés ORB», à la page 492

Tableau 3. Liste de contrôle opérationnelle (suite)

Elément de la liste de contrôle	Pour plus d'informations
Utilisez des paramètres dans le script **startOgServer**. En particulier, utilisez les paramètres suivants : • Définissez les paramètres de segment de mémoire avec l'option **-jvmArgs**. • Définissez les propriétés et le chemin d'accès aux classes avec l'option **-jvmArgs**. • Définissez les paramètres **-jvmArgs** pour configurer la surveillance de l'agent. **Paramètres de port** WebSphere eXtreme Scale doit ouvrir des ports pour les communications pour certains transports. Ces ports sont tous définis de manière dynamique. Toutefois, si un pare-feu est utilisé entre les conteneurs, vous devez spécifier les ports. Utilisez les informations suivantes sur les ports : **Port du programme d'écoute** Vous pouvez utiliser l'argument **-listenerPort** pour spécifier le port utilisé pour les communications entre les processus. **Port du groupe central** Vous pouvez utiliser l'argument **-haManagerPort** pour spécifier le port utilisé pour la détection des incidents. Cet argument correspond à peerPort. Notez que les groupes centraux n'ayant pas besoin de communiquer entre les zones, il n'est pas nécessaire de définir ce port si le pare-feu est ouvert pour tous les membres d'une même zone. **Port du service JMX** Vous pouvez utiliser l'argument **-JMXServicePort** pour spécifier le port à utiliser par le service JMX. **Port SSL** Si vous transmettez `-Dcom.ibm.CSI.SSLPort=1234` comme argument **-jvmArgs**, le port SSL prend la valeur `1234`. Le port SSL est l'homologue de port sécurisé du port du programme d'écoute. **Port du client** Utilisé uniquement dans le service de catalogue. Vous pouvez spécifier cette valeur avec l'argument **-catalogServiceEndPoints**. La valeur de ce paramètre est au format suivant : `nomServeur:nomHôte:portClient:portHomologue`	«Script **startOgServer**», à la page 401
Vérifiez que les paramètres de sécurité sont configurés correctement : • Transport (SSL) • Application (Authentification et autorisation) Pour vérifier vos paramètres de sécurité, vous pouvez essayer d'utiliser un client malveillant pour vous connecter à votre configuration. Par exemple, si le paramètre SSL requis est configuré, un client possédant un paramètre TCP_IP avec ce dernier ou un client possédant un fichier de clés certifiées incorrect ne doit pas pouvoir se connecter au serveur. Si l'authentification est requise, un client sans données d'identification, telles qu'un ID utilisateur et un mot de passe ne doit pas pouvoir se connecter au serveur. Si l'autorisation est appliquée, un client sans autorisation d'accès ne doit pas être autorisé à accéder aux ressources du serveur.	«Intégration de la sécurité à des fournisseurs externes», à la page 520

Tableau 3. Liste de contrôle opérationnelle (suite)

Elément de la liste de contrôle	Pour plus d'informations
Choisissez comment vous allez surveiller votre environnement. • Outil **xscmd** : – Les ports JMX des serveurs de catalogue doivent être visibles de l'outil **xscmd**. Les ports de serveur de conteneur doivent également être accessibles pour certaines commandes qui collectent des informations des conteneurs. • Console de surveillance : Avec la console de surveillance, vous pouvez générer des graphiques des statistiques actuelles et historiques. • Outils de surveillance du fournisseur : – Tivoli Enterprise Monitoring Agent – CA Wily Introscope – Hyperic HQ	• «Surveillance avec l'utilitaire **xscmd**», à la page 462 • «Sécurité JMX (Java Management Extensions)», à la page 517 • «Surveillance à l'aide de la console Web», à la page 443 • «Surveillance à l'aide d'IBM Tivoli Enterprise Monitoring Agent for WebSphere eXtreme Scale», à la page 477 • «Surveillance d'eXtreme Scale à l'aide de Hyperic HQ», à la page 486 • «Surveillance des applications eXtreme Scale à l'aide de CA Wily Introscope», à la page 483

Planification des ports réseau

WebSphere eXtreme Scale est un cache réparti qui nécessite l'ouverture de ports pour communiquer avec l'ORB (Object Request Broker) et la pile TCP (Transmission Control Protocol) sur les machines virtuelles Java. Planifiez et contrôlez vos ports, en particulier dans un environnement comportant un pare-feu, et lorsque vous utilisez un service de catalogue et des conteneurs sur plusieurs ports.

Important : Lorsque vous spécifiez des numéros de port, évitez de définir les ports qui se trouvent dans la plage éphémère du système d'exploitation afin d'éviter les conflits de ports.

Domaine de services de catalogue

Un domaine de services de catalogue nécessite que soient définis les ports suivants :

peerPort
Spécifie le port qui permet au gestionnaire de haute disponibilité (HA) de communiquer entre serveurs de catalogue homologues dans une pile TCP. Dans WebSphere Application Server, ce paramètre est hérité par la configuration de port du gestionnaire haut disponibilité.

clientPort
Spécifie le port qui permet aux serveurs de catalogue d'accéder aux données des services de catalogue. Dans WebSphere Application Server, ce port est défini par le biais de la configuration du domaine de services de catalogue.

listenerPort
Indique le numéro de port auquel se connecte l'ORB (Object Request Broker). Ce paramètre configure les conteneurs et les clients pour communiquer avec le service de catalogue via l'ORB. Dans WebSphere Application Server, le port d'écoute est hérité par la configuration de port BOOTSTRAP_ADDRESS. Cette propriété s'applique au serveur de conteneur et au service de catalogue.

Valeur par défaut :2809

JMXConnectorPort
Définit le port SSL (Secure Sockets Layer) auquel se connecte le service Java Management Extensions (JMX).

Serveurs conteneurs

Les serveurs conteneurs WebSphere eXtreme Scale requièrent également que plusieurs ports soient en fonctionnement. Par défaut, le serveur conteneur eXtreme Scale génère automatiquement son port de gestionnaire HA et son port d'écoute ORB avec des ports dynamiques. Pour un environnement qui dispose d'un pare-feu, il est avantageux pour vous de planifier et de contrôler les ports. Pour les serveurs de conteneur à démarrer avec des ports spécifiques, vous pouvez utiliser les options suivantes dans la commande **startOgServer**.

haManagerPort
Synonyme avec port homologue. Indique le numéro de port utilisé par le gestionnaire de haute disponibilité. Si cette propriété n'est pas définie, le service de catalogue génère automatiquement un port disponible. Cette propriété s'applique à la fois au serveur conteneur et au service de catalogue. (Requis pour les environnements WebSphere Application Server uniquement.)

listenerPort
Indique le numéro de port auquel se connecte l'ORB (Object Request Broker). Ce paramètre configure les conteneurs et les clients pour communiquer avec le service de catalogue via l'ORB. Dans WebSphere Application Server, le port d'écoute est hérité par la configuration de port BOOTSTRAP_ADDRESS. Cette propriété s'applique au serveur de conteneur et au service de catalogue.

Valeur par défaut :2809

JMXConnectorPort
Définit le port SSL (Secure Sockets Layer) auquel se connecte le service Java Management Extensions (JMX).

7.1.1+ **xioChannel.xioContainerTCPSecure.Port**
Indique le numéro de port SSL de eXtremeIO sur le serveur. Cette propriété est utilisée uniquement lorsque la propriété **transportType** a la valeur `SSL-Supported` ou `SSL-Required`.

7.1.1+ **xioChannel.xioContainerTCPNonSecure.Port**
Indique le numéro de port d'écoute non sécurisé de eXtremeIO sur le serveur. Si vous ne définissez pas de valeur, un port éphémère est utilisé. Cette propriété est utilisée uniquement lorsque la propriété **transportType** a la valeur `TCP/IP`.

Une planification du port de contrôle est essentielle lorsque des centaines de machines virtuelles Java sont démarrées dans un serveur. Si un conflit de port existe, les serveurs de conteneur ne démarrent pas.

Clients

Les clients WebSphere eXtreme Scale peuvent recevoir des rappels de serveurs lorsque vous utilisez l'API DataGrid ou plusieurs autres commandes. Utilisez la propriété **listenerPort** dans le fichier de propriétés du client afin de spécifier le port sur lequel le client écoute les rappels à partir du serveur.

haManagerPort
Synonyme avec port homologue. Indique le numéro de port utilisé par le gestionnaire de haute disponibilité. Si cette propriété n'est pas définie, le service de catalogue génère automatiquement un port disponible. Cette

propriété s'applique à la fois au serveur conteneur et au service de catalogue. (Requis pour les environnements WebSphere Application Server uniquement.)

jvmArgs (facultatif)
Spécifie la liste des arguments JVM (Java virtual machine). Lorsque la sécurité est activée, vous devez utiliser l'argument suivant pour configurer le port SSL (Secure Socket Layer) : `-jvmArgs -Dcom.ibm.CSI.SSLPort=<sslPort>`.

listenerPort
Indique le numéro de port auquel se connecte l'ORB (Object Request Broker). Ce paramètre configure les conteneurs et les clients pour communiquer avec le service de catalogue via l'ORB. Dans WebSphere Application Server, le port d'écoute est hérité par la configuration de port BOOTSTRAP_ADDRESS. Cette propriété s'applique au serveur de conteneur et au service de catalogue.

Valeur par défaut :2809

Ports dans WebSphere Application Server

- La valeur **listenerPort** est héritée de la valeur **BOOTSTRAP_ADDRESS** pour chaque serveur d'applications WebSphere Application Server.
- Les valeurs **haManagerPort** et **peerPort** héritées de la valeur **DCS_UNICAST_ADDRESS** pour chaque serveur d'applications WebSphere Application Server.

Vous pouvez définir un domaine de services de catalogue dans la console d'administration, comme indiqué dans «Création de domaines de services de catalogue dans WebSphere Application Server», à la page 258.

Vous pouvez afficher les ports d'un serveur particulier en cliquant sur un des chemins suivants dans la console d'administration :

- WebSphere Application Server Network Deployment Version 6.1 : **Serveurs** > **Serveurs d'applications** > *server_name* > **Ports** > *end_point_name*.
- WebSphere Application Server Network Deployment Version 7.0 : **Serveurs** > **Types de serveur** > **Serveurs d'applications WebSphere** > *server_name* > **Ports** > *port_name*

Sécurité

WebSphere eXtreme Scale permet de sécuriser l'accès aux données et l'intégration de fournisseurs de sécurité externes.

Remarque : Dans un magasin de données non mis en cache, une base de données, par exemple, il est probable que certaines fonctions pré-intégrées de sécurité ne vous serviront à rien pour la configuration ou l'activation. Cependant, une fois vos données mises en cache avec eXtreme Scale, vous devez prendre en compte le fait que vos fonctions de sécurité du dorsal ne sont plus actives. Vous pouvez configurer la sécurité de eXtreme Scale aux niveaux nécessaires, de sorte que votre nouvelle architecture mise en cache soit également sécurisée.
Vous trouverez ci-dessous un bref récapitulatif des fonctions de sécurité de eXtreme Scale. Pour des informations plus détaillées sur la configuration de la sécurité, voir *Guide d'administration* et *Guide de programmation*.

Notions de base sur la sécurité répartie

La sécurité répartie eXtreme Scale se base sur trois concepts :

Authentification approuvée
: Possibilité de déterminer l'identité du demandeur. WebSphere eXtreme Scale prend en charge l'authentification client-serveur et serveur-serveur.

Autorisation
: Possibilité d'octroyer des droits d'accès au demandeur. WebSphere eXtreme Scale prend en charge différentes autorisations pour des opérations diverses.

Transfert sécurisé
: Transmission sécurisé des données sur le réseau. WebSphere eXtreme Scale prend en charge les protocoles Transport Layer Security/Secure Sockets Layer (TLS/SSL).

Authentification

WebSphere eXtreme Scale prend en charge les structures de serveurs clients répartis. Une infrastructure de sécurité du serveur client est en place pour sécuriser l'accès aux serveurs eXtreme Scale. Par exemple, lorsque l'authentification est requise par le serveur eXtreme Scale, un client eXtreme Scale doit fournir ses informations d'identification pour s'authentifier sur le serveur. Ces informations peuvent être un nom d'utilisateur et un mot de passe, un certificat client, un ticket Kerberos ou des données présentées dans un format choisi par le client et le serveur.

Autorisation

Les autorisations WebSphere eXtreme Scale sont basées sur des objets et des permissions. Vous pouvez utiliser le service JAAS (Java Authentication and Authorization Services) pour autoriser l'accès, ou vous pouvez choisir une approche personnalisée, telle que Tivoli Access Manager (TAM), pour gérer les autorisations. Les autorisations suivantes peuvent être octroyées à un client ou un groupe :

Autorisation de mappes
: Effectuez des opérations d'insertion, de lecture, de mise à jour, d'expulsion ou de suppression sur les mappes.

Autorisation ObjectGrid
: Lancez des requêtes sur un objet ou une entité et des requêtes de flux sur les objets ObjectGrid.

Autorisation de l'agent DataGrid
: Permet aux agents DataGrid d'être déployés en une base de données ObjectGrid.

Autorisation de mappes côté serveur
: Répliquez une mappe de serveur côté client ou créez un index dynamique pour la mappe de serveur.

Autorisation d'administration
: Effectuez des tâches d'administration.

Sécurité du transfert

Pour sécuriser la communication du serveur client, WebSphere eXtreme Scale prend en charge les protocoles TLS/SSL. Ces protocoles fournissent une sécurité de couche de transport, avec des fonctions d'authentification, d'intégrité et de confidentialité pour une connexion sécurisée entre le client eXtreme Scale et le serveur.

Sécurité de grille

Dans un environnement sécurisé, un serveur doit être capable de vérifier l'authenticité d'un autre serveur. WebSphere eXtreme Scale utilise un mécanisme de clé secrète partagée dans ce but. Ce mécanisme est similaire à un mot de passe partagé. Tous les serveurs eXtreme Scale s'accordent sur une clé secrète partagée. Lorsqu'un serveur rejoint la grille de données, il est invité à présenter la chaîne secrète. Si la clé secrète du serveur tentant de se joindre correspond à la clé sur serveur principal, le serveur peut se joindre à la grille. Dans le cas contraire, la requête de jointure est rejetée.

L'envoi d'une clé secrète en texte clair n'est pas sécurisé. L'infrastructure de sécurité eXtreme Scale fournit un plug-in SecureTokenManager pour permettre au serveur de sécuriser cette clé secrète avant l'envoi. Vous pouvez choisir la façon dont vous souhaitez implémenter l'opération sécurisée. Avec WebSphere eXtreme Scale, une opération sécurisée est implémentée pour chiffrer et signer la clé secrète.

Fonctions de sécurité Java Management Extensions (JMX) dans une topologie de déploiement dynamique

Les fonctions de sécurité JMX MBeans sont prises en charge dans toutes les versions de eXtreme Scale. Les clients des beans gérés de serveur de catalogue et de serveur de conteneur peuvent être authentifiés, et l'accès aux opérations MBean peut être forcé.

Sécurité eXtreme Scale locale

La sécurité eXtreme Scale locale est différente du modèle eXtreme Scale réparti car l'application s'instancie directement et utilise une instance ObjectGrid. Votre application et les instances eXtreme Scale se trouvent dans la même machine virtuelle Java (JVM). Etant donné qu'aucun concept client-serveur n'existe dans ce modèle, l'authentification n'est pas prise en charge. Vos applications doivent gérer leur propre authentification, puis transmettre l'objet authentifié à eXtreme Scale. Cependant, le mécanisme d'autorisation utilisé pour le modèle de programmation eXtreme Scale est le même que celui utilisé pour le modèle client-serveur.

Configuration et programmation

Pour plus d'informations sur la configuration et la programmation de la sécurité, voir «Intégration de la sécurité à des fournisseurs externes», à la page 520 et API de sécurité.

Chapitre 3. Tutoriels

Vous pouvez utiliser les tutoriels pour mieux comprendre les scénarios d'utilisation du produit, y compris le gestionnaire d'entités, les requêtes et la sécurité.

Tutoriel : Configuration de la sécurité Java SE

Le tutoriel suivant vous permet de créer un environnement eXtreme Scale dans un environnement Java Platform, Standard Edition.

Avant de commencer

Assurez-vous que vous connaissez les principes de base d'une configuration eXtreme Scale répartie.

Pourquoi et quand exécuter cette tâche

Dans ce tutoriel, le serveur de catalogue, le serveur de conteneur et le client s'exécutent tous dans un environnement Java SE. Chaque étape du tutoriel est liée à l'étape qui la précède. Effectuez toutes les étapes afin de sécuriser un eXtreme Scale réparti et de développer une application Java SE simple pour accéder au eXtreme Scale sécurisé.

Commencer le tutoriel

Procédure

1. «Tutoriel sur la sécurité Java SE - Etape 1», à la page 70
   - Démarrer un serveur de catalogue non sécurisé
   - Démarrer un serveur de conteneur non sécurisé
   - Démarrer un client pour accéder aux données
   - Utiliser l'utilitaire **xscmd** pour afficher une taille de mappe
   - Arrêter le serveur
2. «Tutoriel sur la sécurité Java SE - Etape 2», à la page 73
   - Utiliser CredentialGenerator
   - Utiliser Authenticator
   - Démarrer un serveur de catalogue sécurisé
   - Démarrer un serveur de conteneur sécurisé
   - Démarrer un client pour accéder à l'ObjectGrid sécurisé
   - Utiliser l'utilitaire **xscmd** pour afficher une taille de mappe
   - Arrêter le serveur sécurisé
3. «Tutoriel sur la sécurité Java SE - Etape 3», à la page 79
   - Utiliser la politique d'autorisation JAAS
4. «Tutoriel sur la sécurité Java SE - Etape 4», à la page 83
   - Créer un fichier de clés et un fichier de clés certifiées
   - Configurer les propriétés SSL pour le serveur
   - Configurer les propriétés SSL pour le client

- Utiliser l'utilitaire **xscmd** pour afficher une taille de mappe
- Arrêter le serveur sécurisé

Tutoriel sur la sécurité Java SE - Etape 1

Cette rubrique décrit *un exemple simple non sécurisé*. Des fonctions de sécurité supplémentaires sont ajoutées au fur et à mesure des étapes du tutoriel afin d'augmenter le niveau de sécurité intégrée disponible.

Avant de commencer

Remarque : Tous les fichiers requis pour cette étape du tutoriel sont fournis dans la section suivante.

Procédure

Exécution de l'exemple
Démarrez le service de catalogue en utilisant les scripts suivants. Pour plus d'informations sur le démarrage du service de catalogue, voir «Démarrage d'un service de catalogue autonome», à la page 395.

1. Placez-vous dans le répertoire bin : `cd objectgridRoot/bin`
2. Démarrez un serveur de catalogue nommé catalogServer :
   - UNIX Linux `startOgServer.sh catalogServer`
   - Windows `startOgServer.bat catalogServer`
3. Placez-vous dans le répertoire bin `cd objectgridRoot/bin`
4. Démarrez un serveur de conteneur nommé c0 avec le script suivant :
   - UNIX Linux

     ```
     startOgServer.sh c0 -objectGridFile ../xml/SimpleApp.xml -deploymentPolicyFile ../xml/SimpleDP.xml
     -catalogServiceEndPoints localhost:2809
     ```

   - Windows

     ```
     startOgServer.bat c0 -objectGridFile ../xml/SimpleApp.xml - deploymentPolicyFile ../xml/SimpleDP.xml
     -catalogServiceEndPoints localhost:2809
     ```

Exemple

Pour plus d'informations sur le démarrage des serveurs de conteneurs, voir «Démarrage des serveurs de conteneur», à la page 398.

Une fois le serveur de catalogue et le serveur de conteneur démarrés, lancez le client comme suit.

1. Placez-vous dans le répertoire bin une nouvelle fois.
2. `java -classpath ../lib/objectgrid.jar;../applib/secsample.jar com.ibm.websphere.objectgrid.security.sample.guide.SimpleApp`

Le fichier `secsample.jar` contient la classe SimpleApp.

La sortie générée par ce programme est la suivante :

```
Le nom de client pour ID 0001 est fName lName
```

Vous pouvez également utiliser l'utilitaire **xscmd** pour afficher les tailles de mappes de la grille "accounting".

- Placez-vous dans le répertoire `objectgridRoot/bin`.
- Utilisez la commande **xscmd** pour afficher les tailles de mappes :

- UNIX Linux `xscmd.sh -c showMapSizes -g accounting -ms mapSet1`
- Windows `xscmd.bat -c showMapSizes -g accounting -ms mapSet1`

Arrêt des serveurs

Serveur de conteneur

Utilisez la commande suivante pour arrêter le serveur de conteneur c0.

UNIX Linux
```
stopOgServer.sh c0 -catalogServiceEndPoints localhost:2809
```

Windows
```
stopOgServer.bat c0 -catalogServiceEndPoints localhost:2809
```

Le message suivant s'affiche.

```
CWOBJ2512I: ObjectGrid server c0 stopped.
```

Serveur de catalogue

Vous pouvez arrêter un serveur de catalogue avec la commande suivante.

UNIX Linux
```
stopOgServer.sh catalogServer -catalogServiceEndPoints localhost:2809
```

Windows
```
stopOgServer.bat catalogServer -catalogServiceEndPoints localhost:2809
```

Si vous arrêtez le serveur de catalogue, le message suivant s'affiche.

```
CWOBJ2512I: ObjectGrid server catalogServer stopped.
```

Fichiers requis

Le fichier ci-dessous correspond à la classe Java pour SimpleApp.

SimpleApp.java

```
// Cet exemple de programme est fourni TEL QUEL et peut être utilisé, exécuté, copié et modifié
// gratuitement par le client
// (a) à des fins d'études,
// (b) afin de développer des applications conçues pour être exécutées
//  avec un produit IBM WebSphere,
// soit pour un usage interne soit pour une redistribution par le client, en tant que partie de
// l'application, au sein des produits du client.
//   Eléments sous licence - Propriété d'IBM
// 5724-J34 (C) COPYRIGHT International Business Machines Corp. 2007-2009
package com.ibm.websphere.objectgrid.security.sample.guide;

import com.ibm.websphere.objectgrid.ClientClusterContext;
import com.ibm.websphere.objectgrid.ObjectGrid;
import com.ibm.websphere.objectgrid.ObjectGridManager;
import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
import com.ibm.websphere.objectgrid.ObjectMap;
import com.ibm.websphere.objectgrid.Session;

public class SimpleApp {

    public static void main(String[] args) throws Exception {

        SimpleApp app = new SimpleApp();
        app.run(args);
    }
```

```
    /**
     * read and write the map
     * @throws Exception
     */
    protected void run(String[] args) throws Exception {
        ObjectGrid og = getObjectGrid(args);

        Session session = og.getSession();

        ObjectMap customerMap = session.getMap("customer");

        String customer = (String) customerMap.get("0001");

        if (customer == null) {
            customerMap.insert("0001", "fName lName");
        } else {
            customerMap.update("0001", "fName lName");
        }
        customer = (String) customerMap.get("0001");

        System.out.println("The customer name for ID 0001 is " + customer);
    }

    /**
     * Get the ObjectGrid
     * @return an ObjectGrid instance
     * @throws Exception
     */
    protected ObjectGrid getObjectGrid(String[] args) throws Exception {
        ObjectGridManager ogManager = ObjectGridManagerFactory.getObjectGridManager();

        // Crée un ObjectGrid
        ClientClusterContext ccContext = ogManager.connect("localhost:2809", null, null);
        ObjectGrid og = ogManager.getObjectGrid(ccContext, "accounting");

        return og;

    }

}
```

La méthode getObjectGrid de cette classe obtient un ObjectGrid et la méthode run lit un enregistrement à partir de la mappe client et met à jour la valeur.

Pour exécuter cet échantillon de code dans un environnement réparti, un fichier XML de descripteur d'ObjectGrid, `SimpleApp.xml`, et un fichier XML de déploiement, `SimpleDP.xml`, sont créés. Les fichiers sont inclus dans l'exemple suivant :

SimpleApp.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
    <objectGrids>
        <objectGrid name="accounting">
            <backingMap name="customer" readOnly="false" copyKey="true"/>
        </objectGrid>
    </objectGrids>
</objectGridConfig>
```

Le fichier XML suivant permet de configurer l'environnement de déploiement.

SimpleDP.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

 <objectgridDeployment objectgridName="accounting">
  <mapSet name="mapSet1" numberOfPartitions="1" minSyncReplicas="0" maxSyncReplicas="2"
maxAsyncReplicas="1">
```

```
   <map ref="customer"/>
  </mapSet>
 </objectgridDeployment>
</deploymentPolicy>
```

Il s'agit une configuration ObjectGrid simple avec une instance ObjectGrid nommée "accounting" et une mappe nommée "customer" (faisant partie de l'ensemble de mappes "mapSet1"). Le fichier `SimpleDP.xml` contient un ensemble de mappes configuré avec 1 partition et 0 réplique minimum requise.

Etape suivante du tutoriel

Tutoriel sur la sécurité Java SE - Etape 2

Une fois les étapes précédentes effectuées, la rubrique suivante illustre l'implémentation d'une authentification client dans un environnement eXtreme Scale réparti.

Avant de commencer

Assurez-vous d'avoir effectué les étapes du «Tutoriel sur la sécurité Java SE - Etape 1», à la page 70.

Pourquoi et quand exécuter cette tâche

Une fois l'authentification client activée, un client est authentifié avant de se connecter au serveur eXtreme Scale. Cette section illustre comment activer l'authentification client dans un environnement de serveur eXtreme Scale et inclut des extraits de code et des scripts.

Comme tout autre mécanisme d'authentification, l'authentification se compose au minimum des étapes suivantes :

1. L'administrateur modifie la configuration afin de rendre l'authentification obligatoire.
2. Le client fournit des données d'identification au serveur.
3. Le serveur compare les données d'identification fournies au registre.

Procédure

1. **Données d'identification client**

   Les données d'identification client sont représentées par une interface com.ibm.websphere.objectgrid.security.plugins.Credential. Les données d'identification client peuvent être une paire nom-mot de passe, un ticket Kerberos, un certificat client ou des données au format convenu par le client et le serveur. Pour plus d'informations, référez-vous à la documentation sur l'API de données d'identification.

   Cette interface définit explicitement les méthodes equals(Object) et hashCode(). Ces deux méthodes sont importantes car les objets Subject authentifiés sont mis en cache par le biais de l'objet Credential en tant que clé sur le serveur.

   eXtreme Scale fournit également un plug-in pour générer des données d'identification. Ce plug-in est représenté par l'interface com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator et est utilisé pour générer des données d'identification client. Cela est utile en cas de données d'identification temporaires. Dans ce cas, la méthode getCredential() est appelée pour renouveler les données d'identification. Pour plus d'informations, référez-vous à la documentation sur l'API CredentialGenerator.

Vous pouvez implémenter ces deux interfaces pour l'exécution du client eXtreme Scale afin d'obtenir les données d'identification client.

Cet exemple utilise les deux extraits d'implémentation de plug-in suivants fournis par eXtreme Scale.

```
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredential
```

```
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator
```

Pour plus d'informations sur ces plug-ins, voir Programmation de l'authentification de client.

2. **Authentificateur de serveur** Après que le client eXtreme Scale a récupéré l'objet Credential par le biais de l'objet CredentialGenerator, cet objet Credential est envoyé en même temps de la requête client au serveur eXtreme Scale. Le serveur eXtreme Scale authentifie l'objet Credential avant de traiter la requête. Si l'objet Credential est authentifié, un objet Subject est renvoyé pour représenter ce client.

   Cet objet Subject est alors mis en cache et expire lorsque le délai d'expiration de la session est atteint. Ce délai peut être défini par le biais de la propriété loginSessionExpirationTime dans le fichier XML du cluster. Par exemple, le paramètre `loginSessionExpirationTime="300"` entraîne l'expiration de l'objet Subject dans 300 secondes.Cet objet Subject est alors utilisé pour autoriser la requête, ce qui sera illustré plus loin.

   Un serveur eXtreme Scale utilise le plug-in Authenticator pour authentifier l'objet Credential. Référez-vous à la documentation sur l'API de l'Authenticator pour plus de détails.

   Cet exemple utilise une implémentation pré-intégrée à eXtreme Scale, KeyStoreLoginAuthenticator, utilisée à des fins de test et d'exemple (un fichier de clés est un registre d'utilisateurs et ne doit pas être utilisé pour la production). Pour plus d'informations, consultez la rubrique relative au plug-in Authenticator Programmation de l'authentification de client.

   Cet KeyStoreLoginAuthenticator utilise un KeyStoreLoginModule pour authentifier l'utilisateur avec le fichier de clés par le biais du module de connexion JAAS "KeyStoreLogin". Le fichier de clés peut être configuré en tant qu'option de la classe KeyStoreLoginModule. L'exemple suivant illustre l'alias keyStoreLogin configuré dans le fichier de configuration JAAS og_jaas.config :

   ```
   KeyStoreLogin{
   com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginModule required
        keyStoreFile="../security/sampleKS.jks" debug = true;
   };
   ```

   Les commandes suivantes créent un fichier de clés sampleKS.jks dans le répertoire %OBJECTGRID_HOME%/security avec le mot de passe sampleKS1. De plus, ces trois certificats utilisateur représentant les utilisateurs administrator, manager et cashier sont créés avec leur propre mot de passe.

   a. Placez-vous dans le répertoire racine eXtreme Scale.

      ```
      cd objectgridRoot
      ```

   b. Créez un répertoire nommé "security".

      ```
      mkdir security
      ```

   c. Placez-vous dans le répertoire security que vous venez de créer.

      ```
      cd security
      ```

   d. Utilisez la commande keytool (sous le répertoire `javaHOME/bin`) pour créer un utilisateur "administator" avec le mot de passe "administrator1" dans le fichier de clés sampleKS.jks.

      ```
      keytool -genkey -v -keystore ./sampleKS.jks -storepass sampleKS1
      -alias administrator -keypass administrator1
      -dname CN=administrator,O=acme,OU=OGSample -validity 10000
      ```

e. Utilisez la commande keytool (sous le répertoire javaHOME/bin) pour créer un utilisateur "administrator" avec le mot de passe "administrator1" dans le fichier de clés sampleKS.jks.

```
keytool -genkey -v -keystore ./sampleKS.jks -storepass sampleKS1
-alias manager -keypass manager1
-dname CN=manager,O=acme,OU=OGSample -validity 10000
```

f. Utilisez la commande keytool (sous le répertoire javaHOME/bin) pour créer un utilisateur "cashier" avec le mot de passe "cashier1" dans le fichier de clés sampleKS.jks.

```
keytool -genkey -v -keystore ./sampleKS.jks -storepass sampleKS1
-alias cashier -keypass cashier1 -dname CN=cashier,O=acme,OU=OGSample
-validity 10000
```

La configuration des paramètres de sécurité client est définie dans le fichier de propriétés client. Utilisez la commande suivante pour créer une copie dans le répertoire %OBJECTGRID_HOME%/security :

a. Placez-vous dans le répertoire security.

```
cd objectgridRoot/security
```

b. Copiez le fichier sampleClient.properties vers le fichier client.properties.

```
cp ../properties/sampleClient.properties client.properties
```

Les propriétés suivantes sont mises en évidence dans le fichier client.properties situé sous le répertoire security.

a. **securityEnabled :** définissez securityEnabled sur true (valeur par défaut) ; cela active la sécurité client, ce qui inclut l'authentification.

b. **credentialAuthentication :** définissez credentialAuthentication sur Supported (valeur par défaut) ; cela signifie que le client prend en charge l'authentification des données d'identification.

c. **transportType :** définissez transportType sur TCP/IP, ce qui signifie qu'aucune couche Secure Sockets Layer ne sera utilisée.

d. **singleSignOnEnabled :** définissez ce paramètre sur false (valeur par défaut). La connexion unique (Single sign-on) n'est pas disponible.

3. **Configuration des paramètres de sécurité du serveur**

La configuration des paramètres de sécurité du serveur est définie dans le fichier XML du descripteur de sécurité et dans le fichier des propriétés de sécurité du serveur.Le fichier XML du descripteur de sécurité décrit les propriétés de sécurité communes à tous les serveurs (y compris les serveurs de catalogue et les serveurs de conteneur). Un exemple de propriété est la configuration de l'authentificateur qui représente le registre utilisateur et le mécanisme d'authentification.

Voici le fichier security.xml à utiliser pour cet exemple :

```
<?xml version="1.0" encoding="UTF-8"?>
<securityConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config/security ../objectGridSecurity.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config/security">

 <security securityEnabled="true" loginSessionExpirationTime="300" >

        <authenticator className ="com.ibm.websphere.objectgrid.security.plugins.builtins
.KeyStoreLoginAuthenticator">
        </authenticator>
    </security>

</securityConfig>
```

a. **securityEnabled :** définissez ce paramètre sur true, ce qui active la sécurité du serveur, dont la fonction d'authentification.

b. **loginSessionExpirationTime :** définissez la valeur sur 300 (valeur par défaut).

c. **authenticator :** ajoutez la classe d'authentificateur KeyStoreLoginAuthenticator au fichier XML du cluster, comme suit :

```
<authenticator className ="com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginAuthenticator">
        </authenticator>
```

d. **credentialAuthentication :** définissez l'attribut credentialAuthentication sur Required afin d'activer l'authentification sur le serveur

Pour plus d'informations concernant le fichier `security.xml`, voir Fichier XML du descripteur de sécurité.

Copiez le fichier des propriétés du serveur dans le répertoire security. Pour le moment, aucune modification n'est requise dans le fichier.

a. Placez-vous dans le répertoire security.

```
cd objectgridRoot/security
```

b. Copiez le fichier d'exemple `sampleServer.properties` de l'objectGrid, du répertoire property vers le fichier `server.properties`.

```
cp ../properties/containerServer.properties server.properties
```

Effectuez les modifications suivantes dans le fichier `server.properties` :

a. **securityEnabled :** définissez l'attribut **`securityEnabled`** sur true.

b. **transportType :** définissez l'attribut **`transportType`** sur TCP/IP, ce qui signifie qu'aucune couche Secure Sockets Layer ne sera utilisée.

c. **secureTokenManagerType :** définissez l'attribut **`secureTokenManagerType`**sur sur none pour ne pas configurer le gestionnaire des jetons sécurisés.

4. **Client sécurisé** Connectez l'application client au serveur en toute sécurité, tel qu'illustré dans l'exemple suivant :

```
package com.ibm.websphere.objectgrid.security.sample.guide;

import com.ibm.websphere.objectgrid.ClientClusterContext;
import com.ibm.websphere.objectgrid.ObjectGrid;
import com.ibm.websphere.objectgrid.ObjectGridManager;
import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
import com.ibm.websphere.objectgrid.security.config.ClientSecurityConfiguration;
import com.ibm.websphere.objectgrid.security.config.ClientSecurityConfigurationFactory;
import com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator;
import com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator;

public class SecureSimpleApp extends SimpleApp {

    public static void main(String[] args) throws Exception {

        SecureSimpleApp app = new SecureSimpleApp();
        app.run(args);
    }

    /**
     * Get the ObjectGrid
     * @return an ObjectGrid instance
     * @throws Exception
     */
    protected ObjectGrid getObjectGrid(String[] args) throws Exception {
        ObjectGridManager ogManager = ObjectGridManagerFactory.getObjectGridManager();
        ogManager.setTraceFileName("logs/client.log");
        ogManager.setTraceSpecification("ObjectGrid*=all=enabled:ORBRas=all=enabled");

        // crée un objet ClientSecurityConfiguration à l'aide du fichier spécifié
        ClientSecurityConfiguration clientSC = ClientSecurityConfigurationFactory
                .getClientSecurityConfiguration(args[0]);

        // crée un CredentialGenerator en utilisant le nom de l'utilisateur
           et le mot de passe fournis.
        CredentialGenerator credGen = new UserPasswordCredentialGenerator(args[1], args[2]);
        clientSC.setCredentialGenerator(credGen);

        // crée un ObjectGrid en se connectant au serveur de catalogue
        ClientClusterContext ccContext = ogManager.connect("localhost:2809", clientSC, null);
        ObjectGrid og = ogManager.getObjectGrid(ccContext, "accounting");

        return og;
```

```
    }
}
```

Trois éléments diffèrent par rapport à l'application non sécurisée :

a. A créé un objet ClientSecurityConfiguration en fournissant le fichier `client.properties` configuré.

b. A créé un UserPasswordCredentialGenerator en utilisant l'ID utilisateur et le mot de passe fournis.

c. S'est connecté au serveur de catalogue pour obtenir un ObjectGrid auprès du ClientClusterContext en fournissant un objet ClientSecurityConfiguration.

5. **Exécution de l'application**

Pour exécuter l'application, démarrez le serveur de catalogue. Utilisez les options -clusterFile et -serverProps de la ligne de commande pour indiquer les propriétés de sécurité :

a. Placez-vous dans le répertoire bin :

```
cd objectgridRoot/bin
```

b. Lancez le serveur de catalogue :

- UNIX Linux

```
startOgServer.sh catalogServer -clusterSecurityFile ../security/security.xml
-serverProps ../security/server.properties -jvmArgs
-Djava.security.auth.login.config="../security/og_jaas.config"
```

- Windows

```
startOgServer.bat catalogServer -clusterSecurityFile ../security/security.xml
-serverProps ../security/server.properties -jvmArgs
-Djava.security.auth.login.config="../security/og_jaas.config"
```

Lancez ensuite un serveur de conteneur sécurisé en utilisant le script suivant :

a. Placez-vous de nouveau dans le répertoire bin :

```
cd objectgridRoot/bin
```

b. Lancez un serveur de conteneur sécurisé :

- Linux UNIX

```
startOgServer.sh c0 -objectgridFile ../xml/SimpleApp.xml
-deploymentPolicyFile ../xml/SimpleDP.xml
-catalogServiceEndPoints localhost:2809
-serverProps ../security/server.properties
-jvmArgs -Djava.security.auth.login.config="../security/og_jaas.config"
```

- Windows

```
startOgServer.bat c0 -objectgridFile ../xml/SimpleApp.xml
-deploymentPolicyFile ../xml/SimpleDP.xml
-catalogServiceEndPoints localhost:2809
-serverProps ../security/server.properties
-jvmArgs -Djava.security.auth.login.config="../security/og_jaas.config"
```

Le fichier de propriétés du serveur est indiqué par le biais de l'option -serverProps.

Une fois le serveur démarré, démarrez le client en utilisant la commande suivante :

a. `cd objectgridRoot/bin`

b.

```
java -classpath ../lib/objectgrid.jar;../applib/secsample.jar
  com.ibm.websphere.objectgrid.security.sample.guide.SecureSimpleApp
  ../security/client.properties manager manager1
```

Linux Utilisez le signe deux-points (:) pour le séparateur de chemins d'accès aux classes au lieu du point-virgule (;) comme dans l'exemple précédent.

Le fichier `secsample.jar` contient la classe SimpleApp.

L'application SecureSimpleApp utilise les trois paramètres suivants :

a. Le fichier `../security/client.properties` est le fichier de propriétés de la sécurité client.

b. `manager` est l'ID utilisateur.

c. `manager1` est le mot de passe.

Une fois la classe publiée, la sortie est la suivante :

Le nom du client pour `ID 0001 est fName lName`.

Vous pouvez également utiliser l'utilitaire **xscmd** pour afficher les tailles de mappes de la grille "accounting".

- Placez-vous dans le répertoire `objectgridRoot/bin`.
- Utilisez la commande **xscmd** avec l'option -c showMapSizes commande, comme suit.
  - UNIX Linux `xscmd.sh -c showMapSizes -g accounting -m mapSet1 -username manager -password manager1`
  - Windows `xscmd.bat -c showMapSizes -g accounting -m mapSet1 -username manager -password manager1`

A présent, vous pouvez utiliser la commande **stopOgServer** pour arrêter le processus serveur de conteneur ou service de catalogue. Il est néanmoins nécessaire de fournir un fichier de configuration des paramètres de sécurité. L'exemple de fichier de propriétés du client définit les deux propriétés suivantes pour générer un ID d'utilisateur et un mot de passe (manager/manager1).

```
credentialGeneratorClass=com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator
credentialGeneratorProps=manager manager1
```

Arrêtez le conteneur c0 avec la commande suivante.

- UNIX Linux `stopOgServer.sh c0 -catalogServiceEndPoints localhost:2809 -clientSecurityFile ..\security\client.properties`
- Windows `stopOgServer.bat c0 -catalogServiceEndPoints localhost:2809 -clientSecurityFile ..\security\client.properties`

Si vous ne définissez pas l'option **-clientSecurityFile**, l'exception suivante se produit avec le message suivant.

```
>> SERVER (id=39132c79, host=9.10.86.47) TRACE START:
>> org.omg.CORBA.NO_PERMISSION : le serveur requiert une
authentification par données d'identification mais aucun contexte de
sécurité n'est fourni par le client. Cela est généralement dû au fait
que le client ne fournit pas de données d'identification au serveur.
vmcid: 0x0
code mineur : 0
terminé : non
```

Vous pouvez également arrêter le serveur de catalogue à l'aide de la commande suivante. Cependant, si vous souhaitez effectuer la prochaine étape du tutoriel, vous pouvez maintenir l'exécution du serveur de catalogue.

- UNIX Linux `stopOgServer.sh catalogServer -catalogServiceEndPoints localhost:2809 -clientSecurityFile ..\security\client.properties`
- Windows `stopOgServer.bat catalogServer -catalogServiceEndPoints localhost:2809 -clientSecurityFile ..\security\client.properties`

Si vous arrêtez le serveur de catalogue, la sortie suivante s'affiche.

```
CWOBJ2512I: ObjectGrid server catalogServer stopped
```

Votre système est à présent partiellement sécurisé grâce à l'activation de l'authentification. Vous avez configuré le serveur pour activer le registre utilisateur, configuré le client pour fournir des données d'identification client et modifié le fichier de propriétés du client et le fichier XML du cluster pour activer l'authentification.

Si vous avez fourni un mot de passe non valide, une exception s'affiche et indique que le nom d'utilisateur ou le mot de passe est incorrect.

Pour plus d'informations sur l'authentification du client, voir «Authentification du client d'application», à la page 507.

Etape suivante du tutoriel

Tutoriel sur la sécurité Java SE - Etape 3

Après avoir authentifié un client, comme dans l'étape précédente, vous pouvez attribuer des privilèges de sécurité par le biais des mécanismes d'autorisation eXtreme Scale.

Avant de commencer

Vous devez avoir terminé «Tutoriel sur la sécurité Java SE - Etape 2», à la page 73 avant d'effectuer cette tâche.

Pourquoi et quand exécuter cette tâche

Au cours de l'étape précédente, vous avez appris à activer l'authentification dans une grille eXtreme Scale. Par conséquent, aucun client non authentifié ne peut se connecter à votre serveur ni soumettre des requêtes à votre système. Toutefois, tous les clients authentifiés possèdent les mêmes permissions ou privilèges liés au serveur, tels que la lecture, l'écriture ou la suppression des données stockées dans les mappes ObjectGrid. Les clients peuvent également soumettre tout type de requête. Cette section explique comment utiliser l'autorisation eXtreme Scale pour attribuer différents privilèges aux utilisateurs authentifiés.

A l'instar de nombreux autres systèmes, eXtreme Scale adopte un mécanisme d'autorisation basé sur les permissions. WebSphere eXtreme Scale permet d'utiliser plusieurs catégories de permission, chacune étant représentée par une classe distincte. Cette rubrique décrit MapPermission. Pour la liste des catégories d'autorisations, voir Programmation d'autorisations client.

Dans WebSphere eXtreme Scale, la classe com.ibm.websphere.objectgrid.security.MapPermission représente les permissions liées aux ressources eXtreme Scale, notamment les méthodes des interfaces ObjectMap ou JavaMap. WebSphere eXtreme Scale définit les chaînes de permission suivantes pour accéder aux méthodes des interfaces ObjectMap et JavaMap :

- read : accorde la permission de lire les données de la mappe.
- write : accorde la permission de mettre à jour les données de la mappe.

- insert : accorde la permission d'insérer les données dans la mappe.
- remove : accorde la permission de supprimer les données de la mappe.
- invalidate : accorde la permission d'invalider les données de la mappe.
- all : accorde les permissions de lire, d'écrire, d'insérer, de supprimer et d'invalider.

L'autorisation est accordée lorsque le client invoque une méthode de l'interface ObjectMap ou JavaMap. Le moteur d'exécution eXtreme Scale vérifie différentes permissions de mappe pour différentes méthodes. Si les permissions nécessaires ne sont pas accordées au client, une AccessControlException se produit.

Ce tutoriel montre comment utiliser l'autorisation JAAS (Java Authentication and Authorization Service) afin d'autoriser plusieurs utilisateurs à accéder à la mappe.

Procédure

1. **Activation de l'autorisation eXtreme Scale**. Pour activer l'autorisation sur l'ObjectGrid, vous devez définir l'attribut securityEnabled sur `true` pour cet ObjectGrid spécifique dans le fichier XML. L'activation de la sécurité sur cet ObjectGrid revient à activer l'autorisation. Utilisez les commandes suivantes pour créer un fichier XML ObjectGrid en activant la sécurité.
   a. Accédez au répertoire `xml`.
      ```
      cd objectgridRoot/xml
      ```
   b. Copiez le fichier `SimpleApp.xml` dans le fichier `SecureSimpleApp.xml`.
      ```
      cp SimpleApp.xml SecureSimpleApp.xml
      ```
   c. Ouvrez le fichier `SecureSimpleApp.xml` et ajoutez `securityEnabled="true"` au niveau de l'ObjectGrid comme indiqué dans la syntaxe XML ci-dessous :
      ```
      <?xml version="1.0" encoding="UTF-8"?>
      <objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
         xmlns="http://ibm.com/ws/objectgrid/config">
          <objectGrids>
              <objectGrid name="accounting" securityEnabled="true">
                  <backingMap name="customer" readOnly="false" copyKey="true"/>
              </objectGrid>
          </objectGrids>
      </objectGridConfig>
      ```
2. **Définition de la politique d'autorisation.** Dans la section d'authentification pré-client, vous avez créé trois utilisateurs dans le fichier de clés : cashier (caissier), manager (gestionnaire) et administrator (administrateur). Dans cet exemple, l'utilisateur "cashier" dispose uniquement de la permission de lecture sur toutes les mappes. L'utilisateur "manager", quant à lui, dispose de toutes les permissions. L'autorisation JAAS est utilisée dans cet exemple. L'autorisation JAAS utilise le fichier de politique d'autorisation pour accorder les permissions aux principaux. Le fichier est défini dans le répertoire de sécurité :
   ```
   grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
       principal javax.security.auth.x500.X500Principal "CN=cashier,O=acme,OU=OGSample" {
       permission com.ibm.websphere.objectgrid.security.MapPermission "accounting.*", "read ";
   };

   grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
       principal javax.security.auth.x500.X500Principal "CN=manager,O=acme,OU=OGSample" {
       permission com.ibm.websphere.objectgrid.security.MapPermission "accounting.*", "all";
   };
   ```

   Remarque :

   - La `base de code "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"` est une URL réservée spécifiquement à l'ObjectGrid. Toutes les permissions ObjectGrid accordées aux principaux utilisent cette base de code spécifique.

- La première déclaration d'attribution accorde la permission de lecture ("read") de mappe à l'utilisateur principal `"CN=cashier,O=acme,OU=OGSample"`, de sorte que le caissier dispose uniquement de la permission de lecture sur toutes les mappes de l'ObjectGrid accounting.
- La deuxième déclaration d'attribution accorde toutes ("all") les permissions à l'utilisateur principal `"CN=manager,O=acme,OU=OGSample"`, de sorte que le gestionnaire dispose de toutes les permissions sur toutes les mappes de l'ObjectGrid accounting.

Vous pouvez désormais démarrer un serveur avec une politique d'autorisation. Le fichier de politique d'autorisation JAAS peut être défini à l'aide de la propriété -D standard : -Djava.security.auth.policy=../security/ogAuth.policy

3. **Exécutez l'application.**

   Après avoir créé les fichiers mentionnés ci-dessus, vous pouvez exécuter l'application.

   Démarrez le serveur de catalogue à l'aide des commandes suivantes. Pour plus d'informations sur le démarrage du service de catalogue, voir «Démarrage d'un service de catalogue autonome», à la page 395.

   a. Accédez au répertoire bin : `cd objectgridRoot/bin`

   b. Démarrez le serveur de catalogue.

      - UNIX Linux
      ```
      startOgServer.sh catalogServer
      -clusterSecurityFile ../security/security.xml -serverProps
      ../security/server.properties -jvmArgs
      -Djava.security.auth.login.config="../security/og_jaas.config"
      ```
      - Windows
      ```
      startOgServer.bat catalogServer -clusterSecurityFile
      ../security/security.xml -serverProps ../security/server.properties
      -jvmArgs -Djava.security.auth.login.config="../security/
      og_jaas.config"
      ```

      Vous avez créé les fichiers `security.xml` et `server.properties` au cours de l'étape précédente de ce tutoriel.

      T

   c. Vous pouvez démarrer un serveur de conteneur sécurisé à l'aide du script suivant. Exécutez le script suivant à partir du répertoire bin :

      - UNIX Linux
      ```
      # startOgServer.sh c0 -objectGridFile
      ../xml/SecureSimpleApp.xml -deploymentPolicyFile
      ../xml/SimpleDP.xml -catalogServiceEndPoints localhost:2809
      -serverProps ../security/server.properties -jvmArgs
      -Djava.security.auth.login.config="../security/og_jaas.config"
      -Djava.security.auth.policy="../security/og_auth.policy"
      ```
      - Windows
      ```
      startOgServer.bat c0 -objectGridFile ../xml/
      SecureSimpleApp.xml -deploymentPolicyFile ../xml/SimpleDP.xml
      -catalogServiceEndPoints localhost:2809 -serverProps
      ../security/server.properties -jvmArgs
      -Djava.security.auth.login.config="../security/og_jaas.config"
      -Djava.security.auth.policy="../security/og_auth.policy"
      ```

Veuillez noter les différences suivantes concernant la commande de démarrage de serveur de conteneur précédente :

- Utilisez le fichier `SecureSimpleApp.xml` au lieu du fichier `SimpleApp.xml`.
- Ajoutez un autre argument `-Djava.security.auth.policy` pour définir le fichier de police d'autorisation JAAS sur le processus de serveur de conteneur.

Utilisez la même commande que dans l'étape précédente du tutoriel :

a. Accédez au répertoire bin.

b. `java -classpath ../lib/objectgrid.jar;../applib/secsample.jar com.ibm.websphere.objectgrid.security.sample.guide.SecureSimpleApp ../security/client.properties manager manager1`

   L'application s'exécute correctement car l'utilisateur "manager" dispose de toutes les permissions sur les mappes de l'ObjectGrid accounting.

   Désormais, utilisez l'utilisateur "cashier" et non "manager" pour lancer l'application client.

c. Accédez au répertoire bin.

d. `java -classpath ../lib/objectgrid.jar;../applib/secsample.jar com.ibm.ws.objectgrid.security.sample.guide.SecureSimpleApp ../security/client.properties cashier cashier1`

Vous obtenez l'exception suivante :

```
Exception in thread "P=387313:O=0:CT" com.ibm.websphere.objectgrid.TransactionException:
rolling back transaction, see caused by exception
 at com.ibm.ws.objectgrid.SessionImpl.rollbackPMapChanges(SessionImpl.java:1422)
  at com.ibm.ws.objectgrid.SessionImpl.commit(SessionImpl.java:1149)
  at com.ibm.ws.objectgrid.SessionImpl.mapPostInvoke(SessionImpl.java:2260)
  at com.ibm.ws.objectgrid.ObjectMapImpl.update(ObjectMapImpl.java:1062)
  at com.ibm.ws.objectgrid.security.sample.guide.SimpleApp.run(SimpleApp.java:42)
 at com.ibm.ws.objectgrid.security.sample.guide.SecureSimpleApp.main(SecureSimpleApp.java:27)
Caused by: com.ibm.websphere.objectgrid.ClientServerTransactionCallbackException:
   Client Services - received exception from remote server:
     com.ibm.websphere.objectgrid.TransactionException: transaction rolled back,
   see caused by Throwable
        at com.ibm.ws.objectgrid.client.RemoteTransactionCallbackImpl.processReadWriteResponse(
            RemoteTransactionCallbackImpl.java:1399)
        at com.ibm.ws.objectgrid.client.RemoteTransactionCallbackImpl.processReadWriteRequestAndResponse(
            RemoteTransactionCallbackImpl.java:2333)
        at com.ibm.ws.objectgrid.client.RemoteTransactionCallbackImpl.commit(RemoteTransactionCallbackImpl.java:557)
        at com.ibm.ws.objectgrid.SessionImpl.commit(SessionImpl.java:1079)
        ... 4 more
Caused by: com.ibm.websphere.objectgrid.TransactionException: transaction rolled back, see caused by Throwable
        at com.ibm.ws.objectgrid.ServerCoreEventProcessor.processLogSequence(ServerCoreEventProcessor.java:1133)
        at com.ibm.ws.objectgrid.ServerCoreEventProcessor.processReadWriteTransactionRequest
     (ServerCoreEventProcessor.java:910)
        at com.ibm.ws.objectgrid.ServerCoreEventProcessor.processClientServerRequest(ServerCoreEventProcessor.java:1285)

        at com.ibm.ws.objectgrid.ShardImpl.processMessage(ShardImpl.java:515)
        at com.ibm.ws.objectgrid.partition.IDLShardPOA._invoke(IDLShardPOA.java:154)
        at com.ibm.CORBA.poa.POAServerDelegate.dispatchToServant(POAServerDelegate.java:396)
        at com.ibm.CORBA.poa.POAServerDelegate.internalDispatch(POAServerDelegate.java:331)
        at com.ibm.CORBA.poa.POAServerDelegate.dispatch(POAServerDelegate.java:253)
        at com.ibm.rmi.iiop.ORB.process(ORB.java:503)
        at com.ibm.CORBA.iiop.ORB.process(ORB.java:1553)
        at com.ibm.rmi.iiop.Connection.respondTo(Connection.java:2680)
        at com.ibm.rmi.iiop.Connection.doWork(Connection.java:2554)
        at com.ibm.rmi.iiop.WorkUnitImpl.doWork(WorkUnitImpl.java:62)
        at com.ibm.rmi.iiop.WorkerThread.run(ThreadPoolImpl.java:202)
        at java.lang.Thread.run(Thread.java:803)
Caused by: java.security.AccessControlException: Access denied (
   com.ibm.websphere.objectgrid.security.MapPermission accounting.customer write)
        at java.security.AccessControlContext.checkPermission(AccessControlContext.java:155)
        at com.ibm.ws.objectgrid.security.MapPermissionCheckAction.run(MapPermissionCheckAction.java:141)
        at java.security.AccessController.doPrivileged(AccessController.java:275)
        at javax.security.auth.Subject.doAsPrivileged(Subject.java:727)
        at com.ibm.ws.objectgrid.security.MapAuthorizer$1.run(MapAuthorizer.java:76)
        at java.security.AccessController.doPrivileged(AccessController.java:242)
        at com.ibm.ws.objectgrid.security.MapAuthorizer.check(MapAuthorizer.java:66)
        at com.ibm.ws.objectgrid.security.SecuredObjectMapImpl.checkMapAuthorization(SecuredObjectMapImpl.java:429)
        at com.ibm.ws.objectgrid.security.SecuredObjectMapImpl.update(SecuredObjectMapImpl.java:490)
        at com.ibm.ws.objectgrid.SessionImpl.processLogSequence(SessionImpl.java:1913)
        at com.ibm.ws.objectgrid.SessionImpl.processLogSequence(SessionImpl.java:1805)
        at com.ibm.ws.objectgrid.ServerCoreEventProcessor.processLogSequence(ServerCoreEventProcessor.java:1011)
        ... 14 more
```

Cette exception se produit car l'utilisateur "cashier" ne dispose pas de permission d'écriture et ne peut donc mettre à jour le client de mappe.

Votre système prend désormais en charge l'autorisation. Vous pouvez définir des politiques d'autorisation pour accorder différentes permissions à différents utilisateurs. Pour plus d'informations sur l'autorisation, voir «Autorisation du client d'application», à la page 509.

Que faire ensuite

Effectuez l'étape suivante du tutoriel. Voir «Tutoriel sur la sécurité Java SE - Etape 4».

Tutoriel sur la sécurité Java SE - Etape 4

L'étape suivante vous explique comment activer une couche de sécurité pour la communication entre les nœuds finaux de votre environnement.

Avant de commencer

Assurez-vous d'avoir terminé le «Tutoriel sur la sécurité Java SE - Etape 3», à la page 79 avant de commencer cette étape.

Pourquoi et quand exécuter cette tâche

La topologie eXtreme Scale prend en charge les protocoles TLS (Transport Layer Security) et SSL (Secure Sockets Layer) pour sécuriser la communication entre les nœuds finaux de l'ObjectGrid (client, serveurs de conteneur et serveurs de catalogue). Cette étape du tutoriel se base sur les étapes précédentes pour activer la sécurité du transport.

Procédure

1. **Création de clés et de fichiers de clés TLS/SSL**

   Afin d'activer la sécurité du transport, vous devez créer un fichier de clés et un fichier de clés certifiées. Cet exercice crée une seule clé et une seule paire fichier de clés-fichier de clés certifiées. Ces fichiers sont utilisés pour les clients, les serveurs de conteneur et les serveurs de catalogue ObjectGrid et sont créés par le biais de la commande keytool du Java Development Kit.

   - *Créer une clé privée dans le fichier de clés*

     ```
     keytool -genkey -alias ogsample -keystore key.jks -storetype JKS
     -keyalg rsa -dname "CN=ogsample, OU=Your Organizational Unit, O=Your
     Organization, L=Your City, S=Your State, C=Your Country" -storepass
     ogpass -keypass ogpass -validity 3650
     ```

     Cette commande permet de créer un fichier de clés key.jks contenant une clé "ogsample". Ce fichier de clés key.jks sera utilisé en tant que fichier de clés SSL.

   - *Exporter le certificat public*

     ```
     keytool -export -alias ogsample -keystore key.jks -file temp.key
     -storepass ogpass
     ```

     Cette commande permet d'extraire et de stocker le certificat public de la clé "ogsample" dans le fichier temp.key.

   - *Importer le certificat public du client dans le fichier de clés certifiées*

     ```
     keytool -import -noprompt -alias ogsamplepublic -keystore trust.jks
     -file temp.key -storepass ogpass
     ```

     Cette commande permet d'ajouter le certificat public au fichier de clés trust.jks. Ce fichier trust.jks est utilisé en tant que fichier de clés certifiées SSL.

2. **Configuration des fichiers de propriétés d'ObjectGrid**

   Au cours de cette étape, vous devez configurer le fichier de propriétés ObjectGrid pour activer la sécurité du transport.

   Tout d'abord, copiez les fichiers key.jks et trust.jks dans le répertoire objectgridRoot/security.

   Définissez les propriétés suivantes dans les fichiers `client.properties` et `server.properties`.

   ```
   transportType=SSL-Required

   alias=ogsample
   contextProvider=IBMJSSE2
   protocol=SSL
   keyStoreType=JKS
   keyStore=../security/key.jks
   keyStorePassword=ogpass
   trustStoreType=JKS
   trustStore=../security/trust.jks
   trustStorePassword=ogpass
   ```

   transportType : la valeur de transportType est définie sur "SSL-Required", ce qui signifie que le transport requiert la couche Secure Sockets Layer. La configuration SSL doit être définie pour tous les points de contact ObjectGrid (clients, serveurs de catalogue et serveurs de conteneur). Tous les transports seront chiffrés.

   Les autres propriétés sont utilisées pour définir les configurations SSL. Pour plus d'informations, voir «Protocole TLS et couche de connexion sécurisée», à la page 515. Veillez à suivre les instructions de cette rubrique pour mettre à jour votre fichier `orb.properties`.

   Assurez-vous de suivre les instructions de cette page pour mettre à jour le fichier `orb.properties`.

   Dans le fichier `server.properties`, vous devez ajouter une propriété supplémentaire clientAuthentication et la définir sur false. Du côté serveur, vous n'avez pas besoin de certifier le client.

   ```
   clientAuthentication=false
   ```

3. **Exécution de l'application**

   Ces commandes sont identiques aux commandes de la section «Tutoriel sur la sécurité Java SE - Etape 3», à la page 79.

   Utilisez les commandes suivantes pour démarrer un serveur de catalogue.

   a. Placez-vous dans le répertoire bin : `cd objectgridRoot/bin`

   b. Démarrez le serveur de catalogue :

   - Linux UNIX

     ```
     startOgServer.sh catalogServer -clusterSecurityFile ../security/security.xml
     -serverProps ../security/server.properties -JMXServicePort 11001
     -jvmArgs -Djava.security.auth.login.config="../security/og_jaas.config"
     ```

   - Windows

     ```
     startOgServer.bat catalogServer -clusterSecurityFile ../security/security.xml
     -serverProps ../security/server.properties -JMXServicePort 11001 -jvmArgs
     -Djava.security.auth.login.config="../security/og_jaas.config"
     ```

     Les fichiers `security.xml` et `server.properties` ont été créés au cours de la procédure de la section «Tutoriel sur la sécurité Java SE - Etape 2», à la page 73.

     Utilisez l'option **-JMXServicePort** pour spécifier explicitement le port JMX pour le serveur. Cette option est nécessaire pour pouvoir utiliser l'utilitaire **xscmd**.

     Démarrez le serveur de conteneur ObjectGrid sécurisé :

c. Placez-vous de nouveau dans le répertoire bin : `cd objectgridRoot/bin`

d.

- Linux UNIX

```
startOgServer.sh c0 -objectGridFile ../xml/SecureSimpleApp.xml
-deploymentPolicyFile ../xml/SimpleDP.xml -catalogServiceEndPoints
localhost:2809 -serverProps ../security/server.properties
-JMXServicePort 11002 -jvmArgs
-Djava.security.auth.login.config="../security/og_jaas.config"
-Djava.security.auth.policy="../security/og_auth.policy"
```

- Windows

```
startOgServer.bat c0 -objectGridFile ../xml/SecureSimpleApp.xml
-deploymentPolicyFile ../xml/SimpleDP.xml -catalogServiceEndPoints localhost:2809
-serverProps ../security/server.properties -JMXServicePort 11002
-jvmArgs -Djava.security.auth.login.config="../security/og_jaas.config"
-Djava.security.auth.policy="../security/og_auth.policy"
```

Notez les différences par rapport à la commande start du précédent serveur de conteneur :

- Utilisez `SecureSimpleApp.xml` à la place des fichiers `SimpleApp.xml`.
- Ajoutez un autre argument -Djava.security.auth.policy pour définir le fichier de règles de l'autorisation JAAS sur le processus du serveur de conteneur.

Exécutez la commande suivante pour l'authentification de client :

a. `cd objectgridRoot/bin`

b.

```
javaHome/java -classpath ../lib/objectgrid.jar;../applib/secsample.jar
com.ibm.websphere.objectgrid.security.sample.guide.SecureSimpleApp
../security/client.properties manager manager1
```

Etant donné que l'utilisateur "manager" bénéficie de droits d'accès à toutes les mappes de l'ObjectGrid accounting, l'application s'exécute correctement.

Vous pouvez utiliser l'utilitaire **xscmd** pour afficher les tailles de mappes de la grille "accounting".

- Placez-vous dans le répertoire `objectgridRoot/bin`.
- Utilisez la commande **xscmd** pour afficher les tailles de mappes :
  - UNIX Linux

    ```
    xscmd.sh -c showMapsizes -g accounting -m mapSet1  -jp 11001 -ssl
    -ts ..\security\trust.jks -tsp ogpass -tst jks
    -user manager -pwd manager1
    ```

  - Windows

    ```
    xscmd.bat -c showMapsizes -g accounting -m mapSet1  -jp 11001 -ssl
    -ts ..\security\trust.jks -tsp ogpass -tst jks
    -user manager -pwd manager1
    ```

  Notez que nous spécifions ici le port JMX du service de catalogue par le biais de -p 11001.

  La sortie suivante s'affiche.

```
Cet utilitaire administratif est fourni à titre d'exemple uniquement
et ne doit pas être considéré en tant que composant pris en charge par WebSphere eXtreme Scale.
Connexion au service de catalogue au localhost:1099
********** Résultats pour la grille - accounting, MapSet - mapSet1 **********
*** Liste des mappes pour c0 ***
Nom de la mappe : customer N° de partition #: 0 Taille de la mappe : 1 Type de fragment : principal
Nombre de serveurs : 1
Nombre de domaines : 1
```

Exécution de l'application avec un fichier de clés incorrect

Si votre fichier de clés certifiées ne contient pas le certificat public de la clé publique dans le fichier de clés, une exception s'affiche et vous indique que la clé n'est pas certifiée.

Pour afficher cela, créez un autre fichier de clés key2.jks.

```
keytool -genkey -alias ogsample -keystore key2.jks -storetype JKS
-keyalg rsa -dname "CN=ogsample, OU=Your Organizational Unit, O=Your
```

```
Organization, L=Your City, S=Your State, C=Your Country" -storepass
ogpass -keypass ogpass -validity 3650
```

Modifiez ensuite server.properties pour faire pointer le fichier de clés vers ce nouveau fichier de clés key2.jks:

```
keyStore=../security/key2.jks
```

Exécutez la commande suivante pour démarrer le serveur de catalogue :

a. Placez-vous dans le répertoire bin : `cd objectgridRoot/bin`

b. Démarrez le serveur de catalogue :

Linux UNIX

```
startOgServer.sh c0 -objectGridFile ../xml/SecureSimpleApp.xml
-deploymentPolicyFile ../xml/SimpleDP.xml -catalogServiceEndPoints localhost:2809
-serverProps ../security/server.properties -jvmArgs
-Djava.security.auth.login.config="../security/og_jaas.config"
-Djava.security.auth.policy="../security/og_auth.policy"
```

Windows

```
startOgServer.bat c0 -objectGridFile ../xml/SecureSimpleApp.xml
-deploymentPolicyFile ../xml/SimpleDP.xml -catalogServiceEndPoints localhost:2809
 -serverProps ../security/server.properties -jvmArgs
-Djava.security.auth.login.config="../security/og_jaas.config"
-Djava.security.auth.policy="../security/og_auth.policy"
```

L'exception suivante s'affiche :

```
Caused by: com.ibm.websphere.objectgrid.ObjectGridRPCException:
    com.ibm.websphere.objectgrid.ObjectGridRuntimeException:
        SSL connection fails and plain socket cannot be used.
```

Enfin, modifiez le fichier `server.properties` pour utiliser de nouveau le fichier `key.jks`.

Tutoriel : Intégration de la sécurité WebSphere eXtreme Scale à WebSphere Application Server

Ce tutoriel explique comment sécuriser un déploiement de serveur WebSphere eXtreme Scale dans un environnement WebSphere Application Server.

Objectifs d'apprentissage

Les objectifs d'apprentissage de ce tutoriel sont les suivants :

- Configurer WebSphere eXtreme Scale pour utiliser les plug-ins d'authentification WebSphere Application Server
- Configurer la sécurité du transport WebSphere eXtreme Scale pour utiliser la configuration WebSphere Application Server CSIv2
- Utiliser l'autorisation JAAS (Java Authentication and Authorization Service) dans WebSphere Application Server
- Utiliser un module de connexion personnalisé pour l'autorisation JAAS de groupe
- Utiliser l'utilitaire WebSphere eXtreme Scale **xscmd** dans l'environnement WebSphere Application Server

Durée

Ce tutoriel dure environ 4 heures.

Présentation : Intégration de la sécurité WebSphere eXtreme Scale à WebSphere Application Server en utilisant des plug-ins WebSphere Application Server Authentication

Dans ce tutoriel, vous intégrez la sécurité WebSphere eXtreme Scale à WebSphere Application Server. Tout d'abord, vous devez configurer l'authentification avec une application Web simple qui utilise les données d'identification de l'utilisateur authentifié à partir de l'unité d'exécution en cours pour se connecter à ObjectGrid. Ensuite, vous examinez le chiffrement des données qui sont transférées entre le client et le serveur avec la sécurité de la couche de transport. Pour accorder aux utilisateurs différents niveaux des autorisations, vous pouvez configurer Java Authentication and Authorization Service (JAAS). Une fois la configuration effectuée, vous pouvez utiliser l'utilitaire **xscmd** pour contrôler les grilles de données et les mappes.

Ce tutoriel suppose que l'ensemble de vos clients, serveurs de conteneur et serveurs de catalogue WebSphere eXtreme Scale sont déployés dans l'environnement WebSphere Application Server.

Objectifs d'apprentissage

Les objectifs d'apprentissage de ce tutoriel sont les suivants :

- Configurer WebSphere eXtreme Scale pour utiliser les plug-ins d'authentification WebSphere Application Server
- Configurer la sécurité du transport WebSphere eXtreme Scale pour utiliser la configuration WebSphere Application Server CSIv2
- Utiliser l'autorisation JAAS (Java Authentication and Authorization Service) dans WebSphere Application Server
- Utiliser un module de connexion personnalisé pour l'autorisation JAAS de groupe
- Utiliser l'utilitaire WebSphere eXtreme Scale **xscmd** dans l'environnement WebSphere Application Server

Durée

Ce tutoriel prend 4 heures environ.

Niveau de compétence

Intermédiaire.

Public ciblé

Développeurs et les administrateurs qui sont intéressés par l'intégration de la sécurité entre WebSphere eXtreme Scale et WebSphere Application Server.

Configuration système requise et topologie

- WebSphere Application Server Version 6.1, Version 7.0.0.11 ou version suivante
- Mettez à jour l'environnement d'exécution Java pour appliquer le correctif suivant : IZ79819: IBMJDK FAILS TO READ PRINCIPAL STATEMENT WITH WHITESPACE FROM SECURITY FILE

Ce tutoriel utilise dans l'exemple quatre serveurs d'applications WebSphere Application Server et un gestionnaire de déploiement.

Prérequis

Une connaissance de base des éléments suivants est utile avant de démarrer ce tutoriel :

- Modèle de programmation WebSphere eXtreme Scale
- Concepts de sécurité de base WebSphere eXtreme Scale
- Concepts de sécurité de base WebSphere Application Server

Pour plus d'informations sur l'intégration de la sécurité WebSphere eXtreme Scale et WebSphere Application Server, voir «Intégration de la sécurité dans WebSphere Application Server», à la page 525.

Module 1 : Préparation de WebSphere Application Server

Avant de commencer le tutoriel d'intégration à WebSphere eXtreme Scale, vous devez créer une configuration de sécurité de base dans WebSphere Application Server.

Objectifs d'apprentissage

A la fin des leçons de ce module, vous saurez :

- configurer la sécurité WebSphere Application Server pour utiliser un fichier interne basé sur un référentiel fédéré sous la forme d'un registre de comptes utilisateur ;
- créer des groupes d'utilisateurs et des utilisateurs ;
- créer des clusters pour l'application et les serveurs WebSphere eXtreme Scale.

Durée

Ce module prend 60 minutes environ.

Leçon 1.1 : Compréhension de la topologie et obtention des fichiers du tutoriel

Pour préparer votre environnement pour le tutoriel, vous devez configurer la sécurité WebSphere Application Server. Vous configurez la sécurité d'administration et d'application en utilisant des référentiels fédérés basés sur un fichier interne comme registre de comptes utilisateur.

Cette leçon vous guide dans l'exemple de topologie et les applications qui sont utilisés dans le tutoriel. Pour commencer à exécuter le tutoriel, vous devez télécharger les applications et placer les fichiers de configuration dans les emplacements propres à votre environnement. Vous pouvez télécharger l'exemple d'application depuis le wiki WebSphere eXtreme Scale.

Exemple de topologie WebSphere Application Server : Ce tutoriel vous guide tout au long de la création de quatre serveurs d'applications WebSphere Application Server pour montrer l'utilisation des exemples d'applications avec la sécurité activée. Ces serveurs d'applications sont regroupés dans deux clusters contenant deux serveurs :

- **Cluster appCluster** : héberge l'exemple d'application d'entreprise EmployeeManagement. Ce cluster contient les deux serveurs d'applications s1 et s2.
- **Clster xsCluster** : héberge les serveurs de conteneur eXtreme Scale. Ce cluster contient les deux serveurs d'applications xs1 et xs2.

Dans cette topologie de déploiement, les serveurs d'applications s1 et s2 sont les serveurs client qui accèdent aux données qui sont stockées dans la grille de données. Les serveurs xs1 et xs2 sont les serveurs de conteneurs qui hébergent la grille de données.

Le serveur de catalogue est déployé dans le processus du gestionnaire de déploiement par défaut. Ce tutoriel utilise le comportement par défaut. L'hébergement du serveur de catalogue dans le gestionnaire de déploiement n'est pas une pratique recommandée dans un environnement de production. Dans un environnement de production, vous devez créer un domaine de services de catalogue pour définir où les serveurs de catalogue démarrent. Pour plus d'informations, voir «Création de domaines de services de catalogue dans WebSphere Application Server», à la page 258.

Autre configuration : vous pouvez héberger tous les serveurs d'applications dans un seul cluster, tel que appCluster. Avec cette configuration, tous les serveurs du cluster sont les clients et les serveurs de conteneur. Ce tutoriel utilise deux clusters pour distinguer les serveurs d'applications qui hébergent les clients et les serveurs conteneurs.

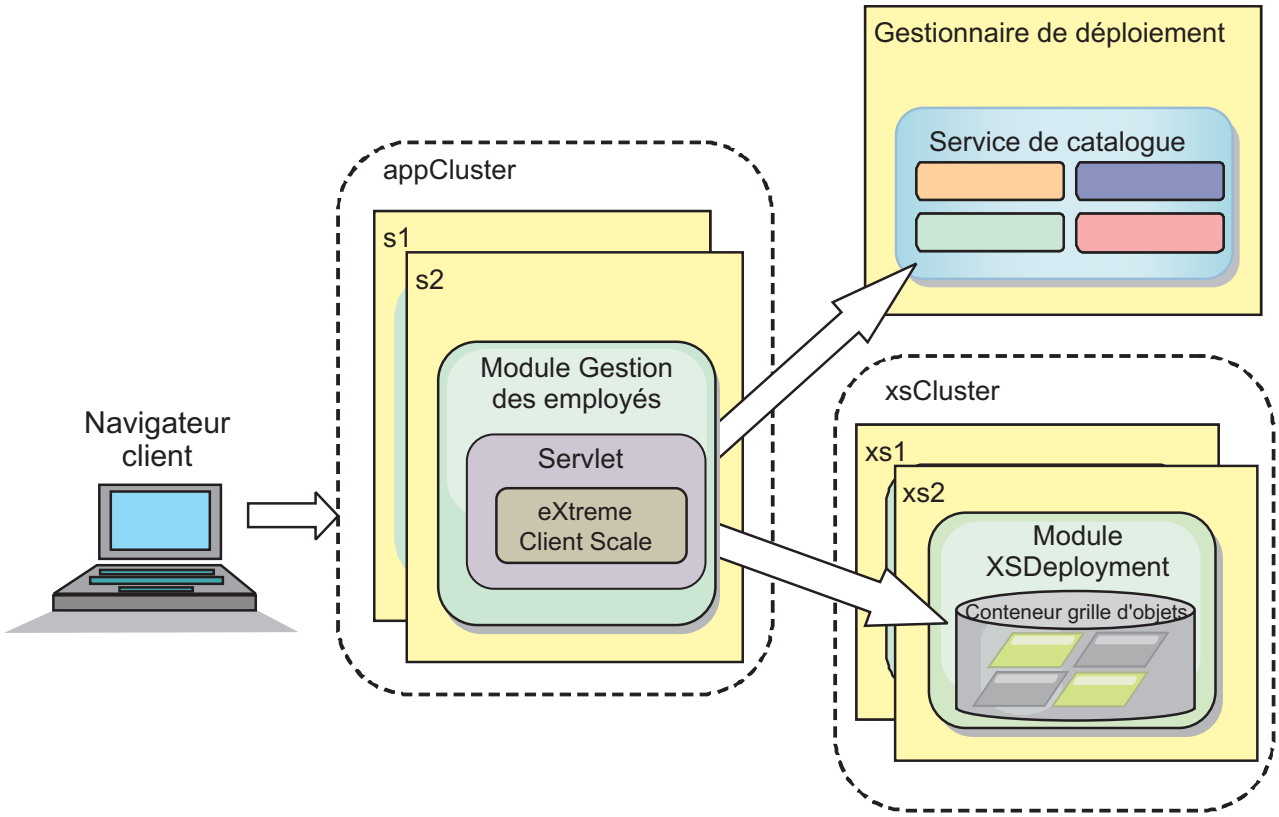


Figure 19. Topologie du tutoriel

Applications : Dans ce tutoriel, vous utilisez deux applications et un fichier de bibliothèque partagée :

- **`EmployeeManagement.ear`** : l'application `EmployeeManagement.ear` est une application d'entreprise simplifiée Java 2 Platform, Enterprise Edition (J2EE). Elle contient un module Web pour gérer les profils des employés. Le module Web

contient le fichier management.jsp pour afficher, insérer, mettre à jour et supprimer les profils d'employés qui sont stockés dans les serveurs de conteneur.

- **XSDeployment.ear** : cette application contient un module d'application d'entreprise, sans artefacts d'application. Les objets cache sont regroupés dans le fichier EmployeeData.jar. Le fichier EmployeeData.jar est déployé comme bibliothèque partagée pour le fichier XSDeployment.ear pour que le fichier XSDeployment.ear puisse accéder aux classes. Cette application à pour fonction de modulariser le fichier de configuration eXtreme Scale. Lorsque cette application d'entreprise est démarrée, les fichiers de configuration eXtreme Scale sont automatiquement détectés par l'environnement d'exécution eXtreme Scale pour créer les serveurs de conteneur. Ces fichiers de configuration incluent les fichiers objectGrid.xml et objectGridDeployment.xml.
- **EmployeeData.jar** : ce fichier JAR contient une classe, com.ibm.websphere.sample.xs.data.EmployeeData. Cette classe représente les données de l'employé qui sont stockés dans la grille. Ce fichier d'archive Java (JAR) est déployé avec les fichiers EmployeeManagement.ear et XSDeployment.ear comme bibliothèque partagée.

Obtention des fichiers du tutoriel :

1. Téléchargez les fichiers WASSecurity.zip et security.zip. Vous pouvez télécharger l'exemple d'application depuis le wiki WebSphere eXtreme Scale.
2. Extrayez le ficher WASSecurity.zip dans un répertoire pour afficher les données binaires et les artefacts source (par exemple un répertoire wxs_samples/). Ce répertoire est *samples_home* pour le reste du tutoriel. Pour la description du contenu du fichier WASSecurity.zip et savoir comment charger le source dans votre espace de travail Eclipse, voir le fichier README.txt dans le package.
3. Extrayez le fichier security.zip vers le répertoire *samples_home* Le fichier security.zip contient les fichiers de configuration de sécurité suivants qui sont utilisés dans ce tutoriel :
   - catServer2.props
   - server2.props
   - client2.props
   - securityWAS2.xml
   - xsAuth2.props

A propos des fichiers de configuration :

Les fichiers objectGrid.xml et objectGridDeployment.xml créent les grilles de données et les mappes qui stockent les données d'application.

Ces fichiers de configuration doivent s'appeler objectGrid.xml et objectGridDeployment.xml. Lorsque le serveur d'applications démarre, eXtreme Scale détecte ces fichiers dans le répertoire META-INF de l'EJB et des modules Web. Si ces fichiers sont trouvés, il suppose que la machine JVM (Java virtual machine) fait office de serveur de conteneur pour les grilles de données définie dans les fichiers de configuration.

Fichier objectGrid.xml

Le fichier objectGrid.xml défini un ObjectGrid nommé Grid. La grille de données Grid a une mappe, la mappe Map1, qui stocke le profil d'employé pour l'application.

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">

 <objectGrids>
        <objectGrid name="Grid" txTimeout="15">
            <backingMap name="Map1" />
        </objectGrid>
    </objectGrids>

</objectGridConfig>
```

Fichier objectGridDeployment.xml

Le fichier objectGridDeployment.xml indique comment déployer la grille de données Grid. Lorsque la grille est déployée, elle dispose de cinq partitions et d'une réplique synchrone.

```
<?xml version="1.0" encoding="UTF-8"?>

<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

    <objectgridDeployment objectgridName="Grid">
        <mapSet name="mapSet" numberOfPartitions="5" minSyncReplicas="0" maxSyncReplicas="1" >
            <map ref="Map1"/>
        </mapSet>
    </objectgridDeployment>

</deploymentPolicy>
```

Point de contrôle de la leçon :

Dans cette leçon, vous avez découvert la topologie du tutoriel et ajouté les fichiers de configuration et les exemples d'applications à votre environnement.

Si vous souhaitez en savoir plus sur le démarrage automatique des serveurs de conteneur, voir «Configuration des applications WebSphere Application Server pour démarrer automatiquement les serveurs de conteneur», à la page 275.

Leçon 1.2 : Configuration de l'environnement WebSphere Application Server

Pour préparer votre environnement pour le tutoriel, vous devez configurer la sécurité WebSphere Application Server. Activez la sécurité d'administration et d'application en utilisant des référentiels fédérés basés sur un fichier interne comme registre de comptes utilisateur. Ensuite, vous pouvez créer des clusters de serveurs pour héberger les serveurs d'applications client et les serveurs de conteneur.

Les étapes suivantes s'appliquent à WebSphere Application Server Version 7.0. Toutefois, vous pouvez appliquez les concepts aux versions antérieures de WebSphere Application Server.

Configuration de la sécurité WebSphere Application Server :

1. Configurez la sécurité WebSphere Application Server.
   a. Dans la console d'administration WebSphere Application Server, cliquez sur **Sécurité** > **Sécurité globale**.
   b. Sélectionnez **Référentiels fédérés** comme **référentiel de comptes utilisateur**. Cliquez sur **Définir comme actif**.
   c. Cliquez sur **Configurer** pour accéder au panneau des **référentiels fédérés**.

d. Entrez le **nom de l'administrateur principal**, tel que `admin`. Cliquez sur **Appliquer.**

e. Lorsque vous y êtes invité, entrez le mot de passe de l'administrateur et cliquez sur **OK**. Sauvegardez vos modifications.

f. Dans la page **Sécurité globale**, vérifiez que le paramètre **Référentiels fédérés** est affecté du registre de comptes utilisateur en cours.

g. Sélectionnez **Activer la sécurité administrative**, **Activer la sécurité de l'application** et **Utiliser la sécurité Java 2 pour limiter l'accès de l'application aux ressources locales**. Cliquez sur **Appliquer** et enregistrez les modifications.

h. Redémarrez le gestionnaire de déploiement et les serveurs d'applications actifs.

La sécurité administrative WebSphere Application Server est activée à l'aide des référentiels fédérés basés sur un fichier interne comme registre de comptes utilisateur.

2. Créez deux groupes d'utilisateurs : adminGroup et operatorGroup.
   a. Cliquez sur **Utilisateurs et groupes** > **Gérer les groupes** > **Créer...**
   b. Entrez `adminGroup` comme nom de groupe. Entrez `Administration de groupe` comme description. Cliquez sur **Créer**.
   c. Cliquez sur **Créer à l'identique**. Entrez `operatorGroup` comme nom de groupe. Entrez `Groupe d'opérateurs` comme description. Cliquez sur **Créer**.
   d. Cliquez sur **Fermer**.
3. Créez les utilisateurs `admin1` et `operator1`.
   a. Cliquez sur **Utilisateurs et groupes** > **Gérer les utilisateurs** > **Créer...**
   b. Créez l'utilisateur `admin1` avec le prénom `Jean` et le nom `Doe` avec le mot de passe `admin1`. Cliquez sur **Créer**.
   c. Créer un second utilisateur. Cliquez sur **Créer à l'identique** pour créer l'utilisateur `operator1` avec le prénom `Jane` et le nom `Doe` avec le mot de passe `operator1`. Cliquez sur **Créer**. Cliquez sur **Fermer**.
4. Ajouter des utilisateurs aux groupes d'utilisateurs. Ajoutez l'utilisateur `admin1` à `adminGroup` et l'utilisateur `operator1` à `operatorGroup`.
   a. Cliquez sur **Utilisateurs et groupes** > **Gérer les utilisateurs**.
   b. Recherchez des utilisateurs à ajouter aux groupes. Cliquez sur **Rechercher..** et définissez un astérisque (*) comme valeur de recherche pour afficher tous les utilisateurs.
   c. Dans le résultat de la recherche, sélectionnez l'utilisateur `admin1` et cliquez sur l'onglet **Groupes**. Cliquez sur **Ajouter** pour ajouter le groupe.
   d. Recherchez les groupes pour identifier les groupes disponibles. Cliquez sur `adminGroup` et sur **Ajouter**.
   e. Répétez ces étapes pour ajouter l'utilisateur `operator1` au groupe d'utilisateurs `operatorGroup`.
5. Sauvegardez vos modifications, déconnectez-vous de la console d'administration, puis redémarrez le gestionnaire de déploiement et l'agent de noeud pour activer les paramètres de sécurité.

Vous avez activé la sécurité et créé des utilisateurs et les groupes d'utilisateurs ont un accès administrateur et opérateur à la configuration WebSphere Application Server.

Création de clusters de serveurs :

Créez deux clusters de serveurs dans votre configuration WebSphere Application Server : `appCluster` pour héberger l'exemple d'application du tutoriel et `xsCluster` pour héberger la grille de données.

1. Dans la console d'administration WebSphere Application Server, ouvrez le panneau des clusters. Cliquez sur **Serveurs** > **Clusters** > **Clusters de serveurs d'applications WebSphere** > **Nouveau**.
2. Entrez `appCluster` comme nom de cluster, ne renseignez pas l'option **Environnement local préféré** et cliquez sur **Suivant**.
3. Créez des serveurs dans le cluster. Créez le serveur `s1` en conservant les options par défaut. Ajoutez le membre `s2` au cluster.
4. Exécutez les étapes restantes dans l'assistant pour créer le cluster. Sauvegardez les modifications.
5. Répétez ces étapes pour créer le cluster `xsCluster`. Ce cluster contient les deux serveurs `xs1` et `xs2`.

Point de contrôle de la leçon :

Vous avez activé la sécurité globale de la cellule WebSphere Application Server, créé des utilisateurs et des groupes d'utilisateurs et créé des clusters pour héberger l'application et la grille de données.

Module 2 : Configuration de WebSphere eXtreme Scale pour utiliser les plug-ins WebSphere Application Server Authentication

Une fois que vous avez créé la configuration WebSphere Application Server, vous pouvez intégrer l'authentification WebSphere eXtreme Scale à WebSphere Application Server.

Lorsqu'un client WebSphere eXtreme Scale se connecte à un serveur conteneur qui exige une authentification, le client doit fournir un générateur de données d'identification représenté par l'interface com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator. Un générateur de données d'identification est une fabrique pour créer des données d'identification de client. Les données d'identification de client peuvent être une paire nom-mot de passe, un ticket Kerberos, un certificat client ou des données d'identification de client dans n'importe quel format convenu entre le client et le serveur. Voir la documentation sur l'API Credential pour plus d'informations. Dans cet exemple, le client WebSphere eXtreme Scale est l'application Web EmployeeManagment qui est déployée dans le cluster appCluster. Les données d'identification de client sont un jeton de sécurité WebSphere qui représente l'identité de l'utilisateur Web.

Objectifs d'apprentissage

A la fin des leçons de ce module, vous saurez :

- configurer la sécurité du serveur du client ;
- configurer la sécurité du serveur de catalogue ;
- configurer la sécurité du serveur de conteneur ;
- installer et exécuter l'exemple d'application.

Durée

Ce module prend 60 minutes environ.

Leçon 2.1 : Configuration de la sécurité du serveur de conteneur

Le fichier des propriétés du client indique la classe d'implémentation CredentialGenerator à utiliser.

Configurez le fichier de propriétés du client avec la propriété JVM **-Dobjectgrid.client.props**. Le nom de fichier spécifié pour cette propriété est un chemin absolu, tel que *samples_home*/security/client2.props. Voir Fichier de propriétés du client pour plus d'informations sur le fichier des propriétés du client.

Contenu du fichier des propriétés du client :

Cet exemple utilise des jetons de sécurité WebSphere Application Server comme données d'identification. Le fichier client2.props se trouve dans le répertoire *samples_home*/security. Le fichier client2.props contient les paramètres suivants :

securityEnabled
Lorsque la valeur est true, elle indique que le client doit envoyer les informations de sécurité disponibles au serveur.

credentialAuthentication
Lorsque la valeur est Supported, elle indique que le client prend en charge l'authentification des données d'identification.

credentialGeneratorClass
Indique la classe d'implémentation de com.ibm.websphere.objectgrid.security.plugins.builtins. Classe WSTokenCredentialGenerator pour que le client puisse extraire les jetons de sécurité de l'unité d'exécution. Voir «Intégration de la sécurité dans WebSphere Application Server», à la page 525 pour plus d'informations sur les jetons de sécurité extraits.

Définition du fichier des propriétés du client en utilisant des propriétés JVM (Java virtual machine) :

Dans la console d'administration, procédez comme suit pour les serveurs s1 et s2 dans le cluster appCluster. Si vous utilisez une topologie différente, procédez comme suit pour tous les serveurs d'applications sur lesquels l'application EmployeeManagement est déployée.

1. **Serveurs** > **Serveurs d'applications WebSphere** > *server_name* > **Java et gestion de processus** > **Définition de processus** > **Java Virtual Machine**.
2. Créez la propriété JVM générique suivante pour définir l'emplacement du fichier de propriétés du client :
   -Dobjectgrid.client.props=*samples_home*/security/client2.props
3. Cliquez sur **OK** et enregistrez les modifications.

Point de contrôle de la leçon :

Vous avez édité le fichier de propriétés du client et configuré les serveurs dans le cluster appCluster pour utiliser le fichier des propriétés du client. Le fichier des propriétés indique la classe d'implémentation CredentialGenerator à utiliser.

Leçon 2.2 : Configuration de la sécurité du serveur de catalogue

Propriétés de sécurité communes à tous les serveurs WebSphere eXtreme Scale, y compris les serveurs de service de catalogue et de conteneur et les propriétés de sécurité du serveur de catalogue.

Les propriétés de sécurité qui sont communes aux serveurs de catalogue et aux serveurs de conteneur sont configurées dans le fichier descripteur XML de sécurité. La configuration de l'authentificateur, qui représente le registre d'utilisateurs et le mécanisme d'authentification, est un exemple des propriétés communes. Voir Fichier XML du descripteur de sécurité pour plus d'informations sur les propriétés de sécurité.

Pour configurer le fichier descripteur XML de sécurité, créez une propriété -Dobjectgrid.cluster.security.xml.url dans l'argument de machine virtuelle Java. Le nom de fichier spécifié pour cette propriété doit avoir le format URL, tel que `file:///samples_home/security/securityWAS2.xml`.

Fichier `securityWAS2.xml` :

Dans ce tutoriel, le fichier `securityWAS2.xml` se trouve dans le répertoire *samples_home*`/security`. Contenu du fichier `securityWAS2.xml` avec les commentaires supprimés :

```
<securityConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config/security ../objectGridSecurity.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config/security">

 <security securityEnabled="true">
  <authenticator
 className="com.ibm.websphere.objectgrid.security.plugins.builtins.WSTokenAuthenticator">
  </authenticator>
 </security>
</securityConfig>
```

Les propriétés suivantes sont définies dans le fichier `securityWAS2.xml` :

securityEnabled
: La propriété securityEnabled a la valeur `true` pour indiquer au serveur de catalogue que la sécurité globale WebSphere eXtreme Scale est activée.

authenticator
: L'authentificateur est configuré comme suit : com.ibm.websphere.objectgrid.security.plugins.builtins. Classe WSTokenAuthenticator. Avec cette implémentation intégrées dans le plug-in Authenticator, le serveur WebSphere eXtreme Scale peut convertir les jetons de sécurité en objet Subject. Voir «Intégration de la sécurité dans WebSphere Application Server», à la page 525 pour plus d'informations sur la conversion des jetons de sécurité.

Fichier `catServer2.props` :

Le fichier de propriétés du serveur stocke les propriétés du serveur qui incluent ses propriétés du serveur. Pour plus d'informations, voir Fichier de propriétés du serveur. Vous pouvez configurer le fichier de propriétés du serveur avec la propriété `-Dobjectgrid.server.props` dans l'argument JVM. Définissez la valeur de nom de fichier de cette propriété dans un chemin absolu, tel que *samples_home*`/security/catServer2.props`. Pour ce tutoriel, un fichier `catServer2.props` est inclus dans le répertoire *samples_home*`/security`. Contenu du fichier `catServer2.props` avec les commentaires supprimés :

securityEnabled
: La propriété securityEnabled a la valeur `true` pour indiquer que le serveur est un serveur sécurisé.

credentialAuthentication
La propriété credentialAuthentication a la valeur `Requis`. Par conséquent, un client qui se connecte au serveur doit fournir des données d'identification.

secureTokenManagerType
La propriété secureTokenManagerType a la valeur `none` pour indiquer que la valeur secrète d'authentification n'est pas chiffrée lors du regroupement avec les serveurs existants.

authenticationSecret
La propriété authenticationSecret a la valeur `ObjectGridDefaultSecret`. Cette chaîne secrète est utilisée pour devenir membre du cluster de serveurs eXtreme Scale. Lorsqu'un serveur rejoint la grille de données, il est invité à présenter la chaîne secrète. Si la chaîne secrète du serveur qui rejoint le cluster correspond à la chaîne dans le serveur de catalogue, le serveur devient membre du cluster. Dans le cas contraire, la demande de jointure est rejetée.

transportType
La propriété transportType a la valeur `TCP/IP` initialement. Plus loin dans le tutoriel, la sécurité du transport est activée.

Définition du fichier des propriétés du serveur avec des propriétés JVM :

Définissez le fichier des propriétés de serveur sur le serveur du gestionnaire de déploiement. Si vous utilisez une topologie différente de celle du tutoriel, définissez le fichier de propriétés sur tous les serveurs d'applications que vous utilisez pour héberger les serveurs de conteneur.

1. Ouvrez la configuration de la machine virtuelle Java du serveur. Dans la console d'administration, cliquez sur **Administration du système** > **Gestionnaire de déploiement** > **Java et gestion de processus** > **Définition de processus** > **Java Virtual Machine**.
2. Ajoutez les arguments JVM génériques suivants :
   ```
   -Dobjectgrid.cluster.security.xml.url=file:///samples_home/security/securityWAS2.xml
   -Dobjectgrid.server.props=samples_home/security/catServer2.props
   ```
3. Cliquez sur **OK** et enregistrez les modifications.

Point de contrôle de la leçon :

Vous avez configuré la sécurité du serveur de catalogue en associant les fichiers `securityWAS2.xml` et `catServer2.props` au gestionnaire de déploiement qui héberge le processus serveur de catalogue dans la configuration WebSphere Application Server.

Leçon 2.3 : Configuration de la sécurité du serveur de conteneur

Lorsqu'un serveur de conteneur se connecte au service de catalogue, le serveur de conteneur obtient toutes les configurations de sécurité qui sont configurées dans le fichier XML Object Grid Security XML, telles que la configuration de l'authentificateur, la valeur de temporisation de la session de connexion et d'autres informations de configuration. Un serveur de conteneur dispose également de ses propres propriétés de sécurité dans le fichier de propriétés du serveur.

Configurez le fichier de propriétés du serveur avec la propriété JVM `-Dobjectgrid.server.props` (Java virtual machine). Le nom de fichier spécifié pour cette propriété est un chemin absolu, tel que *samples_home*`/security/server2.props`.

Dans ce tutoriel, les serveurs de conteneur sont hébergés dans les serveurs `xs1` et `xs2` du cluster `xsCluster`.

Fichier `server2.props` :

Le fichier `server2.props` se trouve dans le répertoire *samples_home*`/security` du répertoire `WASSecurity`. Les propriétés suivantes sont définies dans le fichier `server2.props` :

securityEnabled
La propriété securityEnabled a la valeur `true` pour indiquer que ce serveur de conteneur est un serveur sécurisé.

credentialAuthentication
La propriété credentialAuthentication a la valeur `Required`. Par conséquent, un client qui se connecte au serveur doit fournir des données d'identification.

secureTokenManagerType
La propriété secureTokenManagerType a la valeur `none` pour indiquer que la valeur secrète d'authentification n'est pas chiffrée lors de la connexion aux serveurs existants.

authenticationSecret
La propriété authenticationSecret a la valeur `ObjectGridDefaultSecret`. Cette chaîne secrète est utilisée pour devenir membre du cluster de serveurs eXtreme Scale. Lorsqu'un serveur devient membre de la grille de données, il est invité à présenter la chaîne secrète. Si la chaîne secrète du serveur qui rejoint le cluster correspond à la chaîne dans le serveur de catalogue, le serveur devient membre du cluster. Dans le cas contraire, la demande de jointure est rejetée.

Définition du fichier de propriétés du serveur avec des propriétés JVM :

Définition du fichier de propriétés des serveurs xs1 et xs2. Si vous n'utilisez pas la topologie du tutoriel, définissez le fichier de propriétés de serveur sur tous les serveurs d'applications que vous utilisez pour héberger les serveurs de conteneur.

1. Ouvrez la page de la machine virtuelle Java du serveur. **Serveurs** > **Serveurs d'applications** > *server_name* > **Java et gestion de processus** > **Définition de processus** > **Java Virtual Machine**
2. Ajoutez les arguments JVM génériques suivants :
   `-Dobjectgrid.server.props=`*samples_home*`/security/server2.props`
3. Cliquez sur **OK** et enregistrez les modifications.

Point de contrôle de la leçon :

A présent, l'authentification du serveur WebSphere eXtreme Scale est sécurisée. En configurant cette sécurité, toutes les applications qui tentent de se connecter aux serveurs WebSphere eXtreme Scale doivent fournir des données d'identification. Dans ce tutoriel, WSTokenAuthenticator est l'authentificateur. En conséquence, le client doit fournir un jeton de sécurité WebSphere Application Server.

Leçon 2.4 : Installation et exécution de l'exemple

Une fois l'authentification configurée, vous pouvez installer et exécuter l'exemple d'application.

Création d'une bibliothèque partagée pour le fichier `EmployeeData.jar` :

1. Dans la console d'administration de WebSphere Application Server, ouvrez la page **Bibliothèques partagées**. Cliquez sur **Environnement** > **Bibliothèques partagées**.
2. Choisissez la portée **cellule**.
3. Créez la bibliothèque partagée. Cliquez sur **Nouveau**. Entrez `EmployeeManagementLIB` pour le **nom**. Entrez le chemin d'accès au fichier `EmployeeData.jar` dans le chemin de classes, par exemple, *samples_home*`/WASSecurity/EmployeeData.jar`.
4. Cliquez sur **Appliquer**.

Installation de l'exemple :

1. Installez le fichier `EmployeeManagement.ear`.
   a. Pour commencer l'installation, cliquez sur **Applications** > **Nouvelle application** > **Nouvelle application d'entreprise**. Choisissez le chemin détaillé d'installation de l'application.
   b. Dans l'étape d'**association des modules aux serveurs**, définissez le cluster `appCluster` pour installer le module EmployeeManagementWeb.
   c. Dans l'étape d'**association des bibliothèques partagées**, sélectionnez le module `EmployeeManagementWeb`.
   d. Cliquez sur **Bibliothèques partagées de référence**. Sélectionnez la bibliothèque `EmployeeManagementLIB`.
   e. Associez le rôle `webUser` à **Tous authentifiés dans le domaine de l'application**.
   f. Cliquez sur **OK**.

   Les clients s'exécutent dans les serveurs `s1` et `s2` du cluster.
2. Installez l'exemple de fichier `XSDeployment.ear`.
   a. Pour commencer l'installation, cliquez sur **Applications** > **Nouvelle application** > **Nouvelle application d'entreprise**. Choisissez le chemin détaillé pour l'installation de l'application.
   b. Dans l'étape d'**association des modules aux serveurs**, définissez le cluster `xsCluster` pour installer le module Web `XSDeploymentWeb`.
   c. Dans l'étape d'**association des bibliothèques partagées**, sélectionnez le module `XSDeploymentWeb`.
   d. Cliquez sur **Bibliothèques partagées de référence**. Sélectionnez la bibliothèque `EmployeeManagementLIB`.
   e. Cliquez sur **OK**.

   Les serveurs `xs1` et `xs2` de ce cluster hébergent les serveurs de conteneur.
3. Redémarrez le gestionnaire de déploiement. Lorsque le gestionnaire de déploiement démarre, le serveur de catalogue démarre également. Si vous examinez le fichier `SystemOut.log` du gestionnaire de déploiement, vous pouvez voir le message suivant qui indique que le fichier des propriétés du serveur eXtreme Scale est chargé.

   ```
   CWOBJ0913I: Server property files have been loaded:
   /wxs_samples/security/catServer2.props.
   ```
4. Redémarrez le cluster xsCluster. Lorsque xsCluster démarre, l'application `XSDeployment` démarre et un serveur de conteneur est démarré sur les serveurs xs1 et xs2 respectivement. Si vous examinez le fichier `SystemOut.log` du gestionnaire de déploiement, vous pouvez voir le message suivant qui indique que le fichier des propriétés du serveur est chargé :

   ```
   CWOBJ0913I: Server property files have been loaded:
   /wxs_samples/security/server2.props.
   ```

5. Redémarrez le cluster appClusters. Lorsque le cluster appCluster démarre, l'application EmployeeManagement démarre également. Si vous examinez le fichier `SystemOut.log` des serveurs s1 et s2, le message suivant indique que le fichiers des propriétés est chargé.

   ```
   CWOBJ0924I: The client property file {0} has been loaded.
   ```

   Vous pouvez ignorer les messages d'avertissement concernant les propriétés authenticationRetryCount, transportType et clientCertificateAuthentication. Les valeurs par défaut doivent être utilisées, car les valeurs n'ont pas été indiquées dans le fichier des propriétés.Si vous utilisez WebSphere eXtreme Scale Version 7.0, le message CWOBJ9000I s'affiche en anglais et indique que le fichier des propriétés du client a été chargé. Si vous ne voyez pas le message attendu, vérifiez que vous avez configuré la propriété -Dobjectgrid.server.props ou -Dobjectgrid.client.props dans l'argument JVM. Si vous l'avez configurée, vérifiez que le tiret (-) est un caractère UTF.

Exécution de l'exemple d'application :

1. Exécutez le fichier `management.jsp`.. Dans un navigateur Web, accédez à `http://<your_servername>:<port>/EmployeeManagementWeb/management.jsp`. Par exemple, vous pouvez utiliser l'URL `http://localhost:9080/EmployeeManagementWeb/management.jsp`.
2. Fournissez les informations d'authentification à l'application. Entrez les données d'identification de l'utilisateur que vous avez associé au rôle webUser. Par défaut, ce rôle utilisateur est associé à tous les utilisateurs authentifiés. Tapez `admin1` comme ID utilisateur et `admin1` comme mot de passe. Une page pour afficher, ajouter, mettre à jour et supprimer des employés apparaît.
3. Affichez les employés. Cliquez sur **Ajouter un employé**. Entrez `emp1@acme.com` comme adresse électronique et cliquez sur **Soumettre**. Un message indique que l'utilisateur est introuvable.
4. Ajoutez un employé. Cliquez sur **Ajouter un employé**. Entrez `emp1@acme.com` comme adresse électronique, `Joe` comme prénom et `Doe` comme nom. Cliquez sur **Soumettre**. Un message s'affiche pour indiquer qu'un employé avec l'adresse `emp1@acme.com` a été ajouté.
5. Affichez le nouvel employé. Cliquez sur **Afficher un employé**. Entrez `emp1@acme.com` comme adresse électronique avec des zones vides pour les nom et prénom, et cliquez sur **Soumettre**. Un message s'affiche pour indiquer que l'employé a été trouvé et les noms corrects figurent dans les zones du prénom et du nom.
6. Supprimez l'employé. Cliquez sur **Supprimer un employé**. Entrez `emp1@acme.com` et cliquez sur **Soumettre**. Un message s'affiche pour indiquer que l'employé a été supprimé.

Point de contrôle de la leçon :

Vous avez installé et exécuté l'exemple d'application. Comme ce tutoriel utilise l'intégration WebSphere Application Server, vous ne pouvez pas voir le scénario lorsqu'un client ne parvient pas à s'authentifier sur le serveur eXtreme Scale. Si l'utilisateur s'authentifie auprès d' WebSphere Application Server correctement, eXtreme Scale est également correctement authentifié.

Module 3 : Configuration de la sécurité du transport

Configuration de la sécurité du transport pour protéger le transfert des données entre les clients et les serveurs dans la configuration.

Dans le module précédent dans le tutoriel, vous avez activé l'authentification WebSphere eXtreme Scale. Avec l'authentification, une application qui tente de se connecter au serveur WebSphere eXtreme Scale doit fournir des données d'identification. Par conséquent, un client non authentifié peut se connecter au serveur WebSphere eXtreme Scale. Les clients doivent être une application authentifiée qui s'exécute dans une cellule WebSphere Application Server.

Avec la configuration jusqu'à ce module, le transfert de données entre les clients dans le cluster appCluster et les serveurs du cluster xsCluster n'est pas chiffré. Cette configuration peut être acceptable si vos clusters WebSphere Application Server sont installés sur les serveurs derrière un pare-feu. Toutefois, dans certains scénarios, le trafic non chiffré n'est pas accepté pour certaines raisons, même si la topologie est protégée par un pare-feu. Par exemple, une politique gouvernementale pourrait imposer de chiffrer le trafic. WebSphere eXtreme Scale prend en charge Transport Layer Security/Secure Sockets Layer (TLS/SSL) pour sécuriser la communication entre les noeud finaux ObjectGrid, qui incluent des serveurs client, des serveurs de conteneur et des serveurs de catalogue.

Dans cet exemple de déploiement, les clients et les servers de conteneur eXtreme Scale s'exécutent tous dans l'environnement WebSphere Application Server. Les propriétés client ou serveur ne sont pas nécessaires pour configurer les paramètres SSL, car la sécurité du transport eXtreme Scale est gérée par les paramètres de transport CSIV2 (Application Server Common Secure Interoperability Protocol Version 2). Les serveurs WebSphere eXtreme Scale utilisent la même instance ORB (Object Request Broker) que les serveurs d'applications où ils sont exécutés. Définissez tous les paramètres SSL des serveurs client et de conteneur dans la configuration WebSphere Application Server en utilisant ces paramètres de transport CSIv2. Le serveur de catalogue dispose de ses propres chemins de transport propriétaires qui n'utilisent pas IIOP (Internet Inter-ORB Protocol) ni RMI (Remote Method Invocation). En raison de ces chemins de transport propriétaires, le serveur de catalogue ne peuvent pas être gérés par les paramètres de transport WebSphere Application Server CSIV2. Vous devez donc définir les propriétés SSL dans le fichier des propriétés du serveur de catalogue.

Objectifs d'apprentissage

A la fin des leçons de ce module, vous saurez :

- configurer le transport entrant et sortant CSIv2 ;
- ajouter des propriétés SSL dans le fichier de propriétés du serveur de catalogue ;
- vérifier le fichier des propriétés ORB ;
- exécuter l'exemple.

Durée

Ce module prend 60 minutes environ.

Prérequis

Cette étape du tutoriel repose sur les modules précédents. Etudiez les modules précédents du présent tutoriel avant de configurer la sécurité du transport.

Leçon 3.1 : Configuration du transport entrant et sortant CSIv2

Pour configurer Transport Layer Security/Secure Sockets Layer (TLS/SSL) pour le transport du serveur, affectez aux propriétés de transport entrant et sortant Common Secure Interoperability Protocol Version 2 (CSIv2) la valeur `SSL requis`

pour tous les serveurs WebSphere Application Server qui hébergent des clients, des serveurs de catalogue et des serveurs de conteneur.

Dans l'exemple de topologie du tutoriel, vous devez définir ces propriétés pour les serveurs d'applications s1, s2, xs1 et xs2. Procédez comme suit pout définir les transports entrant et sortant de tous les serveurs de la configuration.

Définissez les transports entrant et sortant dans la console d'administration. Vérifiez que la sécurité administrative est activée.

- **WebSphere Application Server Version 6.1** : Cliquez sur **Sécurité** > **Sécuriser l'administration** > **Application** > **Sécurité RMI/IIOP** et remplacez le type de transport par **SSL requis**.
- **WebSphere Application Server Version 7.0** : Cliquez sur **Sécurité** > **Sécurité globale** > **Sécurité RMI/IIOP** > **Communications entrants CSIv2**. Dans la couche de transport CSIv2 remplacez le type de transport par **SSL requis**. Répétez cette étape pour configurer les communications sortantes CSIv2.

Vous pouvez utiliser les paramètres de sécurité de noeud final géré de manière centralisée, ou configurer des référentiels SSL. Voir Paramètres de transport entrant CSIV2 (Common Secure Interoperability Version 2) pour plus d'informations.

Leçon 3.2 : Ajout de propriétés SSL au fichier des propriétés du serveur de catalogue

Le serveur de catalogue dispose de ses propres chemins de transport propriétaires qui ne peuvent pas être gérés par les paramètres de transport WebSphere Application Server Common Secure Interoperability Protocol Version 2 (CSIV2). Vous devez donc définir les propriétés SSL (Secure Sockets Layer) dans le fichier des propriétés du serveur de catalogue.

Pour configurer la sécurité du serveur de catalogue, des étapes supplémentaires sont nécessaires, car le serveur de catalogue dispose de ses propres chemins de transport propriétaires. Ces chemins de transport ne peut pas être gérés par les paramètres de transport du serveur d'applications CSIV2.

1. Editez les propriétés SSL dans le fichier `catServer2.props`. Pour configurer la sécurité du serveur de catalogue, supprimez les propriétés SSL suivantes dans le fichier de propriétés du serveur de catalogue. Pour ce tutoriel, les propriétés du serveur de catalogue se trouvent dans le fichier `catServer2.props`. Mettez à jour les propriétés keyStore et trustStore pour faire référence à l'emplacement approprié dans votre environnement.

   ```
   #alias=default
   #contextProvider=IBMJSSE2
   #protocol=SSL
   #keyStoreType=PKCS12
   #keyStore=/<WAS_HOME>/IBM/WebSphere/AppServer/profiles/<DMGR_NAME>/config/
   cells/<CELL_NAME>/nodes/<NODE_NAME>/key.p12
   #keyStorePassword=WebAS
   #trustStoreType=PKCS12
   #trustStore=/<WAS_HOME>/IBM/WebSphere/AppServer/profiles/<DMGR_NAME>/config/
   cells/<CELL_NAME>/nodes/<NODE_NAME>/trust.p12
   #trustStorePassword=WebAS
   #clientAuthentication=false
   ```

   Le fichier `catServer2.props` utilise le fichier de clés et le fichier de clés certifiées au niveau du noeud WebSphere Application Server par défaut. Si vous déployez un environnement de déploiement plus complexe, vous devez choisir le fichier de clés et le fichier de clés certifiées corrects. Dans certains cas, vous devez créer un fichier de clés et un fichier de clés certifiées et importez les clés depuis les fichiers de clés des autres serveurs. Notez que la chaîne `WebAS`

est le mot de passe par défaut des fichiers de clés et de clés certifiées WebSphere Application Server. Voir Configuration des certificats autosignés pour plus d'informations.

2. Dans le fichier `catServer2.props`, mettez à jour la valeur de la propriété transportType. Pour les étapes précédentes du tutoriel, la valeur est `TCP/IP`. Remplacez la valeur `SSL requis`.
3. Redémarrez le gestionnaire de déploiement pour activer les modifications apportées aux paramètres de sécurité du serveur de catalogue.

Point de contrôle de la leçon :

Vous avez défini les propriétés SSL pour le serveur de catalogue.

Leçon 3.3 : Exécution de l'exemple

Redémarrez tous les serveurs et exécutez de nouveau le modèle d'application. Vous devriez être en mesure d'exécuter les étapes sans aucun problème.

Voir «Leçon 2.4 : Installation et exécution de l'exemple», à la page 97 pour plus d'informations sur l'exécution et l'installation de l'exemple d'application.

Point de contrôle de la leçon :

Vous avez exécuté l'exemple d'application avec la sécurité du transport activée.

Module 4 : Utilisation de l'autorisation JAAS (Java Authentication and Authorization Service) dans WebSphere Application Server

Maintenant que vous avez configuré l'authentification pour les clients, vous pouvez configurer les autorisations de manière plus précise pour accorder aux utilisateurs des autorisations différentes. Par exemple, un "opérateur" peut être autorisé uniquement à afficher les données, alors qu'un "gestionnaire" peut exécuter toutes les opérations.

Après avoir authentifié un client, comme dans le module précédent dans ce tutoriel, vous pouvez attribuer des privilèges de sécurité par le biais des mécanismes d'autorisation eXtreme Scale. Le module précédent de ce tutoriel vous a montré comment activer l'authentification pour une grille de données à l'aide de l'intégration à WebSphere Application Server. Par conséquent, aucun client non authentifié ne peut se connecter aux serveurs eXtreme Scale ni envoyer des demandes au système. Toutefois, tous les clients authentifiés possèdent les mêmes permissions ou privilèges liés au serveur, tels que la lecture, l'écriture ou la suppression des données stockées dans les mappes ObjectGrid. Les clients peuvent également soumettre tout type de requête.

Cette partie du tutoriel explique comment utiliser l'autorisation eXtreme Scale pour attribuer différents privilèges aux utilisateurs authentifiés. WebSphere eXtreme Scale utilise un mécanisme d'autorisation basé sur l'autorisation. Vous pouvez affecter des catégories d'autorisations différentes qui sont représentées par des classes d'autorisation différentes. Ce module utilise la classe MapPermission. Pour la liste de toutes les propriétés possibles, voir Programmation d'autorisations client.

Dans WebSphere eXtreme Scale, la classe `com.ibm.websphere.objectgrid.security.MapPermission` représente les autorisations d'accès aux ressources eXtreme Scale , notamment les méthodes des

interfaces ObjectMap ou JavaMap. WebSphere eXtreme Scale définit les chaînes de permission suivantes pour accéder aux méthodes des interfaces ObjectMap et JavaMap :

- **read** : accorde l'autorisation de lire les données de la mappe.
- **write** : accorde l'autorisation de lire les données de la mappe.
- **insert** : accorde l'autorisation d'insérer les données dans la mappe.
- **remove**: accorde l'autorisation de supprimer les données de la mappe.
- **invalidate** : accorde l'autorisation d'invalider les données dans la mappe.
- **all** : accorde toutes les autorisations ci-dessus.

L'autorisation se produit lorsqu'un client eXtreme Scale utilise une API d'accès aux données, telles que ObjectMap ,JavaMap, ou les API EntityManager. L'environnement d'exécution eXtreme Scale vérifie les autorisations d'exécution de la mappe correspondante lorsque la méthode est appelée. Si les autorisations d'accès requises ne sont pas accordées au client, une exception AccessControlException est générée. Ce module explique comment utiliser l'autorisation JAAS (Java Authentication and Authorization Service) pour accorder des autorisations d'accès à la mappe pour différents utilisateurs.

Objectifs d'apprentissage

A la fin des leçons de ce module, vous saurez :

- activer l'autorisation WebSphere eXtreme Scale ;
- activer les autorisations utilisateur ;
- configure les autorisations de groupe.

Durée

Ce module prend 60 minutes environ.

Prérequis

Vous devez effectuer les modules précédents de ce tutoriel avant de configurer l'authentification.

Leçon 4.1 : Activation de l'autorisation WebSphere eXtreme Scale

Pour activer l'autorisation dans WebSphere eXtreme Scale, vous devez activer la sécurité sur un ObjectGrid spécifique.

Pour activer l'autorisation sur l'ObjectGrid, vous devez affecter à l'attribut **securityEnabled** la valeur true pour cet ObjectGrid spécifique dans le fichier XML. Pour ce tutoriel, vous pouvez utiliser le fichier `XSDeployment_sec.ear` dans le répertoire *samples_home*`/WASSecurity` dont la sécurité est déjà définie dans le fichier `objectGrid.xml` ou vous pouvez modifier le fichier existant `objectGrid.xml` pour activer la sécurité. Cette leçon explique comment modifier le fichier pour activer la sécurité.

1. Extrayez les fichiers dans le fichier `XSDeployment.ear`, puis décompressez le fichier `XSDeploymentWeb.war`.
2. Ouvrez le fichier `objectGrid.xml` et affectez à l'attribut la valeur `true` sur le niveau ObjectGrid. Voir un exemple de cet attribut ci-dessous :

```
<?xml version="1.0" encoding="UTF-8"?>

<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">

       <objectGrids>
           <objectGrid name="Grid" securityEnabled="true">
               <backingMap name="Map1" />
           </objectGrid>
       </objectGrids>

   </objectGridConfig>
```

Si vous avez plusieurs ObjectGrids définis, vous devez définir cet attribut dans chaque grille.

3. Remodularisez les fichiers `XSDeploymentWeb.war` et `XSDeployment.ear` pour inclure vos modifications. Nommez le fichier `XSDeployment_sec.ear` pour ne pas remplacer le package d'origine.
4. Désinstallez l'application existante `XSDeployment` et installez le fichier `XSDeployment_sec.ear`. Voir «Leçon 2.4 : Installation et exécution de l'exemple», à la page 97 pour plus d'informations sur le déploiement des applications.

Point de contrôle de la leçon :

Vous avez activé la sécurité sur l'ObjectGrid, ce qui permet également d'activer l'autorisation dans la grille de données.

Leçon 4.2 : Activation des autorisations basées sur l'utilisateur

Dans le module d'authentification de ce tutoriel, vous avez créé deux utilisateurs : `operator1` et `admin1`. Vous pouvez affecter des droits différents à ces utilisateurs avec l'autorisation JASS (Java Authentication and Authorization Service).

Définition de la règle d'autorisation JAAS (Java Authentication and Authorization Service) en utilisant des principaux d'utilisateur :

Vous pouvez affecter des autorisations aux utilisateurs que vous avez créés. Affectez les autorisations de lecture `operator1` uniquement à toutes les mappes. Affectez à l'utilisateur `admin1` toutes les autorisations. Utilisez le fichier de règle d'autorisation JAAS pour accorder des autorisations aux principaux.

Editez le fichier d'autorisation JAAS. Le fichier `xsAuth2.policy` se trouve dans le répertoire *samples_home*`/security` :

```
grant codebase http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction
Principal com.ibm.ws.security.common.auth.WSPrincipalImpl "defaultWIMFileBasedRealm/operator1" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "Grid.Map1", "read";
};

grant codebase http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction
Principal com.ibm.ws.security.common.auth.WSPrincipalImpl "defaultWIMFileBasedRealm/admin1" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "Grid.Map1", "all";
};
```

Dans ce fichier, le codebase `http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction` est une URL réservée à ObjectGrid. Toutes les autorisations ObjectGrid accordées aux principaux doivent utiliser ce codebase spécial. Les autorisations suivantes sont affectées dans ce fichier :

- Le première instruction accorde l'autorisation de mappe `read` au principal `operator1`. L'utilisateur `operator1` dispose uniquement de l'autorisation de lecture sur la mappe Map1 dans l'instance Grid ObjectGrid.

- La deuxième instruction accorde toutes les autorisations de mappe au principal admin1. L'utilisateur `admin1` dispose de toutes les autorisations sur la mappe Map1 dans l'instance Grid ObjectGrid.
- Le principal s'appelle `defaultWIMFileBasedRealm/operator1` et non pas `Operator1`. WebSphere Application Server ajoute automatiquement le nom de domaine au nom du principal lorsque des référentiels fédérés sont utilisés comme registre de comptes utilisateur. Modifiez cette valeur si nécessaire.

Définition du fichier de règle d'autorisation JAAS à l'aide des propriétés JVM :

Procédez comme suit pour définir les propriétés JVM des serveurs xs1 et xs2qui qui se trouvent dans le cluster xsCluster. Si vous utilisez une topologie qui est différente de l'exemple de topologie qui est utilisé dans ce tutoriel, définissez le fichier sur tous vos serveurs de conteneur.

1. Dans la console d'administration, cliquez sur **Serveurs** > **Serveurs d'applications** > *server_name* > **Java et gestion des processus** > **Définition de processus** > **Java Virtual Machine.**
2. Ajoutez les arguments JVM génériques suivants :

   ```
   -Djava.security.auth.policy=samples_home/security/xsAuth2.policy
   ```
3. Cliquez sur **OK** et enregistrez les modifications.

Exécution de l'exemple d'application pour tester les autorisations :

Vous pouvez utiliser l'exemple d'application pour tester les paramètres d'autorisation. L'administrateur continue d'avoir tous les autorisations dans dans la mappe Map1, y compris l'affichage et l'ajout d'employés. L'opérateur doit pouvoir afficher uniquement les employés, car seule l'autorisation Lecture lui a été affectée.

1. Redémarrez tous les serveurs d'applications qui exécutent des serveurs conteneurs.
2. Ouvrez l'application `EmployeeManagementWeb`. Dans un navigateur Web, ouvrez `http://<host>:<port>/EmployeeManagermentWeb/management.jsp`.
3. Connectez-vous à l'application en tant qu'administrateur. Utilisez le nom d'utilisateur `admin1` et le mot de passe `admin1`.
4. Essayez d'afficher un employé. Cliquez sur **Afficher un employé** et recherchez l'adresse électronique `authemp1@acme.com`. Un message indique que l'utilisateur est introuvable.
5. Ajoutez un employé. Cliquez sur **Ajouter un employé**. Ajoutez l'adresse électronique `authemp1@acme.com`, le prénom `Joe` et le nom `Doe`. Cliquez sur **Soumettre**. Un message indique que l'employé a été ajouté.
6. Connectez-vous en tant qu'opérateur. Ouvrez une seconde fenêtre de navigateur Web et `http://<host>:<port>/EmployeeManagermentWeb/management.jsp`. Utilisez le nom d'utilisateur `operator1` et le mot de passe `operator1`.
7. Essayez d'afficher un employé. Cliquez sur **Afficher un employé** et recherchez l'adresse électronique `authemp1@acme.com`. L'employé est affiché.
8. Ajoutez un employé. Cliquez sur **Ajouter un employé**. Ajoutez l'adresse électronique `authemp2@acme.com`, le prénom `Joe` et le nom `Doe`. Cliquez sur **Soumettre**. Le message suivant s'affiche :

   ```
   An exception occurs when Add the employee. See below for detailed exception messages.
   ```

   L'exception suivante se trouve dans la chaîne d'exception :

   ```
   java.security.AccessControlException: Access denied
   (com.ibm.websphere.objectgrid.security.MapPermission Grid.Map1 insert)
   ```

Ce message s'affiche, car l'utilisateur operator1 n'est pas autorisé à insérer des données dans la mappe Map1.

Si vous utilisez une version WebSphere Application Server antérieure à la version 7.0.0.11, une erreur java.lang.StackOverflowError peut s'afficher sur le serveur de conteneur. Elle est provoquée par l'IBM Developer Kit. Le problème est résolu dans l'IBM Developer Kit fourni avec WebSphere Application Server Version 7.0.0.11 et les versions suivantes.

Point de contrôle de la leçon :

Dans cette leçon, vous avez configuré l'autorisation en attribuant des autorisations à des utilisateurs spécifiques.

Leçon 4.4 : Configuration des autorisations de groupe

Dans la leçon précédente, vous avez affecté des autorisations utilisateur individuelles aux principaux utilisateur dans la règle d'autorisation JAAS (Java Authentication and Authorization CService). Cependant, lorsque vous avez des centaines ou des milliers d'utilisateurs, utilisez l'autorisation de groupe, qui autorise l'accès en fonction des groupes plutôt que des utilisateurs individuels.

Malheureusement, l'objet Subject qui est authentifié depuis de WebSphere Application Server contient uniquement un principal utilisateur. Cet objet ne contient pas de principal. Vous pouvez ajouter un module de connexion personnalisé pour alimenter le principal de groupe dans l'objet Subject.

Pour ce tutoriel, le module de connexion personnalisé s'appelle com.ibm.websphere.samples.objectgrid.security.lm.WASAddGroupLoginModule. Le module se trouve dans le fichier groupLM.jar. Placez le fichier JAR dans le répertoire WAS-INSTALL/lib/ext.

WASAddGroupLoginModule extrait les données d'identification de groupe public depuis le sujet WebSphere Application Server et crée un principal Groupe, com.ibm.websphere.samples.objectgrid.security.WSGroupPrincipal, pour représenter le groupe. Ce principal de groupe peut ensuite être utilisé pour l'autorisation de groupe. Les groupes sont définis dans le fichier xsAuthGroup2.policy :

```
grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
principal com.ibm.websphere.sample.xs.security.WSGroupPrincipal
  "defaultWIMFileBasedRealm/cn=operatorGroup,o=defaultWIMFileBasedRealm" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "Grid.Map1", "read";
};

grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
principal com.ibm.websphere.sample.xs.security.WSGroupPrincipal
 "defaultWIMFileBasedRealm/cn=adminGroup,o=defaultWIMFileBasedRealm" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "Grid.Map1", "all";
};
```

Le nom du principal est WSGroupPrincipal qui représente le groupe.

Ajout du module de connexion personnalisé :

Le module de connexion personnalisé doit être ajouté à chacune des entrées de module de connexion système suivantes : si vous utilisez l'authentification LTPA (Lightweight Third Party Authentication), ajoutez l'entrée aux modules de connexion RMI_INBOUND. LTPA est le mécanisme d'authentification par défaut pour WebSphere Application Server Version 7.0. Pour une configuration WebSphere

Application Server Network Deployment, il suffit de configurer les entrées de configuration du mécanisme d'authentification LTPA.

Procédez comme suit pour configurer le module de connexion com.ibm.websphere.samples.objectgrid.security.lm.WASAddGroupLoginModule :

1. Dans la console d'administration, cliquez sur **Sécurité** > **Sécurité globale** > **Java Authentication and Authorization Service** > **Connexions système** > ***login_module_name*** > **Modules de connexion JAAS** > **Nouveau**.
2. Entrez le nom de classe `com.ibm.websphere.sample.xs.security.lm.WASAddGroupLoginModule`.
3. Facultatif : Ajoutez une propriété `debug` et définissez la valeur `true`.
4. Cliquez sur **Appliquer** pour ajouter le nouveau module à la liste des modules de connexion.

Définition du fichier de règle d'autorisation JAAS à l'aide des propriétés JVM :

Dans la console d'administration, effectuez les étapes suivantes pour les serveurs xs1 et xs2 dans xsCluster. Si une topologie de déploiement différente est utilisée, effectuez les étapes suivantes pour les serveurs d'applications qui hébergent les serveurs de conteneur.

1. Dans la console d'administration, cliquez sur **Serveurs** > **Serveurs d'applications** > ***server_name*** > **Java et gestion des processus** > **Définition de processus** > **Java virtual machine**
2. Entrez les arguments JVM génériques suivants ou remplacer l'entrée `-Djava.security.auth.policy` par le texte suivant :

   `-Djava.security.auth.policy=`*`samples_home`*`/security/xsAuthGroup2.policy`
3. Cliquez sur **OK** et enregistrez les modifications.

Test d'autorisation de groupe avec l'exemple d'application :

Vous pouvez vérifier que l'autorisation de groupe est configurée par le module de connexion avec l'exemple d'application.

1. Redémarrez les serveurs de conteneur. Pour ce tutoriel, les serveurs de conteneur sont les serveurs xs1 et xs2.
2. Connectez-vous à l'exemple d'application. Dans un navigateur Web, ouvrez `http://<host>:<port>/EmployeeManagementWeb/management.jsp` et connectez-vous avec le nom d'utilisateur `admin1` et le mot de passe `admin1`.
3. Affichez un employé. Cliquez sur **Afficher un employé** et recherchez l'adresse électronique `authemp2@acme.com`. Un message indique que l'utilisateur est introuvable.
4. Ajoutez un employé. Cliquez sur **Ajouter un employé**. Ajoutez l'adresse électronique `authemp2@acme.com`, le prénom `Joe` et le nom `Doe`. Cliquez sur **Soumettre**. Un message indique que l'employé a été ajouté.
5. Connectez-vous en tant qu'opérateur. Ouvrez une seconde fenêtre de navigateur Web et l'URL `http://<host>:<port>/EmployeeManagermentWeb/management.jsp`. Utilisez le nom d'utilisateur `operator1` et le mot de passe `operator1`.
6. Essayez d'afficher un employé. Cliquez sur **Afficher un employé** et recherchez l'adresse électronique `authemp2@acme.com`. L'employé est affiché.
7. Ajoutez un employé. Cliquez sur **Ajouter un employé**. Ajoutez l'adresse électronique `authemp3@acme.com`, le prénom `Joe` et le nom `Doe`. Cliquez sur **Soumettre**. Le message suivant s'affiche :

```
An exception occurs when Add the employee. See below for detailed exception messages.
```

L'exception suivante se trouve dans la chaîne d'exception :

```
java.security.AccessControlException: Access denied
(com.ibm.websphere.objectgrid.security.MapPermission Grid.Map1 insert)
```

Ce message s'affiche, car l'utilisateur operator n'est pas autorisé à insérer des données dans la mappe Map1.

Point de contrôle de la leçon :

Vous avez configuré des groupes pour simplifier l'attribution des droits aux utilisateurs de votre application.

Module 5 : Utilisation de l'utilitaire `xscmd` pour surveiller les grilles de données et les mappes

Vous pouvez utiliser l'utilitaire **xscmd** pour afficher les grilles de données principales et les tailles de mappe de la grille de données `Grid`. L'outil **xscmd** utilise le bean géré pour interroger tous les artefacts de grille de données, telles que les fragments primaires, les fragments de réplique, les serveurs de conteneur, les tailles de mappe.

Dans ce tutoriel, les serveurs de conteneur et de catalogue s'exécutent sur des serveurs d'applications WebSphere Application Server. L'environnement d'exécution WebSphere eXtreme Scale enregistre les beans gérés (MBean) avec le serveur de bean créé par l'environnement d'exécution WebSphere Application Server. La sécurité qui est utilisée par l'outil **xscmd** est fournie par la sécurité WebSphere Application Server MBean. Par conséquent, une configuration de sécurité WebSphere eXtreme Scale n'est pas nécessaire.

1. A l'aide d'un outil de ligne de commande, ouvrez le répertoire *DMGR_PROFILE*`/bin`.
2. Exécutez l'outil **xscmd**.

   Utilisez la commande **-c listObjectGridPlacement -sf P** pour lister le placement des fragments primaires. Linux UNIX

   ```
   xscmd.sh -g Grid -ms mapSet -c showPlacement -sf P
   ```

   Windows

   ```
   xscmd.bat -g Grid -ms mapSet -c showPlacement -sf P
   ```

   Pour pouvoir afficher la sortie, le système vous demande de vous connecter avec votre ID et votre mot de passe WebSphere Application Server.

Point de contrôle de la leçon

Vous avez utilisé l'outil **xscmd** dans WebSphere Application Server.

Tutoriel : Intégration de la sécurité WebSphere eXtreme Scale dans un environnement mixte avec un authentificateur externe

Ce tutoriel explique comment sécuriser les serveurs WebSphere eXtreme Scale partiellement déployés dans un environnement WebSphere Application Server.

Dans le déploiement de ce tutoriel, les serveurs de conteneur sont déployés dans WebSphere Application Server. Le serveur de catalogue est déployé en tant que serveur autonome lancé dans un environnement Java SE (Java Standard Edition).

Comme le serveur de catalogue n'est pas déployé dans WebSphere Application Server, vous ne pouvez pas utiliser les plug-ins WebSphere Application Server Authentication. Pour plus d'informations sur le processus de configuration des plug-ins WebSphere Application Server Authentication, voir «Tutoriel : Intégration de la sécurité WebSphere eXtreme Scale à WebSphere Application Server», à la page 86. Dans ce tutoriel, un authentificateur différent est requis pour l'authentification du serveur de catalogue. Vous pouvez configurer un authentificateur de fichier de clés pour authentifier les clients.

Objectifs d'apprentissage

Les objectifs d'apprentissage de ce tutoriel sont les suivants :

- Configuration de WebSphere eXtreme Scale pour utiliser le plug-in KeyStoreLoginAuthenticator
- Configuration de la sécurité du transport WebSphere eXtreme Scale pour utiliser la configuration WebSphere Application Server CSIv2 et le fichier de propriétés WebSphere eXtreme Scale
- Utilisation de l'autorisation JAAS (Java Authentication and Authorization Service) dans WebSphere Application Server
- Utilisation de l'utilitaire **xscmd** pour surveiller les grilles de données et les mappes que vous avez créés dans le tutoriel.

Durée

Ce tutoriel dure environ 4 heures.

Introduction : Sécurité dans un environnement mixte

Dans ce tutoriel, vous intégrez la sécurité WebSphere eXtreme Scale dans un environnement mixte. Les serveurs de conteneur s'exécutent dans WebSphere Application Server et le service de catalogue s'exécute en mode autonome. Etant donné que le serveur de catalogue est en mode autonome, vous devez configurer un authentificateur externe.

Important : Si le serveur de conteneur et le serveur de catalogue fonctionnent dans WebSphere Application Server, vous pouvez utiliser des plug-ins WebSphere Application Server Authentication ou un authentificateur externe. Pour plus d'informations sur l'utilisation des plug-ins WebSphere Application Server Authentication, voir «Tutoriel : Intégration de la sécurité WebSphere eXtreme Scale à WebSphere Application Server», à la page 86.

Objectifs d'apprentissage

Les objectifs d'apprentissage de ce tutoriel sont les suivants :

- Configuration de WebSphere eXtreme Scale pour utiliser le plug-in KeyStoreLoginAuthenticator
- Configuration de la sécurité du transport WebSphere eXtreme Scale pour utiliser la configuration WebSphere Application Server CSIv2 et le fichier de propriétés WebSphere eXtreme Scale
- Utilisation de l'autorisation JAAS (Java Authentication and Authorization Service) dans WebSphere Application Server
- Utilisation de l'utilitaire **xscmd** pour surveiller les grilles de données et les mappes que vous avez créés dans le tutoriel.

Durée

Ce tutoriel prend 4 heures environ.

Niveau de compétence

Intermédiaire

Audience

Les développeurs et les administrateurs qui veulent intégrer la sécurité entre WebSphere eXtreme Scale et WebSphere Application Server et configurer des authentificateurs externes.

Configuration requise

- WebSphere Application Server Version 6.1, Version 7.0.0.11 et versions suivantes avec les correctifs suivants appliqués : correctif temporaire PM20613 et correctif temporaire PM15818.
- Le serveur de catalogue doit être en cours d'exécution sur une installation autonome et non pas une installation qui est intégrée à WebSphere Application Server.
- Mettez à jour l'environnement d'exécution Java pour appliquer le correctif IZ79819: IBMJDK FAILS TO READ PRINCIPAL STATEMENT WITH WHITESPACE FROM SECURITY FILE
- Le noeud autonome qui exécute le service de catalogue doit utiliser IBM Software Development Kit Version 1.6 J9. Ce kit de développement de logiciels est inclus dans l'installation de WebSphere Application Server. Le noeud de serveur de catalogue doit être une installation autonome, car vous ne pouvez pas exécuter la commande **startOgServer** dans une installation de WebSphere eXtreme Scale sur WebSphere Application Server.

Ce tutoriel utilise dans l'exemple quatre serveurs d'applications WebSphere Application Server et un gestionnaire de déploiement.

Prérequis

Une connaissance de base des éléments suivants est utile avant de démarrer ce tutoriel :

- Modèle de programmation WebSphere eXtreme Scale
- Concepts de sécurité de base WebSphere eXtreme Scale
- Concepts de sécurité de base WebSphere Application Server

Pour plus d'informations sur l'intégration de la sécurité WebSphere eXtreme Scale et WebSphere Application Server, voir «Intégration de la sécurité dans WebSphere Application Server», à la page 525.

Module 1 : Préparation de l'environnement WebSphere Application Server et autonome

Avant de commencer le tutoriel, vous devez créer une topologie de base qui inclut des serveurs de conteneur qui s'exécutent dans WebSphere Application Server. Dans ce tutoriel, les serveurs de catalogue s'exécutent en mode autonome.

Objectifs d'apprentissage

A la fin des leçons de ce module, vous :

- comprendrez la topologie mixte et les fichiers qui sont nécessaires pour le tutoriel ;
- saurez configurer WebSphere Application Server pour exécuter les serveurs de conteneur.

Durée

Ce module prend 60 minutes environ.

Leçon 1.1 : Compréhension de la topologie et obtention des fichiers du tutoriel

Pour préparer votre environnement pour le tutoriel, vous devez configurer les serveurs de catalogue et de conteneur de la topologie.

Cette leçon vous guide dans l'exemple de topologie et les applications qui sont utilisés dans le tutoriel. Pour commencer à exécuter le tutoriel, vous devez télécharger les applications et placer les fichiers de configuration dans les emplacements propres à votre environnement. Vous pouvez télécharger l'exemple d'application depuis le wiki WebSphere eXtreme Scale .

Topologie : Dans ce tutoriel, vous créez les clusters suivants dans la cellule WebSphere Application Server :

- **Cluster appCluster** : héberge l'exemple d'application d'entreprise EmployeeManagement. Ce cluster contient les deux serveurs d'applications s1 et s2.
- **Clster xsCluster** : héberge les serveurs de conteneur eXtreme Scale. Ce cluster contient les deux serveurs d'applications xs1 et xs2.

Dans cette topologie de déploiement, les serveurs d'applications s1 et s2 sont les serveurs client qui accèdent aux données qui sont stockées dans la grille de données. Les serveurs xs1 et xs2 sont les serveurs de conteneurs qui hébergent la grille de données.

Autre configuration : vous pouvez héberger tous les serveurs d'applications dans un seul cluster, tel que appCluster. Avec cette configuration, tous les serveurs du cluster sont les clients et les serveurs de conteneur. Ce tutoriel utilise deux clusters pour distinguer les serveurs d'applications qui hébergent les clients et les serveurs conteneurs.

Dans ce tutoriel, vous configurez un domaine de services de catalogue qui se compose d'un serveur distant qui ne se trouve pas dans la cellule WebSphere Application Server. Cette configuration n'étant pas la valeur par défaut, les serveurs de catalogue sont exécutés dans le gestionnaire de déploiement et les autres processus le sont dans la cellule WebSphere Application Server. Voir «Création de domaines de services de catalogue dans WebSphere Application Server», à la page 258 pour plus d'informations sur la création d'un domaine de services de catalogue constitué de serveurs distants.

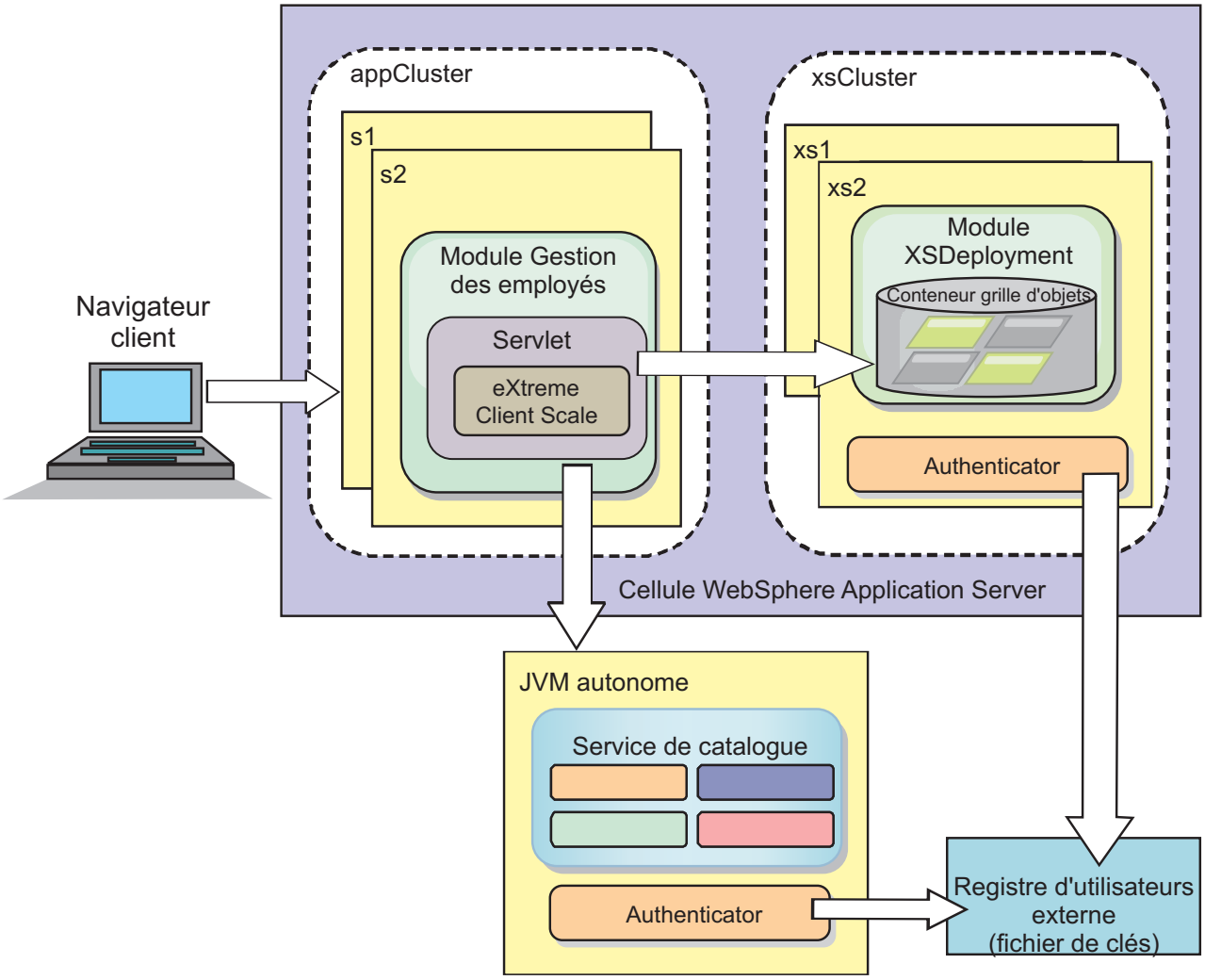


Figure 20. Topologie du tutoriel

Applications : Dans ce tutoriel, vous utilisez deux applications et un fichier de bibliothèque partagée :

- **`EmployeeManagement.ear`** : l'application `EmployeeManagement.ear` est une application d'entreprise simplifiée Java 2 Platform, Enterprise Edition (J2EE). Elle contient un module Web pour gérer les profils des employés. Le module Web contient le fichier `management.jsp` pour afficher, insérer, mettre à jour et supprimer les profils d'employés qui sont stockés dans les serveurs de conteneur.
- **`XSDeployment.ear`** : cette application contient un module d'application d'entreprise, sans artefacts d'application. Les objets cache sont regroupés dans le fichier `EmployeeData.jar`. Le fichier `EmployeeData.jar` est déployé comme bibliothèque partagée pour le fichier `XSDeployment.ear` pour que le fichier `XSDeployment.ear` puisse accéder aux classes. Le but de cette application consiste à modulariser le fichier de configuration eXtreme Scale et le fichier de propriétés. Lorsque cette application d'entreprise est démarrée, les fichiers de configuration eXtreme Scale sont automatiquement détectés par l'environnement d'exécution eXtreme Scale pour créer les serveurs de conteneur. Ces fichiers de configuration incluent les fichiers `objectGrid.xml` et `objectGridDeployment.xml`.
- **`EmployeeData.jar`** : ce fichier jar contient une classe, com.ibm.websphere.sample.xs.data.EmployeeData. Cette classe représente les

données de l'employé qui est stocké dans la grille. Ce fichier d'archive Java (JAR) est déployé avec les fichiers `EmployeeManagement.ear` et `XSDeployment.ear` comme bibliothèque partagée.

Obtention des fichiers du tutoriel :

1. Téléchargez les fichiers `WASSecurity.zip` et `security_extauth.zip` depuis le wiki WebSphere eXtreme Scale .
2. Extrayez le ficher `WASSecurity.zip` dans un répertoire pour afficher les données binaires et les artefacts source (par exemple un répertoire `wxs_samples/`). Ce répertoire est *samples_home* pour le reste du tutoriel. Consultez le fichier `README.txt` dans le package pour la description de son contenu et savoir comment charger le code source dans votre espace de travail Eclipse. Les fichiers de configuration ObjectGrid suivants se trouvent dans le répertoire `META-INF` :
   - `objectGrid.xml`
   - `objectGridDeployment.xml`
3. Créez un répertoire pour stocker les fichiers de propriétés qui sont utilisés pour sécuriser cet environnement. Par exemple, vous pouvez créer le répertoire `/opt/wxs/security`.
4. Extrayez le fichier `security_extauth.zip` vers *samples_home*. Le fichier `security_extauth.zip` contient les fichiers de configuration de sécurité suivants qui sont utilisés dans ce tutoriel : Fichiers de configuration :
   - `catServer3.props`
   - `server3.props`
   - `client3.props`
   - `security3.xml`
   - `xsAuth3.props`
   - `xsjaas3.config`
   - `sampleKS3.jks`

A propos des fichiers de configuration :

Les fichiers `objectGrid.xml` et `objectGridDeployment.xml` créent les grilles de données et les mappes qui stockent les données d'application.

Ces fichiers de configuration doivent s'appeler `objectGrid.xml` et `objectGridDeployment.xml`. Lorsque le serveur d'applications démarre, eXtreme Scale détecte ces fichiers dans le répertoire `META-INF` de l'EJB et des modules Web. Si ces fichiers sont trouvés, il suppose que la machine JVM (Java virtual machine) fait office de serveur de conteneur pour les grilles de données définie dans les fichiers de configuration.

Fichier `objectGrid.xml`

Le fichier `objectGrid.xml` défini un ObjectGrid nommé `Grid`. La grille de données `Grid` a une mappe, la mappe `Map1`, qui stocke le profil d'employé pour l'application.

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">

 <objectGrids>
        <objectGrid name="Grid" txTimeout="15">
            <backingMap name="Map1" />
```

```
        </objectGrid>
    </objectGrids>

</objectGridConfig>
```

Fichier objectGridDeployment.xml

Le fichier objectGridDeployment.xml indique comment déployer la grille de données Grid. Lorsque la grille est déployée, elle dispose de cinq partitions et d'une réplique synchrone.

```
<?xml version="1.0" encoding="UTF-8"?>

<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

    <objectgridDeployment objectgridName="Grid">
        <mapSet name="mapSet" numberOfPartitions="5" minSyncReplicas="0" maxSyncReplicas="1" >
            <map ref="Map1"/>
        </mapSet>
    </objectgridDeployment>

</deploymentPolicy>
```

Point de contrôle de la leçon :

Dans cette leçon, vous avez découvert la topologie du tutoriel et ajouté les fichiers de configuration et les exemples d'applications à votre environnement.

Leçon 1.2 : Configuration de l'environnement WebSphere Application Server

Pour préparer votre environnement pour le tutoriel, vous devez configurer la sécurité WebSphere Application Server. Activez la sécurité d'administration et d'application en utilisant des référentiels fédérés basés sur un fichier interne comme registre de comptes utilisateur. Ensuite, vous pouvez créer des clusters de serveurs pour héberger les serveurs d'applications client et les serveurs de conteneur. Vous devez créer et démarrer le serveurs de catalogue également.

Les étapes suivantes s'appliquent à WebSphere Application Server Version 7.0. Toutefois, vous pouvez appliquez les concepts aux versions antérieures de WebSphere Application Server.

Configuration de la sécurité WebSphere Application Server :

Créez et étendez des profils pour le gestionnaire de déploiement et les noeuds avec WebSphere eXtreme Scale. Pour plus d'informations, voir «Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client avec WebSphere Application Server», à la page 163.

Configurez la sécurité WebSphere Application Server.

1. Dans la console d'administration WebSphere Application Server, cliquez sur **Sécurité** > **Sécurité globale**.
2. Sélectionnez **Référentiels fédérés** comme **référentiel de comptes utilisateur**. Cliquez sur **Définir comme actif**.
3. Cliquez sur **Configurer** pour accéder au panneau des **référentiels fédérés**.
4. Entrez le **nom de l'administrateur principal**, tel que admin. Cliquez sur **Appliquer.**
5. Lorsque vous y êtes invité, entrez le mot de passe de l'administrateur et cliquez sur **OK**. Sauvegardez vos modifications.

6. Dans la page **Sécurité globale**, vérifiez que le paramètre **Référentiels fédérés** est affecté du registre de comptes utilisateur en cours.
7. Sélectionnez **Activer la sécurité administrative**, **Activer la sécurité de l'application** et **Utiliser la sécurité Java 2 pour limiter l'accès de l'application aux ressources locales**. Cliquez sur **Appliquer** et enregistrez les modifications.
8. Redémarrez le gestionnaire de déploiement et les serveurs d'applications actifs.

La sécurité administrative WebSphere Application Server est activée à l'aide des référentiels fédérés basés sur un fichier interne comme registre de comptes utilisateur.

Création de clusters de serveurs :

Créez deux clusters de serveurs dans votre configuration WebSphere Application Server : `appCluster` pour héberger l'exemple d'application du tutoriel et `xsCluster` pour héberger la grille de données.

1. Dans la console d'administration WebSphere Application Server, ouvrez le panneau des clusters. Cliquez sur **Serveurs** > **Clusters** > **Clusters de serveurs d'applications WebSphere** > **Nouveau**.
2. Entrez `appCluster` comme nom de cluster, ne renseignez pas l'option **Environnement local préféré** et cliquez sur **Suivant**.
3. Créez des serveurs dans le cluster. Créez le serveur `s1` en conservant les options par défaut. Ajoutez le membre `s2` au cluster.
4. Exécutez les étapes restantes dans l'assistant pour créer le cluster. Sauvegardez les modifications.
5. Répétez ces étapes pour créer le cluster `xsCluster`. Ce cluster contient les deux serveurs `xs1` et `xs2`.

Créer un domaine de services de catalogue :

Après avoir configuré le cluster de serveurs et la sécurité, vous devez indiquer où les serveurs de catalogue démarrent.

Définissez un domaine de services de catalogue dans WebSphere eXtreme Scale

1. Dans la console d'administration de WebSphere Application Server, cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **Domaines de services de catalogue**.
2. Créez le domaine de services de catalogue. Cliquez sur **Nouveau**. Créez le domaine de services de catalogue avec le nom `catalogService1` et activez le domaine de services de catalogue comme valeur par défaut.
3. Ajoutez les serveurs distants au domaine du service de catalogue. Sélectionnez **Serveur distant**. Indiquez le nom d'hôte sur lequel le serveur de catalogue est en cours d'exécution. Utilisez la valeur de port d'écoute de `16809` pour cet exemple.
4. Cliquez sur **OK** et enregistrez les modifications.

Point de contrôle de la leçon :

Vous avez activé la sécurité dans WebSphere Application Server et créé une topologie de serveur pour WebSphere eXtreme Scale.

Module 2 : Configuration de l'authentification WebSphere eXtreme Scale dans un environnement mixte

En configurant l'authentification, vous pouvez déterminer l'identité du demandeur. WebSphere eXtreme Scale prend en charge l'authentification client-serveur et serveur-serveur.

Flux d'authentification

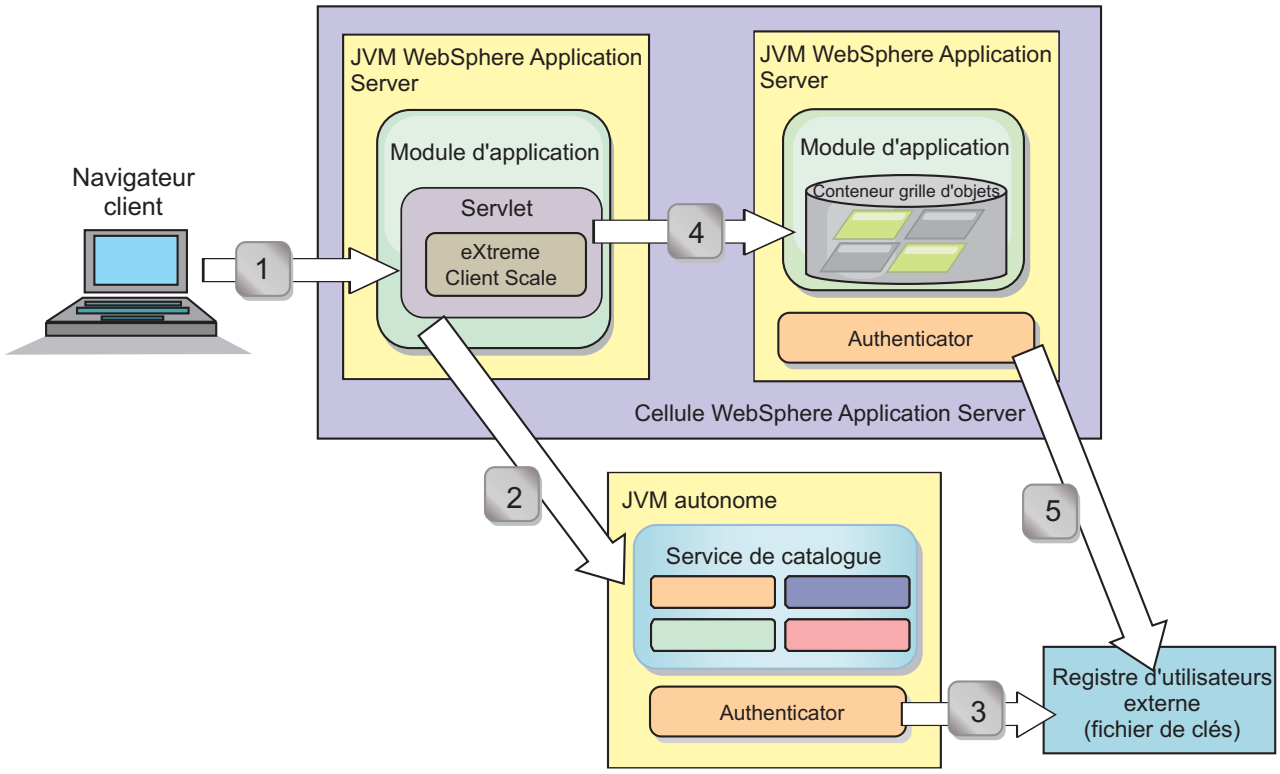


Figure 21. Flux d'authentification

Le diagramme suivant montre deux serveurs d'applications. Le premier serveur d'applications héberge l'application Web, qui est également un client WebSphere eXtreme Scale. Le second serveur d'applications héberge un serveur de conteneur. Le serveur de catalogue s'exécute dans une machine JVM (Java Virtual Machine) et non pas WebSphere Application Server.

Les flèches numérotées dans le diagramme indiquent le flux d'authentification :

1. Un utilisateur d'application d'entreprise accède au navigateur Web et se connecte au premier serveur d'applications avec un nom d'utilisateur et un mot de passe. Le premier serveur d'applications envoie le nom d'utilisateur et le mot de passe du client à l'infrastructure de sécurité pour s'authentifier auprès du registre des utilisateurs. Le registre d'utilisateurs est un fichier de clés. En conséquence, les informations de sécurité sont stockées sur l'unité d'exécution WebSphere Application Server.
2. Le fichier JSP (JavaServer Pages) fait office de client WebSphere eXtreme Scale pour extraire les informations de sécurité à partir du fichier des propriétés du client. L'application JSP qui fait office de client WebSphere eXtreme Scale envoie les données d'identification de sécurité du client WebSphere eXtreme Scale avec la demande au serveur de catalogue. L'envoi des données d'identification de sécurité avec la demande est un modèle *runAs*. Dans ce

modèle, le client du navigateur Web s'exécute en tant que client WebSphere eXtreme Scale pour accéder aux données stockées dans le serveur de conteneur. Le client utilise les données d'identification de client JVM (Java virtual machine) pour se connecter aux serveurs WebSphere eXtreme Scale. Le modèle runAs revient à se connecter à une base de données avec un ID utilisateur et un mot de passe au niveau de la source de données.

3. Le serveur de catalogue reçoit les données d'identification du client WebSphere eXtreme Scale, qui contiennent les jetons de sécurité WebSphere Application Server. Ensuite, le serveur de catalogue appelle le plug-in Authentificateur pour authentifier les données d'identification du client. L'authentificateur se connecte au registre d'utilisateurs externe et envoie les données d'identification du client vers le registre d'utilisateurs pour les authentifier.
4. Le client envoie l'ID utilisateur et le mot de passe au serveur de conteneur qui est hébergé sur le serveur d'applications.
5. Le service de conteneur, hébergé sur le serveur d'applications, reçoit les données d'identification du client WebSphere eXtreme Scale, à savoir la paire ID utilisateur-mot de passe. Ensuite, le serveur de catalogue appelle le plug-in Authentificator pour authentifier les données d'identification du client. L'authentificateur se connecte au registre d'utilisateurs du fichier de clés et envoie les données d'identification du client au registre d'utilisateurs pour les authentifier

Objectifs d'apprentissage

A la fin des leçons de ce module, vous saurez :

- configurer la sécurité du client WebSphere eXtreme Scale ;
- configurer la sécurité du serveur de catalogue WebSphere eXtreme Scale ;
- configurer la sécurité du serveur de conteneur WebSphere eXtreme Scale ;
- installer et exécuter l'exemple d'application.

Durée

Ce module prend 60 minutes environ.

Leçon 2.1 : Configuration de la sécurité du client WebSphere eXtreme Scale

Vous définissez les propriétés du client avec un fichier de propriétés. Ce fichier indique la classe d'implémentation CredentialGenerator à utiliser.

Contenu du fichier des propriétés du client :

Ce tutoriel utilise des jetons de sécurité WebSphere Application Server pour les données d'identification du client. Le répertoire *samples_home*/`security_extauth` contient le fichier `client3.props`.

Le fichier `client3.props` contient les paramètres suivants :

securityEnabled
Active la sécurité du client WebSphere eXtreme Scale. La valeur est `true` pour indiquer que le client doit envoyer les informations de sécurité disponibles au serveur.

credentialAuthentication
Spécifie la prise en charge de l'authentification des données d'identification

du client. La valeur est `Supported` pour indiquer que le client prend en charge l'authentification des données d'identification du client.

credentialGeneratorClass
Indique le nom de la classe qui implémente l'interface com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator. La valeur correspond à la classe com.ibm.websphere.objectgrid.security.plugins.builtins. UserPasswordCredentialGenerator pour que le client extrait les informations de sécurité de la classe UserPasswordCredentialGenerator.

credentialGeneratorProps
Indique le nom et le mot de passe : `manager manager1`. Le nom d'utilisateur est `manager` et le mot de passe est `manager1`. Vous pouvez également utiliser la commande **FilePasswordEncoder.bat|sh** pour coder cette propriété à l'aide d'un algorithme or exclusif (xor).

Définition du fichier des propriétés du client en utilisant des propriétés JVM (Java virtual machine) :

Dans la console d'administration, procédez comme suit pour les serveurs `s1` et `s2` dans le cluster `appCluster`. Si vous utilisez une topologie différente, procédez comme suit pour tous les serveurs d'applications sur lesquels l'application `EmployeeManagement` est déployée.

1. **Serveurs** > **Serveurs d'applications WebSphere** > *server_name* > **Java et gestion de processus** > **Définition de processus** > **Java Virtual Machine**.
2. Créez la propriété JVM générique suivante pour définir l'emplacement du fichier de propriétés du client :
   `-Dobjectgrid.client.props=`*samples_home*`/security_extauth/client3.props`
3. Cliquez sur **OK** et enregistrez les modifications.

Point de contrôle de la leçon :

Vous avez édité le fichier de propriétés du client et configuré les serveurs dans le cluster `appCluster` pour utiliser le fichier de propriétés du client. Le fichier des propriétés indique la classe d'implémentation CredentialGenerator à utiliser.

Leçon 2.2 : Configuration de la sécurité du serveur de catalogue

Un serveur de catalogue contient deux niveaux d'informations de sécurité : le premier niveau contient les propriétés de sécurité communes à tous les serveurs WebSphere eXtreme Scale, y compris les serveurs de service de catalogue et de conteneur. Le deuxième niveau contient les propriétés de sécurité qui sont spécifiques du serveur de catalogue.

Les propriétés de sécurité qui sont communes au serveur de catalogue et au serveur de conteneur sont configurées dans le fichier XML du descripteur de sécurité. La configuration de l'authentificateur, qui représente le registre d'utilisateurs et le mécanisme d'authentification, est un exemple des propriétés communes. Voir Fichier XML du descripteur de sécurité pour plus d'informations sur les propriétés de sécurité.

Pour configurer le fichier descripteur XML de sécurité dans un environnement Java SE, utilisez une option **-clusterSecurityFile** lorsque vous exécutez la commande **startOgServer**. Indiquez une valeur dans un format de fichier, tel que *samples_home*`/security_extauth/security3.xml`.

Fichier `security3.xml` :

Dans ce tutoriel, le fichier `security3.xml` se trouve dans le répertoire *samples_home*/`security_extauth`. Contenu du fichier `security3.xml` avec les commentaires supprimés :

```
<securityConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config/security ../objectGridSecurity.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config/security">

 <security securityEnabled="true">
  <authenticator
 className="com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginAuthenticator">
  </authenticator>
 </security>
</securityConfig>
```

Les propriétés suivantes sont définies dans le fichier `security3.xml` :

securityEnabled
: La propriété securityEnabled a la valeur `true` pour indiquer au serveur de catalogue que la sécurité globale WebSphere eXtreme Scale est activée.

authenticator
: L'authentificateur est configuré com.ibm.websphere.objectgrid.security.plugins.builtins. KeyStoreLoginAuthenticator class. Avec cette implémentation intégrée dans le plug-in Authenticator, l'ID utilisateur et le mot de passe sont transmis pour vérifier qu'il est configuré dans le fichier de clés. La classe KeyStoreLoginAuthenticator utilise un alias de module de connexion `KeyStoreLogin` et une configuration de connexion Java Authentication and Authorization Service (JAAS) est donc nécessaire.

`catServer3.props` file :

Le fichier de propriétés du serveur stocke les propriétés du serveur qui incluent ses propriétés de sécurité. Pour plus d'informations, voir Fichier de propriétés du serveur. Vous pouvez utiliser l'option **`-serverProps`** pour spécifier la propriété de serveur de catalogue lorsque vous exécutez la commande **`startOgServer`**. Pour ce tutoriel, un fichier `catServer3.props` se trouve dans le répertoire c. Contenu du fichier `catServer3.props` avec les commentaires supprimés :

```
securityEnabled=true
credentialAuthentication=Required
transportType=TCP/IP
secureTokenManagerType=none
authenticationSecret=ObjectGridDefaultSecret
```

securityEnabled
: La propriété securityEnabled a la valeur `true` pour indiquer que ce serveur de catalogue est un serveur sécurisé.

credentialAuthentication
: La propriété credentialAuthentication a la valeur `Requis`. Par conséquent, un client qui se connecte au serveur doit fournir des données d'identification. Dans le fichier de propriétés du client, la propriété credentialAuthentication a la valeur `Pris en charge` et le serveur reçoit donc les données d'identification envoyées par le client.

secureTokenManagerType
: La propriété secureTokenManagerType a la valeur `none` pour indiquer que la valeur secrète d'authentification n'est pas chiffrée lors du regroupement avec les serveurs existants.

authenticationSecret
La propriété authenticationSecret a la valeur `ObjectGridDefaultSecret`. Cette chaîne secrète est utilisée pour devenir membre du cluster de serveurs eXtreme Scale. Lorsqu'un serveur rejoint la grille de données, il est invité à présenter la chaîne secrète. Si la chaîne secrète du serveur qui rejoint le cluster correspond à la chaîne dans le serveur de catalogue, le serveur devient membre du cluster. Dans le cas contraire, la demande de jointure est rejetée.

transportType
La propriété transportType a la valeur `TCP/IP` initialement. Plus loin dans le tutoriel, la sécurité du transport est activée.

Fichier `xsjaas3.config` :

Etant donné que l'implémentation KeyStoreLoginAuthenticator utilise un module de connexion, vous devez configurer le modèle de connexion avec un fichier de configuration de connexion d'authentification JAAS. Contenu du fichier `xsjaas3.config` :

```
KeyStoreLogin{
com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginModule required
     keyStoreFile="samples_home/security_extauth/sampleKS3.jks" debug = true;
};
```

Si vous avez utilisé un emplacement pour *samples_home* autre que `/wxs_samples/`, vous devez mettre à jour l'emplacement de keyStoreFile. Cette configuration de connexion indique que le module com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginModule est utilisé en tant que module de connexion. Le fichier de clés est le fichier `sampleKS3.jks`.

L'exemple de fichier de clés `sampleKS3.jks` stocke deux ID utilisateur et les mots de passe : manager/manager1 et cashier/cashier1.

Vous pouvez utiliser les commandes suivantes **`keytool`** pour créer ce fichier de clés :

- `keytool -genkey -v -keystore ./sampleKS3.jks -storepass sampleKS1 -alias manager -keypass manager1 -dname CN=manager,O=acme,OU=OGSample -validity 10000`
- `keytool -genkey -v -keystore ./sampleKS3.jks -storepass sampleKS1 -alias operator -keypass operator1 -dname CN=operator,O=acme,OU=OGSample -validity 10000`

Démarrage du serveur de catalogue avec la sécurité activée :

Pour démarrer le serveur de catalogue, exécutez la commande **`startOgServer`** avec les paramètres **`-clusterFile`** et **`-serverProps`** à transmettre dans les propriétés de sécurité.

Utilisez une installation autonome de WebSphere eXtreme Scale pour exécuter le serveur de catalogue. Lors de l'utilisation de l'image d'installation autonome, vous devez utiliser le SDK IBM. Vous pouvez utiliser le logiciel SDK qui est inclus avec WebSphere Application Server en définissant la variable *JAVA_HOME* pour qu'elle pointe vers le SDK IBM. Par exemple, `set JAVA_HOME=racine_was/IBM/WebSphere/AppServer/java/`

1. Accédez au répertoire `bin`.
   ```
   cd rép_base_wxs/bin
   ```

2. Exécutez la commande **startOgServer**.

Linux UNIX

```
./startOgServer.sh cs1 -listenerPort 16809 -JMXServicePort 16099 -catalogServiceEndPoints
cs1:[HOST_NAME]:16601:16602 -clusterSecurityFile samples_home/security_extauth/security3.xml
-serverProps samples_home/security_extauth/catServer3.props -jvmArgs
-Djava.security.auth.login.config="samples_home/security_extauth/xsjaas3.config"
```

Windows

```
startOgServer.bat cs1 -listenerPort 16809 -JMXServicePort 16099 -catalogServiceEndPoints
cs1:[HOST_NAME]:16601:16602 -clusterSecurityFile samples_home/security_extauth/security3.xml
-serverProps samples_home/security_extauth/catServer3.props -jvmArgs
-Djava.security.auth.login.config="samples_home/security_extauth/xsjaas3.config"
```

Une fois que vous avez exécuté la commande **startOgServer**, un serveur sécurisé commence avec le port d'écoute 16809, le port client 16601, le port homologue 16602 et le port JMX 16099. Si un conflit de port existe, remplacez le numéro de port par un numéro de port non utilisé.

Arrêt d'un serveur de catalogue dont la sécurité est activée :

Utilisez la commande **stopOgServer** pour arrêter le serveur de catalogue.

1. Accédez au répertoire `bin`.

   ```
   cd rép_base_wxs/bin
   ```

2. Exécutez la commande **stopOgServer**. Linux UNIX

   ```
   stopOgServer.sh cs1 -catalogServiceEndPoints localhost:16809 -clientSecurityFile
   samples_home/security_extauth/client3.props
   ```

   Windows

   ```
   stopOgServer.bat cs1 -catalogServiceEndPoints localhost:16809 -clientSecurityFile
   samples_home/security_extauth/client3.props
   ```

Point de contrôle de la leçon :

Vous avez configuré la sécurité du serveur de catalogue en associant les fichiers `security3.xml`, `catServer3.props`, `xsjaas3.config` au service de catalogue.

Leçon 2.3 : Configuration de la sécurité du serveur de conteneur

Lorsqu'un serveur de conteneur se connecte au service de catalogue, le serveur de conteneur obtient toutes les configurations de sécurité qui sont configurées dans le fichier ObjectGrid XML de sécurité. Le fichier XML de sécurité ObjectGrid définit la configuration de l'authentificateur, la valeur de temporisation de la session de connexion et d'autres informations de configuration. Un serveur de conteneur dispose également de ses propres propriétés de sécurité dans le fichier de propriétés de serveur.

Configurez le fichier de propriétés du serveur avec la propriété JVM (Java virtual machine)-Dobjectgrid.server.props. Le nom de fichier spécifié pour cette propriété correspond à un chemin absolu, tel que *samples_home*`/security_extauth/server3.props`.

Dans ce tutoriel, les serveurs de conteneur sont hébergés dans les serveurs `xs1` et `xs2` du cluster `xsCluster`.

Fichier `server3.props` :

Le fichier `server3.props` se trouve dans le répertoire *samples_home*`/security_extauth/`. Contenu du fichier `server3.props` :

```
securityEnabled=true
credentialAuthentication=Required
secureTokenManagerType=none
authenticationSecret=ObjectGridDefaultSecret
```

securityEnabled
La propriété securityEnabled a la valeur `true` pour indiquer que ce serveur de conteneur est un serveur sécurisé.

credentialAuthentication
La propriété credentialAuthentication a la valeur `Required`. Par conséquent, un client qui se connecte au serveur doit fournir des données d'identification. Dans le fichier de propriétés du client, la propriété credentialAuthentication a la valeur `Supported` et le serveur reçoit donc les données d'identification envoyées par le client.

secureTokenManagerType
La propriété secureTokenManagerType a la valeur `none` pour indiquer que la valeur secrète d'authentification n'est pas chiffrée lors de la connexion aux serveurs existants.

authenticationSecret
La propriété authenticationSecret a la valeur `ObjectGridDefaultSecret`. Cette chaîne secrète est utilisée pour devenir membre du cluster de serveurs eXtreme Scale. Lorsqu'un serveur demande à rejoindre la grille de données, il est invité à présenter la chaîne secrète. Si la chaîne secrète du serveur qui rejoint le cluster correspond à la chaîne dans le serveur de catalogue, le serveur devient membre du cluster. Dans le cas contraire, la demande de jointure est rejetée.

Définition du fichier de propriétés du serveur avec des propriétés JVM :

Définition du fichier de propriétés des serveurs xs1 et xs2. Si vous n'utilisez pas la topologie du tutoriel, définissez le fichier de propriétés de serveur sur tous les serveurs d'applications que vous utilisez pour héberger les serveurs de conteneur.

1. Ouvrez la page de la machine virtuelle Java du serveur. **Serveurs** > **Serveurs d'applications WebSphere** > *server_name* > **Java et gestion des processus** > **Définition de processus** > **Java Virtual Machine**.
2. Ajoutez l'argument JVM générique :
   `-Dobjectgrid.server.props=`*samples_home*`/security_extauth/server3.props`
3. Cliquez sur **OK** et enregistrez les modifications.

Ajout du module de connexion personnalisé :

Le serveur de conteneur utilise la même implémentation KeyStoreAuthenticator que le serveur de catalogue. L'implémentation KeyStoreAuthenticator utilise un alias de module de connexion **`KeyStoreLogin`** pour que vous puissiez ajouter un module de connexion aux entrées de modèle de connexion des applications.

1. Dans la console d'administration WebSphere Application Server, cliquez sur **Sécurité** > **Sécurité globale** > **Java Authentication and Authorization Service**.
2. Cliquez sur **Connexions des applications**.
3. Cliquez sur **Nouveau**, ajoutez un alias `KeyStoreLogin`. Cliquez sur **Appliquer**.
4. Sous **Module de connexion JAAS**, cliquez sur **Nouveau**.
5. Entrez `com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginModule` comme nom de classe de module et choisissez **SUFFICIENT** comme stratégie d'authentification. Cliquez sur **Appliquer**.

6. Ajoutez la propriété personnalisée `keyStoreFile` avec la valeur *samples_home*`/security_extauth/sampleKS.jks`.
7. Facultatif : Ajoutez la propriété personnalisée `debug` avec la valeur `true`.
8. Enregistrez le fichier.

Point de contrôle de la leçon :

A présent, le serveur d'authentification est sécurisé WebSphere eXtreme Scale. En configurant cette sécurité, toutes les applications qui tentent de se connecter aux serveurs WebSphere eXtreme Scale doivent fournir des données d'identification. Dans ce tutoriel, KeyStoreLoginAuthenticator est l'authentificateur. En conséquence, le client doit fournir un nom d'utilisateur et un mot de passe.

Leçon 2.4 : Installation et exécution de l'exemple

Une fois l'authentification configurée, vous pouvez installer et exécuter l'exemple d'application.

Création d'une bibliothèque partagée pour le fichier `EmployeeData.jar` :

1. Dans la console d'administration de WebSphere Application Server, ouvrez la page **Bibliothèques partagées**. Cliquez sur **Environnement** > **Bibliothèques partagées**.
2. Choisissez la portée **cellule**.
3. Créez la bibliothèque partagée. Cliquez sur **Nouveau**. Entrez `EmployeeManagementLIB` pour le **nom**. Entrez le chemin d'accès au fichier `EmployeeData.jar` dans le chemin de classes, par exemple, *samples_home*`/WASSecurity/EmployeeData.jar`.
4. Cliquez sur **Appliquer**.

Installation de l'exemple :

1. Installez le fichier `EmployeeManagement_extauth.ear` sous le répertoire *samples_home*`/security_extauth`.

   Important : Le fichier `EmployeeManagement_extauth.ear` est différent du fichier *samples_home*`/WASSecurity/EmployeeManagement.ear`. La manière dont la session ObjectGrid est extraite a été mise à jour pour utiliser les données d'identification mises en cache dans le fichier des propriétés du client dans l'application `EmployeeManagement_extauth.ear`. Voir les commentaires dans la classe com.ibm.websphere.sample.xs.DataAccessor du projet *samples_home*`/WASSecurity/EmployeeManagementWeb` pour identifier le code mis à jour pour cette modification.

   a. Pour commencer l'installation, cliquez sur **Applications** > **Nouvelle application** > **Nouvelle application d'entreprise**. Choisissez le chemin détaillé pour l'installation de l'application.
   b. Dans l'étape d'**association des modules aux serveurs**, définissez le cluster `appCluster` pour installer le module EmployeeManagementWeb module.
   c. Dans l'étape d'**association des bibliothèques partagées**, sélectionnez le module `EmployeeManagementWeb`.
   d. Cliquez sur **Bibliothèques partagées de référence**. Sélectionnez la bibliothèque `EmployeeManagementLIB`.
   e. Associez le rôle `webUser` à **Tous authentifiés dans le domaine de l'application**.
   f. Cliquez sur **OK**.

   Les clients s'exécutent dans les serveurs `s1` et `s2` du cluster.

2. Installez l'exemple de fichier XSDeployment.ear qui se trouve dans le répertoire *samples_home*/WASSecurity.
   a. Pour commencer l'installation, cliquez sur **Applications** > **Nouvelle application** > **Nouvelle application d'entreprise**. Choisissez le chemin détaillé pour l'installation de l'application.
   b. Dans l'étape d'**association des modules aux serveurs**, définissez le cluster xsCluster pour installer le module Web XSDeploymentWeb.
   c. Dans l'étape d'**association des bibliothèques partagées**, sélectionnez le module XSDeploymentWeb.
   d. Cliquez sur **Bibliothèques partagées de référence**. Sélectionnez la bibliothèque EmployeeManagementLIB.
   e. Cliquez sur **OK**.

   Les serveurs xs1 et xs2 de ce cluster hébergent les serveurs de conteneur.
3. Vérifiez que le serveur de catalogue est démarré. Pour plus d'informations sur le démarrage d'un serveur de catalogue pour ce tutoriel, voir «Démarrage du serveur de catalogue avec la sécurité activée», à la page 120.
4. Redémarrez le cluster xsCluster. Lorsque xsCluster démarre, l'application XSDeployment démarre et un serveur de conteneur est démarré sur les serveurs xs1 et xs2 respectivement. Si vous examinez le fichier SystemOut.log des serveurs xs1 et xs2, le message suivant indique que le fichier des propriétés du serveur est chargé :

   ```
   CWOBJ0913I: Server property files have been loaded:
   samples_home/security_extauth/server3.props.
   ```
5. Redémarrez le cluster appClusters. Lorsque appCluster démarre, l'application EmployeeManagement démarre également. Si vous examinez le fichier SystemOut.log des serveurs s1 et s2, le message suivant indique que le fichier des propriétés du client est chargé.

   ```
   CWOBJ0924I: The client property file {0} has been loaded.
   ```

   Si vous utilisez WebSphere eXtreme Scale Version 7.0, le message CWOBJ9000I s'affiche en anglais et indique que le fichier des propriétés du client a été chargé. Si vous ne voyez pas le message attendu, vérifiez que vous avez configuré la propriété -Dobjectgrid.server.props ou -Dobjectgrid.client.props dans l'argument JVM. Si vous l'avez configurée, vérifiez que le tiret (-) est un caractère UTF.

Exécution de l'exemple d'application :

1. Exécutez le fichier management.jsp. Dans un navigateur Web, accédez à http://<*your_servername*>:<*port*>/EmployeeManagementWeb/management.jsp. Par exemple, vous pouvez utiliser l'URL http://localhost:9080/EmployeeManagementWeb/management.jsp.
2. Fournissez les informations d'authentification à l'application. Entrez les données d'identification de l'utilisateur que vous avez associé au rôle webUser. Par défaut, ce rôle utilisateur est associé à tous les utilisateurs authentifiés. Entrez un nom d'utilisateur et un mot de passe valides, tels que le nom d'utilisateur et le mot de passe d'administrateur. Une page pour afficher, ajouter, mettre à jour et supprimer des employés apparaît.
3. Affichez les employés. Cliquez sur **Afficher un employé**. Entrez emp1@acme.com comme adresse électronique et cliquez sur **Soumettre**. Un message indique que l'utilisateur est introuvable.

4. Ajoutez un employé. Cliquez sur **Ajouter un employé**. Entrez `emp1@acme.com` comme adresse électronique, `Joe` comme prénom et `Doe` comme nom. Cliquez sur **Soumettre**. Un message s'affiche pour indiquer qu'un employé avec l'adresse `emp1@acme.com` a été ajouté.
5. Affichez le nouvel employé. Cliquez sur **Afficher un employé**. Entrez `emp1@acme.com` comme adresse électronique avec des zones vides pour les nom et prénom, et cliquez sur **Soumettre**. Un message s'affiche pour indiquer que l'employé a été trouvé et les noms corrects figurent dans les zones du prénom et du nom.
6. Supprimez l'employé. Cliquez sur **Supprimer un employé**. Entrez `emp1@acme.com` et cliquez sur **Soumettre**. Un message s'affiche pour indiquer que l'employé a été supprimé.

Etant donné que le type de transport du serveur de catalogue est `TCP/IP`, vérifiez que le paramètre de transport sortant des serveurs s1 et s2 n'a pas la valeur `SSL requis`. Sinon, une exception se produit. Si vous examinez le fichier de sortie système du serveur de catalogue, `logs/cs1/SystemOut.log`, la sortie de débogage suivante indique l'authentification de fichier de clés :

```
SystemOut     0 [KeyStoreLoginModule] initialize: Successfully loaded key store
SystemOut     0 [KeyStoreLoginModule] login: entry
SystemOut     0 [KeyStoreLoginModule] login: user entered user name: manager
SystemOut     0   Print out the certificates:
...
```

Point de contrôle de la leçon :

Vous avez installé et exécuté l'exemple d'application.

Module 3 : Configuration de la sécurité du transport

Configuration de la sécurité du transport pour protéger le transfert des données entre les clients et les serveurs dans la configuration.

Dans le module précédent du tutoriel, vous avez activé l'authentification WebSphere eXtreme Scale. Avec l'authentification, une application qui tente de se connecter au serveur WebSphere eXtreme Scale doit fournir des données d'identification. Par conséquent, aucun client non authentifié ne peut se connecter au serveur WebSphere eXtreme Scale. Les clients doivent être une application authentifiée qui s'exécute dans une cellule WebSphere Application Server.

Avec la configuration jusqu'à ce module, le transfert de données entre les clients dans le cluster appCluster et les serveurs du cluster xsCluster n'est pas chiffré. Cette configuration peut être acceptable si vos clusters WebSphere Application Server sont installés sur les serveurs derrière un pare-feu. Toutefois, dans certains scénarios, le trafic non chiffré n'est pas accepté pour certaines raisons, même si la topologie est protégée par un pare-feu. Par exemple, une politique gouvernementale pourrait imposer de chiffrer le trafic. WebSphere eXtreme Scale prend en charge Transport Layer Security/Secure Sockets Layer (TLS/SSL) pour sécuriser la communication entre les nœuds finaux ObjectGrid, qui incluent des serveurs client, des serveurs de conteneur et des serveurs de catalogue.

Dans cet exemple de déploiement, les clients et les servers de conteneur eXtreme Scale s'exécutent tous dans l'environnement WebSphere Application Server. Les propriétés client ou serveur ne sont pas nécessaires pour configurer les paramètres SSL, car la sécurité du transport eXtreme Scale est gérée par les paramètres de transport CSIV2 (Application Server Common Secure Interoperability Protocol Version 2). Les serveurs WebSphere eXtreme Scale utilisent la même instance ORB

(Object Request Broker) que les serveurs d'applications où ils sont exécutés. Définissez tous les paramètres SSL des services client et de conteneur dans la configuration WebSphere Application Server en utilisant ces paramètres de transport CSIv2. Vous devez configurer les propriétés SSL dans le fichier des propriétés du serveur de catalogue.

Objectifs d'apprentissage

A la fin des leçons de ce module, vous saurez :

- configurer le transport entrant et sortant CSIv2 ;
- ajouter des propriétés SSL dans le fichier de propriétés du serveur de catalogue ;
- vérifier le fichier des propriétés ORB :
- exécuter l'exemple.

Durée

Ce module prend 60 minutes environ.

Prérequis

Cette étape du tutoriel repose sur les modules précédents. Etudiez les modules précédents du présent tutoriel avant de configurer la sécurité du transport.

Leçon 3.1 : Configuration du transport entrant et sortant CSIv2

Pour configurer Transport Layer Security/Secure Sockets Layer (TLS/SSL) pour le transport du serveur, affectez aux propriétés de transport entrant et sortant Common Secure Interoperability Protocol Version 2 (CSIv2) la valeur `SSL requis` pour tous les serveurs WebSphere Application Server qui hébergent des clients, des serveurs de catalogue et des serveurs de conteneur.

Dans l'exemple de topologie du tutoriel, vous devez définir ces propriétés pour les serveurs d'applications s1, s2, xs1 et xs2. Procédez comme suit pout définir les transports entrant et sortant de tous les serveurs de la configuration.

Définissez les transports entrant et sortant dans la console d'administration. Vérifiez que la sécurité administrative est activée.

- **WebSphere Application Server Version 6.1** : Cliquez sur **Sécurité** > **Sécuriser l'administration** > **Application** > **Sécurité RMI/IIOP** et remplacez le type de transport par **SSL requis**.
- **WebSphere Application Server Version 7.0** : Cliquez sur **Sécurité** > **Sécurité globale** > **Sécurité RMI/IIOP** > **Communications entrants CSIv2**. Dans la couche de transport CSIv2 remplacez le type de transport par **SSL requis**. Répétez cette étape pour configurer les communications sortantes CSIv2.

Vous pouvez utiliser les paramètres de sécurité de noeud final géré de manière centralisée, ou configurer des référentiels SSL. Voir Paramètres de transport entrant CSIV2 (Common Secure Interoperability Version 2) pour plus d'informations.

Leçon 3.2 : Ajout de propriétés SSL au fichier des propriétés du serveur de catalogue

Le serveur de catalogue fonctionne en dehors de WebSphere Application Server. Vous devez donc configurer les propriétés SSL dans le fichier des propriétés du serveur.

Vous configurez les propriétés SSL dans le fichier des propriétés du serveur aussi parce que le serveur de catalogue ne peut pas être géré par les paramètres de transport WebSphere Application Server Common Secure Interoperability Protocol Version 2 (CSIV2). Vous devez donc définir les propriétés SSL (Secure Sockets Layer) dans le fichier des propriétés du serveur de catalogue.

Propriétés SSL dans le fichier `catServer3.props` :

```
alias=default
contextProvider=IBMJSSE2
protocol=SSL
keyStoreType=PKCS12
keyStore=/racine_was/IBM/WebSphere/AppServer/profiles/
<deployment_manager_name>/config/cells/<cell_name>/nodes/
<node_name>/key.p12
keyStorePassword=WebAS
trustStoreType=PKCS12
trustStore=/racine_was/IBM/WebSphere/AppServer/profiles/
<deployment_manager_name>/config/cells/<cell_name>/nodes/
<node_name>/trust.p12
trustStorePassword=WebAS
clientAuthentication=false
```

Le fichier `catServer3.props` utilise le fichier de clés et le fichier de clés certifiées au niveau noeud par défaut WebSphere Application Server. Si vous déployez un environnement de déploiement plus complexe, vous devez choisir le fichier de clés et le fichier de clés certifiées corrects. Dans certains cas, vous devez créer un fichier de clés et un fichier de clés certifiées et importez les clés depuis les fichiers de clés des autres serveurs. Notez que la chaîne `WebAS` est le mot de passe par défaut des fichiers de clés et de clés certifiées WebSphere Application Server. Voir Configuration des certificats autosignés par défaut pour plus d'informations.

Ces entrées sont déjà incluses dans le fichier *samples_home*`/security_extauth/catServer3.props` sous forme de commentaires. Vous pouvez supprimer la mise en commentaire des entrées et effectuer les mises à jour correspondant à l'installation dans les variables *racine_was*, *<deployment_manager_name>*, *<cell_name>*, et *<node_name>*.

Après avoir défini les propriétés SSL, remplacez la valeur de la propriété transportType `TCP/IP` par `SSL-Required`.

Propriétés SSL dans le fichier `client3.props` :

Vous devez également configurer les propriétés SSL dans le fichier `client3.props`, car ce fichier est utilisé lorsque vous arrêtez le serveur de catalogue qui est en cours d'exécution en dehors de WebSphere Application Server.

Ces propriétés n'ont aucune incidence sur les serveurs client qui sont en cours d'exécution dans WebSphere Application Server, car ils utilisent les paramètres de transport WebSphere Application Server Common Security Interoperability Protocol Version 2 (CSIV2). Toutefois, lorsque vous arrêtez le serveur de catalogue, vous devez fournir un fichier de propriétés de client dans la commande **`stopOgServer`**. Définissez les propriétés suivantes dans le fichier <SAMPLES_HOME>/security_extauth/client3.props pour qu'elles correspondent aux valeurs définies ci-dessus dans le fichier catServer3.props :

```
#contextProvider=IBMJSSE2
#protocol=SSL
#keyStoreType=PKCS12
#keyStore=/racine_was/IBM/WebSphere/AppServer/profiles/
<deployment_manager_name>/config/cells/<cell_name>/nodes/
<node_name>/key.p12
#keyStorePassword=WebAS
#trustStoreType=PKCS12
```

```
#trustStore=/racine_was/IBM/WebSphere/AppServer/profiles/
<deployment_manager_name>/config/cells/<cell_name>/nodes/
<node_name>/trust.p12
#trustStorePassword=WebAS
```

Comme avec le fichier `catServer3.props`, vous pouvez utiliser les commentaires déjà fournis dans le fichier *samples_home*`/security_extauth/client3.props` avec les mises à jour correspondant aux variables *racine_was*, *<deployment_manager_name>*, *<cell_name>* et *<node_name>* pour qu'elles correspondent à votre environnement.

Point de contrôle de la leçon :

Vous avez configuré les propriétés SSL pour le serveur de catalogue.

Leçon 3.3 : Exécution de l'exemple

Redémarrez tous les serveurs et exécutez de nouveau le modèle d'application. Vous devriez être en mesure d'exécuter les étapes sans aucun problème.

Voir «Leçon 2.4 : Installation et exécution de l'exemple», à la page 123 pour plus d'informations sur l'exécution et l'installation de l'exemple d'application.

Module 4 : utilisation de l'autorisation JAAS (Java Authentication and Authorization Service) dans WebSphere Application Server

Maintenant que vous avez configuré l'authentification pour les clients, vous pouvez configurer l'autorisation de manière plus précise pour accorder aux utilisateurs des autorisations différentes. Par exemple, un "opérateur" peut être autorisé uniquement à afficher les données, alors qu'un "gestionnaire" peut exécuter toutes les opérations.

Après avoir authentifié un client, comme dans le module précédent dans ce tutoriel, vous pouvez attribuer des privilèges de sécurité par le biais des mécanismes d'autorisation eXtreme Scale. Le module précédent de ce tutoriel vous a montré comment activer l'authentification pour une grille de données à l'aide de l'intégration à WebSphere Application Server. Par conséquent, aucun client non authentifié ne peut se connecter aux serveurs eXtreme Scale ni envoyer des demandes au système. Toutefois, tous les clients authentifiés possèdent les mêmes permissions ou privilèges liés au serveur, tels que la lecture, l'écriture ou la suppression des données stockées dans les mappes ObjectGrid. Les clients peuvent également soumettre tout type de requête.

Cette partie du tutoriel explique comment utiliser l'autorisation eXtreme Scale pour attribuer différents privilèges aux utilisateurs authentifiés. WebSphere eXtreme Scale utilise un mécanisme d'autorisation basé sur l'autorisation. Vous pouvez affecter des catégories d'autorisations différentes qui sont représentées par des classes d'autorisation différentes. Ce module utilise la classe MapPermission. Pour la liste de toutes les propriétés possibles, voir Programmation d'autorisations client.

Dans WebSphere eXtreme Scale, la classe `com.ibm.websphere.objectgrid.security.MapPermission` représente les autorisations d'accès aux ressources eXtreme Scale , notamment les méthodes des interfaces ObjectMap ou JavaMap. WebSphere eXtreme Scale définit les chaînes de permission suivantes pour accéder aux méthodes des interfaces ObjectMap et JavaMap :

- **read** : accorde l'autorisation de lire les données de la mappe.
- **write** : accorde l'autorisation de lire les données de la mappe.

- **insert** : accorde l'autorisation d'insérer les données dans la mappe.
- **remove**: accorde l'autorisation de supprimer les données de la mappe.
- **invalidate** : accorde l'autorisation d'invalider les données dans la mappe.
- **all** : accorde toutes les autorisations ci-dessus.

L'autorisation se produit lorsqu'un client eXtreme Scale utilise une API d'accès aux données, telles que ObjectMap ,JavaMap, ou les API EntityManager. L'environnement d'exécution eXtreme Scale vérifie les autorisations d'exécution de la mappe correspondante lorsque la méthode est appelée. Si les autorisations d'accès requises ne sont pas accordées au client, une exception AccessControlException est générée. Ce module explique comment utiliser l'autorisation JAAS (Java Authentication and Authorization Service) pour accorder des autorisations d'accès à la mappe pour différents utilisateurs.

Objectifs d'apprentissage

A la fin des leçons de ce module, vous saurez :

- activer les autorisations pour WebSphere eXtreme Scale ;
- activer les autorisations utilisateur.

Durée

Ce module prend 60 minutes environ.

Leçon 4.1 : Activation de l'autorisation WebSphere eXtreme Scale

Pour activer l'autorisation dans WebSphere eXtreme Scale, vous devez activer la sécurité sur un ObjectGrid spécifique.

Pour activer l'autorisation sur l'ObjectGrid, vous devez affecter à l'attribut **securityEnabled** la valeur true pour cet ObjectGrid spécifique dans le fichier XML. Pour ce tutoriel, vous pouvez utiliser le fichier `XSDeployment_sec.ear` dans le répertoire *samples_home*`/WASSecurity` dont la sécurité est déjà définie dans le fichier `objectGrid.xml` ou vous pouvez modifier le fichier existant `objectGrid.xml` pour activer la sécurité. Cette leçon explique comment modifier le fichier pour activer la sécurité.

1. Facultatif : Extrayez les fichiers dans le fichier `XSDeployment.ear`, puis décompressez le fichier `XSDeploymentWeb.war`.
2. Facultatif : Ouvrez le fichier `objectGrid.xml` et affectez à l'attribut **securityEnabled** sur `true` sur le niveau ObjectGrid. Voir un exemple de cet attribut ci-dessous :

```
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">

    <objectGrids>
        <objectGrid name="Grid" txTimeout="15" securityEnabled="true">
               <backingMap name="Map1" />
        </objectGrid>
    </objectGrids>

</objectGridConfig>
```

   Si vous avez plusieurs ObjectGrids définis, vous devez définir cet attribut dans chaque grille.

3. Facultatif : Remodularisez les fichiers `XSDeploymentWeb.war` et `XSDeployment.ear` pour inclure vos modifications.
4. Obligatoire : Désinstallez le fichier `XSDeployment.ear`, puis installez la mise à jour `XSDeployment.ear`. Vous pouvez utiliser le fichier que vous avez modifié dans les étapes précédentes ou vous pouvez installer le fichier `XSDeployment_sec.ear` qui est fourni dans le répertoire *samples_home*/`WASSecurity`. Voir «Leçon 2.4 : Installation et exécution de l'exemple», à la page 123 pour plus d'informations sur l'installation de l'application.
5. Redémarrez tous les serveurs d'applications pour activer l'autorisation WebSphere eXtreme Scale.

Point de contrôle de la leçon :

Vous avez activé la sécurité sur l'ObjectGrid, ce qui permet également d'activer l'autorisation dans la grille de données.

Leçon 4.2 : Activation des autorisations utilisateur

Dans le module d'authentification de ce tutoriel, vous avez créé les deux utilisateurs `operator` et `manager`. Vous pouvez affecter des autorisations différentes à ces utilisateurs avec l'autorisation JASS (Java Authentication and Authorization Service).

Définition de la règle d'autorisation JAAS (Java Authentication and Authorization Service) en utilisant des principaux utilisateur :

Vous pouvez affecter des autorisations aux utilisateurs que vous avez créés. Affectez les autorisations de lecture `operator` uniquement à toutes les mappes. Affectez à l'utilisateur `manager` toutes les autorisations. Utilisez le fichier de règle d'autorisation JAAS pour accorder des autorisations aux principaux.

Editez le fichier d'autorisation JAAS. Le fichier `xsAuth3.policy` se trouve dans le répertoire *samples_home*/`security_extauth`.

```
grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
    principal javax.security.auth.x500.X500Principal
    "CN=operator,O=acme,OU=OGSample" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "Grid.Map1", "read";
};

grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
    principal javax.security.auth.x500.X500Principal
    "CN=manager,O=acme,OU=OGSample" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "Grid.Map1", "all";
};
```

Dans ce fichier, le codebase `http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction` est une URL réservée à ObjectGrid. Toutes les autorisations ObjectGrid accordées aux principaux doivent utiliser ce codebase spécial. Les autorisations suivantes sont affectées dans ce fichier :

- Le première instruction accorde l'autorisation de mappe `read` au principal `"CN=operator,O=acme,OU=OGSample"`. L'utilisateur `"CN=operator,O=acme,OU=OGSample"` dispose uniquement de l'autorisation de lecture sur la mappe Map1 dans l'instance Grid ObjectGrid.
- La seconde instruction accorde toutes les autorisations de mappe au principal `"CN=manager,O=acme,OU=OGSample"`. L'utilisateur `"CN=manager,O=acme,OU=OGSample"` dispose uniquement de l'autorisation de lecture sur la mappe Map1 dans l'instance Grid ObjectGrid.

Définition du fichier de règle d'autorisation JAAS à l'aide de propriétés JVM :

Procédez comme suit pour définir les propriétés JVM pour les serveurs xs1 et xs2qui se trouvent dans le cluster xsCluster. Si vous utilisez une topologie qui est différente de l'exemple de topologie qui est utilisé dans ce tutoriel, définissez le fichier sur tous vos serveurs de conteneur.

1. Dans la console d'administration, cliquez sur **Serveurs** > **Serveurs d'applications** > *server_name* > **Java t gestion de processus** > **Définition de processus** > **Java virtual machine.**
2. Ajoutez les arguments JVM génériques suivants :

   ```
   -Djava.security.auth.policy=samples_home/security_extauth/xsAuth3.policy
   ```
3. Cliquez sur **OK** et enregistrez les modifications.

Exécution de l'exemple d'application pour tester les autorisations :

Vous pouvez utiliser l'exemple d'application pour tester les paramètres d'autorisation. Le gestionnaire continue de disposer de toutes les autorisations dans la mappe Map1, y compris des autorisations d'affichage et d'ajout d'employés. L'opérateur doit pouvoir afficher uniquement les employés, car seule l'autorisation de lecture lui a été affectée.

1. Redémarrez tous les serveurs d'applications qui exécutent des serveurs de conteneur. Pour ce tutoriel, redémarrez les serveurs `xs1` et `xs2`.
2. Ouvrez l'application `EmployeeManagementWeb`. Dans un navigateur Web, ouvrez `http://<host>:<port>/EmployeeManagermentWeb/management.jsp`.
3. Connectez-vous à l'application en utilisant un nom et un mot de passe utilisateur.
4. Essayez d'afficher un employé. Cliquez sur **Afficher un employé** et recherchez l'adresse électronique `authemp1@acme.com`. Un message indique que l'utilisateur est introuvable.
5. Ajoutez un employé. Cliquez sur **Ajouter un employé**. Ajoutez l'adresse électronique `authemp1@acme.com`, le prénom `Joe` et le nom `Doe`. Cliquez sur **Soumettre**. Un message indique que l'employé a été ajouté.
6. Editez *samples_home*`/security_extauth/client3.props`. Remplacez la valeur `manager manager1` de la propriété credentialGeneratorProps par `operator operator1`. Après avoir modifié le fichier, le servlet utilise le nom d'utilisateur "operator" et le mot de passe "operator1" pour s'authentifier sur les serveurs WebSphere eXtreme Scale.
7. Redémarrez le cluster appCluster pour appliquer les modifications dans le fichier *samples_home*`/security_extauth/client3.props`.
8. Essayez d'afficher un employé. Cliquez sur **Afficher un employé** et recherchez l'adresse électronique `authemp1@acme.com`. L'employé s'affiche.
9. Ajoutez un employé. Cliquez sur **Ajouter un employé**. Ajoutez l'adresse électronique `authemp2@acme.com`, le prénom `Joe` et le nom `Doe`. Cliquez sur **Soumettre**. Le message suivant s'affiche :

   ```
   An exception occurs when Add the employee. See below for detailed exception messages.
   ```

   Texte de l'exception détaillée :

   ```
   java.security.AccessControlException: Access denied
   (com.ibm.websphere.objectgrid.security.MapPermission Grid.Map1 insert)
   ```

   Ce message s'affiche, car l'`opérateur` n'est pas autorisé à insérer des données dans la mappe Map1.

Si vous utilisez une version de WebSphere Application Server antérieure à la version 7.0.0.11, une erreur java.lang.StackOverflowError peut s'afficher sur le

serveur de conteneur. Elle est provoquée par IBM Developer Kit. Le problème est résolu dans IBM Developer Kit fourni avec WebSphere Application Server Version 7.0.0.11 et les versions suivantes.

Point de contrôle de la leçon :

Dans cette leçon, vous avez configuré l'autorisation en attribuant des autorisations à des utilisateurs spécifiques.

Module 5 : Utilisation de l'utilitaire `xscmd` pour surveiller les grilles de données et les mappes

Vous pouvez utiliser l'utilitaire **xscmd** pour afficher les grilles de données primaires et les tailles de mappe de la grille de données `Gride`. L'outil **xscmd** utilise le bean géré pour interroger tous les artefacts de grille de données, tels que les fragments primaires, les fragments de réplique, des serveurs de conteneur, les tailles de mappe et d'autres données.

Dans ce tutoriel, les serveurs de catalogue s'exécutent comme serveurs Java SE autonomes. Les serveurs de conteneur s'exécutent sur des serveurs d'applications WebSphere Application Server.

Pour le serveur de catalogue, un serveur MBean est créé dans la machine JVM (Java virtual machine). Lorsque vous utilisez l'outil **xscmd** sur le serveur de catalogue, la sécurité WebSphere eXtreme Scale est utilisée.

Pour les serveurs de conteneur, l'exécution WebSphere eXtreme Scale enregistre les beans gérés (MBean) avec le serveur MBean créé par l'exécution WebSphere Application Server. La sécurité est utilisée par l'outil **xscmd** est fournie par la sécurité MBean WebSphere Application Server.

1. A l'aide d'un outil de ligne de commande, ouvrez le répertoire *DMGR_PROFILE*`/bin`.
2. Exécutez l'outil **xscmd**. Utilisez les paramètres **-c showPlacement -st P** comme dans les exemples suivants :

   Linux UNIX

   ```
   xscmd.sh -c listObjectGridPlacement -cep localhost:16099 -g Grid -ms mapSet -sf P
   -user manager -pwd manager1
   ```

   Windows

   ```
   xscmd.bat -c listObjectGridPlacement -cep localhost:16099 -g Grid -m mapSet -sf P
   -user manager -pwd manager1
   ```

   Le nom d'utilisateur et le mot de passe sont transmis au serveur de catalogue pour l'authentification.
3. Affichez les résultats de la commande.

   ```
   *** Showing all primaries for grid - Grid & mapset - mapSet
   Partition Container Host Server
   0 myCell02\myNode04\xs2_C-1 myhost.mycompany.com myCell02\myNode04\xs2
   1 myCell02\myNode04\xs2_C-1 myhost.mycompany.com myCell02\myNode04\xs2
   2 myCell02\myNode04\xs2_C-1 myhost.mycompany.com myCell02\myNode04\xs2
   3 myCell02\myNode04\xs2_C-1 myhost.mycompany.com myCell02\myNode04\xs2
   4 myCell02\myNode04\xs2_C-1 myhost.mycompany.com myCell02\myNode04\xs2
   ```
4. Exécutez l'outil **xscmd**. Utilisez le paramètre **-c showMapSizes** comme dans les exemples suivants :

   Linux UNIX

```
xscmd.sh -c showMapSizes -cep localhost:16099 -g Grid -ms mapSet  -user manager -pwd manager1
```

Windows

```
xscmd.bat -c showMapSizes -cep localhost:16099 -g Grid -ms mapSet -user manager -pwd manager1
```

Le nom d'utilisateur et le mot de passe sont transmis au serveur de catalogue pour être authentifiés. Après avoir exécuté la commande, vous êtes invité à indiquer l'ID utilisateur et le mot de passe WebSphere Application Server pour l'authentification dans WebSphere Application Server. Vous devez fournir ces informations de connexion, car l'option **-c showMapSizes** obtient la taille de mappe de chaque serveur de conteneur qui nécessite la sécurité WebSphere Application Server.

5. Facultatif : Vous pouvez modifier le fichier `PROFILE/properties/sas.client.props` pour exécuter la commande sans que l'ID utilisateur et le mot de passe soient nécessaires. Remplacez la propriété com.ibm.CORBA.loginSource `prompt` par `properties`, puis indiquez l'ID utilisateur et le mot de passe. Voici un exemple des propriétés dans le fichier `PROFILE/properties/sas.client.props` :

   ```
   com.ibm.CORBA.loginSource=properties
   # RMI/IIOP user identity
   com.ibm.CORBA.loginUserid=Admin
   com.ibm.CORBA.loginPassword=xxxxxx
   ```

6. Facultatif : Si vous utilisez la commande **xscmd** sur une installation autonome WebSphere eXtreme Scale, vous devez ajouter les options suivantes :
   - Si vous utilisez la sécurité WebSphere eXtreme Scale :

     ```
     -user
     -pwd
     ```

   - Si vous utilisez la sécurité WebSphere eXtreme Scale avec la génération de données d'identification personnalisée :

     ```
     -user
     -pwd
     -cgc
     -cgp
     ```

   - Si SSL est activé :

     ```
     -tt
     -cxpv
     -prot
     -ks
     -ksp
     -kst
     -ts
     -tsp
     -tst
     ```

   Si la sécurité WebSphere eXtreme Scale et SSL sont tous les deux activés, les deux groupes de paramètres sont requis.

Point de contrôle de la leçon

Vous avez utilisé l'outil **xscmd** pour surveiller les grilles de données et les mappes dans la configuration.

Tutoriel : Exécution des ensembles eXtreme Scale dans la structure OSGi

L'exemple OSGi repose sur les exemples de sérialiseurs Google Protocol Buffers. A la fin de ce groupe de leçons, vous aurez exécuter les exemples de plug-ins du sérialiseur dans l'infrastructure OSGi.

Objectifs d'apprentissage

Cet exemple montre les ensembles OSGi. Le plug-in de sérialiseur est secondaire et il n'est pas requis. L'exemple OSGi est disponible dans la galerie des exemples WebSphere eXtreme Scale. Vous devez télécharger l'exemple et l'extraire dans le répertoire *wxs_home*/samples. Le répertoire racine de l'exemple OSGi est wxs_home/samples/OSGiProto.

L'exemple de sérialiseur Google Buffers Protocole se trouve dans le répertoire *wxs_home*/samples/SerializerProto.

L'exemple de sérialiseur Binary JSON (BSON) se trouve dans le répertoire *wxs_home*/samples/SerializerBSON.

Les exemples de commande dans ce tutoriel suppose que vous exécutez le système d'exploitation UNIX. Vous devez ajuster l'exemple de commande pour l'exécuter sur un système d'exploitation Windows.

A la fin des leçons de ce tutoriel, vous comprendrez les concepts des exemples OSGi et saurez :

- installer l'ensemble de serveur WebSphere eXtreme Scale dans le conteneur OSGi pour démarrer le serveur eXtreme Scale ;
- configurer votre environnement de développement eXtreme Scale pour exécuter l'exemple de client ;
- utiliser la commande xscmd pour interroger le classement de l'exemple d'ensemble, le mettre à niveau vers un nouveau classement de services et vérifier le nouveau classement de services.

Durée

Ce module prend 60 minutes environ.

Prérequis

Dans ce tutoriel, vous devez télécharger et extraire les exemples de sérialiseurs et :

- Installer et extraire le produit eXtreme Scale
- Configurer l'environnement Eclipse Equinox

Introduction : Démarrage et configuration du serveur eXtreme Scale et du conteneur pour exécuter les plug-ins dans la structure OSGi

Dans ce tutoriel, vous allez démarrer un serveur eXtreme Scale dans l'infrastructure OSGi, démarrer un conteneur eXtreme Scale et connecter les exemples de plug-ins avec l'environnement d'exécution eXtreme Scale.

Objectifs d'apprentissage

A la fin des leçons de ce tutoriel, vous comprendrez les concepts des exemples OSGi et saurez :

- installer l'ensemble de serveur WebSphere eXtreme Scale dans le conteneur OSGi pour démarrer le serveur eXtreme Scale ;
- configurer l'environnement de développement eXtreme Scale pour exécuter l'exemple de client ;

- utiliser la commande xscmd pour interroger le classement de l'exemple d'ensemble, le mettre à niveau vers un nouveau classement de services et vérifier le nouveau classement de services.

Durée

Ce tutorial prend 60 minutes environ. Si vous explorez d'autres concepts liés à ce tutoriel, il peut prendre plus de temps.

Niveau de compétence

Intermédiaire

Audience

Les développeurs et les administrateurs qui veulent créer, installer et exécuter des ensembles eXtreme Scale dans l'infrastructure OSGi.

Configuration requise

- Client de ligne de commande Luminis OSGi Configuration Admin, version 0.2.5
- Apache Felix File Install, version 3.0.2
- Lorsque vous utilisez Eclipse Gemini en tant que fournisseur de conteneur Blueprint, les éléments suivants sont requis :
  - Eclipse Gemini Blueprint, version 1.0.0
  - Spring Framework, version 3.0.5
  - SpringSource AOP Alliance API, version 1.0.0
  - SpringSource Apache Commons Logging, version 1.1.1
- Lorsque vous utilisez Apache Aries en tant que fournisseur du conteneur Blueprint, vous devez disposer de la configuration suivante :
  - Dernière image instantanée Aries
  - Bibliothèque ASM
  - Consignation PAX

Prerequis

Pour pouvoir exécuter ce tutoriel, vous devez télécharger l'exemple et l'extraire dans le répertoire `wxs_home/samples`. Le répertoire racine de l'exemple OSGi est `wxs_home/samples/OSGiProto`.

Résultats attendus

A la fin de ce tutoriel, vous aurez installé les exemples d'ensembles et exécuté un client eXtreme Scale pour insérer des données dans la grille. Vous serez également amené à interroger et mettre à jour ces exemples d'ensembles en utilisant les fonctions dynamiques que fournit le conteneur OSGi.

Module 1 : Préparation de l'installation et de la configuration des ensembles de serveur eXtreme Scale

Effectuez ce module pour explorer les exemples d'ensembles OSGi et examiner les fichiers de configuration que vous utilisez pour configurer le serveur eXtreme Scale.

Objectifs d'apprentissage

A la fin des leçons de ce module, vous comprendrez les concepts et saurez :

- localiser et explorer les ensembles qui sont inclus dans le modèle OSG ;
- Examinez les fichiers de configuration utilisés pour configurer la grille et le serveur eXtreme Scale.

Leçon 1.1 : Explication des exemples d'ensembles OSGi

Suivez cette leçon pour localiser et explorer les ensembles fournis dans l'exemple OSGi.

Exemples d'ensembles OSGi :

Hormis les ensembles qui sont configurés dans le fichier `config.ini`, qui est indiqué dans la rubrique sur la configuration de l'environnement Eclipse Equinox, les ensembles supplémentaires suivants sont utilisés dans le modèle OSGi :

objectgrid.jar
Ensemble d'exécution de serveur WebSphere eXtreme Scale. Cet ensemble se trouve dans le répertoire *wxs_home*/lib.

com.google.protobuf_2.4.0a.jar
Ensemble Google Protocol Buffers, version 2.4.0a. Cet ensemble se trouve dans le répertoire *wxs_sample_osgi_root*/lib.

ProtoBufSamplePlugins-1.0.0.jar
Version 1.0.0 de l'ensemble de plug-in utilisateur avec l'exemple ObjectGridEventListener et les implémentations de plug-in MapSerializerPlugin. Cet ensemble se trouve dans le répertoire *wxs_sample_osgi_root*/lib. Les services sont configurés avec le classement de service 1.

Cette version utilise le XML Blueprint standard pour configurer les services de plug-in eXtreme Scale. La classe de service est une classe implémentée par l'utilisateur pour l'interface WebSphere eXtreme Scale, com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactory. La classe implémentée par l'utilisateur crée un bean pour chaque demande et fonctionne de la même manière qu'un bean de portée prototype.

ProtoBufSamplePlugins-2.0.0.jar
Version 2.0.0 de l'ensemble de plug-in utilisateur avec l'exemple ObjectGridEventListener et les implémentations de plug-in MapSerializerPlugin. Cet ensemble se trouve dans le répertoire *wxs_sample_osgi_root*/lib. Les services sont configurés avec le classement de service 2.

Cette version utilise le XML Blueprint standard pour configurer les services de plug-in eXtreme Scale. La classe de service utilise une classe intégrée WebSphere eXtreme Scale, com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactoryImpl, qui utilise le service BlueprintContainer. En utilisant la configuration XML Blueprint standard, les beans peuvent être configurés en tant que portée singleton ou portée prototype. Le bean n'est pas configuré en tant que portée de fragment.

ProtoBufSamplePlugins-Gemini-3.0.0.jar
Version 3.0.0 de l'ensemble de plug-in utilisateur avec l'exemple ObjectGridEventListener et les implémentations de plug-in

MapSerializerPlugin. Cet ensemble se trouve dans le répertoire *wxs_sample_osgi_root*/lib. Les services sont configurés avec le classement de service 3.

Cette version utilise le XML Blueprint XML d'Eclipse Gemini pour configurer les services de plug-in eXtreme Scale. La classe de service utilise une classe de service intégrée, WebSphere eXtreme Scale, com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactoryImpl, qui utilise le service BlueprintContainer. Pour configurer un bean de portée de fragment, utilisez une approche Gemini. Cette version configure le bean myShardListener comme bean de portée de fragment en fournissant {http://www.ibm.com/schema/objectgrid}shard comme valeur de portée et en configurant un attribut factice pour que Gemini reconnaisse la portée personnalisée. Le problème Eclipse est généré par : https://bugs.eclipse.org/bugs/show_bug.cgi?id=348776

ProtoBufSamplePlugins-Aries-4.0.0.jar
Version 4.0.0 de l'ensemble de plug-in utilisateur avec l'exemple ObjectGridEventListener et les implémentations de plug-in MapSerializerPlugin. Cet ensemble se trouve dans le répertoire *wxs_sample_osgi_root*/lib. Les services sont configurés avec le classement de service 4.

Cette version utilise le XML Blueprint standard pur configurer les services de plug-in eXtreme Scale. La classe de service utilise une classe intégrée WebSphere eXtreme Scale, com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactoryImpl, qui utilise le service BlueprintContainer. En utilisant la configuration XML Blueprint standard, les beans peuvent être configurés en utilisant une porté personnalisée. Cette version configure myShardListenerbean comme bean à portée de fragment en fournissant {http://www.ibm.com/schema/objectgrid}shard comme valeur de portée.

ProtoBufSamplePlugins-Activator-5.0.0.jar
Version 5.0.0 de l'ensemble de plug-in utilisateur avec l'exemple ObjectGridEventListener et les implémentations de plug-in MapSerializerPlugin. Cet ensemble se trouve dans le répertoire *wxs_sample_osgi_root*/lib. Les services sont configurés avec le classement de service 5.

Cette version n'utilise pas du tout le conteneur Blueprint. Dans cette version, les services sont enregistrés à l'aide de l'enregistrement de service OSGi. La classe de service est une classe implémentée par l'utilisateur pour l'interface WebSphere eXtreme Scale, com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactory. La classe implémentée par l'utilisateur crée un bean pour chaque demande. Elle fonctionne d'une manière similaire à un bean à portée prototype.

Point de contrôle de la leçon :

En explorant les ensembles qui sont fournis avec le modèle OSGi, vous pouvez mieux comprendre la procédure de développement de vos propres implémentations qui s'exécutent dans le conteneur OSGi.

Vous avez appris :

- les ensembles inclus avec l'exemple OSGi ;
- l'emplacement de ces ensembles ;
- l'élément utilisé pour configurer le classement de service de chaque ensemble.

Leçon 1.2 : Description des fichiers de configuration OSGi

L'exemple OSGi contient trois fichiers de configuration. Vous utilisez ces fichiers pour démarrer et configurer la grille WebSphere eXtreme Scale et le serveur.

Fichiers de configuration OSGi :

Dans cette leçon, vous allez explorer les fichiers de configuration suivants :

- `collocated.server.properties`
- `protoBufObjectGrid.xml`
- `protoBufDeployment.xml`

collocated.server.properties

Une configuration de serveur est nécessaire pour démarrer un serveur. Lorsque l'ensemble de serveur eXtreme Scale est démarré, il ne démarre pas un serveur. Il attend la création du PID de configuration `com.ibm.websphere.xs.server` avec un fichier de propriétés de serveur. Ce fichier de propriétés du serveur indique le nom du serveur, le numéro de port et d'autres propriétés du serveur.

Dans la plupart des cas, vous créerez une configuration pour définir le fichier des propriétés du serveur. Dans de rares cas, vous pouvez vouloir uniquement démarrer un serveur avec chaque propriété affectée d'une valeur par défaut. Dans ce cas, vous pouvez créer une configuration appelée `com.ibm.websphere.xs.server` avec la valeur `default`.

Pour plus d'informations sur le fichier des propriétés du serveur, voir la rubrique Fichier de propriétés du serveur.

L'exemple OSGi inclut le fichier de propriétés de serveur *wxs_sample_osgi_root*`/server/properties/collocated.server.properties`. Ce fichier de propriétés démarre un service de catalogue unique et un serveur de conteneur dans le processus d'infrastructure OSGi. Les clients eXtreme Scale se connectent sur le port 2809 et les clients JMX, sur le port 1099. Contenu du fichier de propriétés de serveur :

```
serverName=collocatedServer
isCatalog=true
catalogClusterEndPoints=collocatedServer:localhost:6601:6602
traceSpec=ObjectGridOSGi=all=enabled
traceFile=logs/trace.log
listenerPort=2809
JMXServicePort=1099
```

protoBufObjectGrid.xml

L'exemple de fichier XML descripteur `protoBufObjectGrid.xml` ObjectGrid contient les éléments suivants avec les commentaires supprimés.

```
<objectGridConfig>
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">

    <objectGrids>
        <objectGrid name="Grid" txTimeout="15">

            <bean id="ObjectGridEventListener"
                osgiService="myShardListener"/>

            <backingMap name="Map" readOnly="false"
                lockStrategy="PESSIMISTIC" lockTimeout="5"
                copyMode="COPY_TO_BYTES"
```

```
                pluginCollectionRef="serializer"/>

        </objectGrid>
    </objectGrids>

    <backingMapPluginCollections>
        <backingMapPluginCollection id="serializer">
            <bean id="MapSerializerPlugin"
                osgiService="myProtoBufSerializer"/>"/>
        </backingMapPluginCollection>
    </backingMapPluginCollections>
 </objectGridConfig>
```

Il existe deux plug-ins configurés dans ce fichier descripteur XML ObjectGrid :

ObjectGridEventListener
Plug-in au niveau du fragment. Pour chaque instance ObjectGrid, il existe une instance de ObjectGridEventListener. Elle est configurée pour utiliser le service OSGi myShardListener. Cela signifie que lorsque la grille est créée, le plug-in ObjectGridEventListener utilise le service OSGi myShardListener avec le classement de service le plus élevé disponible.

MapSerializerPlugin
Plug-in au niveau de la mappe. Pour la mappe de sauvegarde nommée `Map`, il existe un plug-in MapSerializerPlugin configuré. Il est configuré pour utiliser le service OSGI myProtoBufSerializer. Cela signifie que lorsque la grille est créée, le plug-in MapSerializerPlugin utilise le service, myProtoBufSerializer, avec le classement de service le plus élevé disponible.

protoBufDeployment.xml

Le fichier descripteur XML de déploiement décrit la stratégie de déploiement pour la grille `Grid` qui utilise cinq partitions. Reportez-vous à l'exemple de code suivant de ce fichier XML :

```
<deploymentPolicy>
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

  <objectgridDeployment objectgridName="Grid">
    <mapSet name="MapSet" numberOfPartitions="5">
      <map ref="Map"/>
    </mapSet>
  </objectgridDeployment>
</deploymentPolicy>
```

blueprint.xml

Comme alternative à l'utilisation du fichier `collocated.server.properties` en association avec le PID de configuration, `com.ibm.websphere.xs.server`, vous pouvez inclure le fichier XML ObjectGrid et des fichiers XML de déploiement dans un ensemble OSGi, avec un fichier XML Blueprint, comme dans l'exemple suivant :

```
<blueprint>
   xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
       xmlns:objectgrid="http://www.ibm.com/schema/objectgrid"
       default-activation="lazy">

    <objectgrid:server id="server" isCatalog="true"
                name="server"
                tracespec="ObjectGridOSGi=all=enabled"
                tracefile="C:/Temp/logs/trace.log"
                workingDirectory="C:/Temp/working"
                jmxport="1099">
        <objectgrid:catalog host="localhost" port="2809"/>
    </objectgrid:server>
```

```
    <objectgrid:container id="container"
  objectgridxml="/META-INF/objectgrid.xml"
       deploymentxml="/META-INF/deployment.xml"
  server="server"/>
</blueprint>
```

Point de contrôle de la leçon :

Dans cette leçon, vous avez découvert les fichiers de configuration qui sont utilisées dans l'exemple OSGi. Maintenant, lorsque vous démarrez et configurez la grille eXtreme Scale et le serveur, vous savez quels fichiers sont utilisés dans ces processus et comment ces fichiers interagissent avec vos plug-ins dans l'infrastructure OSGi.

Module 2 : Installation et démarrage des ensembles eXtreme Scale dans l'infrastructure OSGi

Utilisez les modules de cette leçon pour installer l'ensemble de serveur eXtreme Scale dans le conteneur OSGi et démarrer le serveur WebSphere eXtreme Scale.

Le démarrage du serveur dans l'infrastructure OSGi n'implique pas que les ensembles OSGi sont prêts à être exécutés. Vous devez configurer les propriétés du serveur et les conteneurs de sorte que les ensembles OSGi que vous installez soient reconnus et puissent s'exécuter correctement.

Objectifs d'apprentissage

A la fin des leçons de ce module, vous comprendrez les concepts et saurez :

- installer les ensembles eXtreme Scale en utilisant la console OSGi Equinox ;
- configurer le serveur eXtreme Scale ;
- configurer le conteneur eXtreme Scale ;
- démarrer les exemples d'ensembles eXtreme Scale.

Prérequis

Pour pouvoir exécuter ce module, vous devez effectuer préalablement les tâches suivantes :

- Installer et extraire le produit eXtreme Scale
- Définir l'environnement Eclipse Equinox

Vous devez également préparer l'accès aux fichiers suivants pour suivre les leçons de ce module :

- Ensemble `objectgrid.jar`. Vous installez cet ensemble eXtreme Scale.
- `Fichier collocated.server.properties`. Vous ajoutez les propriétés du serveur à ce fichier de configuration.
- Vous pouvez envisager d'installer et de démarrer les ensembles suivants :
- `protobuf-java-2.4.0a-bundle.jar`
- `ProtoBufSamplePlugins-1.0.0.jar`
- `ProtoBufSamplePlugins-2.0.0.jar`

Leçon 2.1 : Démarrage de la console et installation de l'ensemble de serveur eXtreme Scale

Dans cette leçon, vous utilisez la console Equinox OSGi pour lancer et installer un WebSphere eXtreme Scale

1. Utilisez la commande suivante pour démarrer la console OSGi Equinox :

```
cd equinox_root

java -jar
plugins\org.eclipse.osgi_3.6.1.R36x_v20100806.jar
-console
```

2. Une fois la console OSGi démarrée, exécutez la commande ss dans la console ; les ensembles suivants sont démarrés :

Sortie Eclipse Gemini :

```
osgi> ss
Framework is launched.
id State Bundle
0 ACTIVE org.eclipse.osgi_3.6.1.R36x_v20100806
1 ACTIVE org.eclipse.osgi.services_3.2.100.v20100503
2 ACTIVE org.eclipse.osgi.util_3.2.100.v20100503
3 ACTIVE org.eclipse.equinox.cm_1.0.200.v20100520
4 ACTIVE com.springsource.org.apache.commons.logging_1.1.1
5 ACTIVE com.springsource.org.aopalliance_1.0.0
6 ACTIVE org.springframework.aop_3.0.5.RELEASE
7 ACTIVE org.springframework.asm_3.0.5.RELEASE
8 ACTIVE org.springframework.beans_3.0.5.RELEASE
9 ACTIVE org.springframework.context_3.0.5.RELEASE
10 ACTIVE org.springframework.core_3.0.5.RELEASE
11 ACTIVE org.springframework.expression_3.0.5.RELEASE
12 ACTIVE org.apache.felix.fileinstall_3.0.2
13 ACTIVE net.luminis.cmc_0.2.5
14 ACTIVE org.eclipse.gemini.blueprint.core_1.0.0.RELEASE
15 ACTIVE org.eclipse.gemini.blueprint.extender_1.0.0.RELEASE
16 ACTIVE org.eclipse.gemini.blueprint.io_1.0.0.RELEASE
```

Sortie Apache Aries :

```
osgi> ss
Framework is launched.
id State Bundle
0 ACTIVE org.eclipse.osgi_3.6.1.R36x_v20100806
1 ACTIVE org.eclipse.osgi.services_3.2.100.v20100503
2 ACTIVE org.eclipse.osgi.util_3.2.100.v20100503
3 ACTIVE org.eclipse.equinox.cm_1.0.200.v20100520
4 ACTIVE org.ops4j.pax.logging.pax-logging-api_1.6.3
5 ACTIVE org.ops4j.pax.logging.pax-logging-service_1.6.3
6 ACTIVE org.objectweb.asm.all_3.3.0
7 ACTIVE org.apache.aries.blueprint_0.3.2.SNAPSHOT
8 ACTIVE org.apache.aries.util_0.4.0.SNAPSHOT
9 ACTIVE org.apache.aries.proxy_0.4.0.SNAPSHOT
10 ACTIVE org.apache.felix.fileinstall_3.0.2
11 ACTIVE net.luminis.cmc_0.2.5
```

3. Installez l'ensemble objectgrid.jar. Pour démarrer un serveur dans la machine JVM (Java virtual machine), vous devez installer un ensemble de serveur eXtreme Scale. Cet ensemble de serveur eXtreme Scale peut démarrer un serveur et créer des conteneurs. Utilisez la commande suivante pour installer le fichier objectgrid.jar :

```
osgi> install file:///wxs_home/lib/objectgrid.jar
```

Reportez-vous à l'exemple suivant :

```
osgi> install
file:///opt/wxs/ObjectGrid/lib/objectgrid.jar
```

Equinox affiche son ID d'ensemble, par exemple :

```
Bundle id is 19
```

A faire : Votre ID d'ensemble peut être différent. Le chemin de ficher doit être une adresse URL absolue dans le chemin de l'ensemble. Les chemins relatifs ne sont pas pris en charge.

Point de contrôle de la leçon :

Dans cette leçon, vous avez utilisé la console OSGi Equinox pour installer l'ensemble `objectgrid.jar` que vous allez utiliser pour démarrer et créer ensuite un conteneur dans ce tutoriel.

Leçon 2.2 : Personnalisation et configuration du serveur eXtreme Scale

Suivez cette leçon pour personnaliser et ajouter les propriétés au serveur WebSphere eXtreme Scale.

1. Editez le fichier `wxs_sample_osgi_root/server/properties/collocated.server.properties`.
   a. Remplacez la propriété workingDirectory par `equinox_root`.
   b. Remplacez la propriété traceFile par `equinox_root/logs/trace.log`.
2. Enregistrez le fichier.
3. Entrez les lignes de code suivantes dans la console OSGi pour créer la configuration du serveur à partir du fichier :

```
osgi> cm create com.ibm.websphere.xs.server

osgi> cm put com.ibm.websphere.xs.server
objectgrid.server.props
wxs_sample_osgi_root/server/properties/collocated.serve
r.properties
```

4. Pour afficher la configuration, exécutez la commande suivante :

```
osgi> cm get com.ibm.websphere.xs.server
Configuration for service (pid) "com.ibm.websphere.xs.server"
(bundle location = null)
key value
------ ------
objectgrid.server.props objectgrid.server.props
```

Point de contrôle de la leçon :

Au cours de cette leçon, vous avez édité le fichier `wxs_sample_osgi_root/server/properties/collocated.server.properties` pour spécifier les paramètres du serveur, telles que le répertoire de travail et l'emplacement des fichiers journaux de trace.

Leçon 2.3 : Configuration du conteneur eXtreme Scale

Suivez cette leçon pour configurer un conteneur qui inclut le fichier descripteur XML d'ObjectGrid et le fichier XML de déploiement d'ObjectGrid WebSphere eXtreme Scale. Ces fichiers incluent la configuration de la grille et sa topologie.

Pour créer un conteneur, commencez par créer un service de configuration à l'aide du PID (process identification number) de la fabrique de services gérés, com.ibm.websphere.xs.container. La configuration de service est une fabrique de services gérés qui permet de créer plusieurs PID de service depuis le PID de la fabrique. Pour démarrer le service de conteneur, affectez aux PID objectgridFile et deploymentPolicyFile chaque PID de service.

Procédez comme suit pour personnaliser et ajouter les propriétés du serveur à l'infrastructure OSGi :

1. Dans la console OSGI, entrez la commande suivante pour créer le conteneur depuis le fichier :

   ```
   osgi> cm createf com.ibm.websphere.xs.container
   PID: com.ibm.websphere.xs.container-1291179621421-0
   ```

2. Entrez la commande suivante pour lier le PID nouvellement créé aux fichiers XML ObjectGrid.

   A faire : Le numéro PID sera différent de celui de cet exemple.

   ```
   osgi> cm put com.ibm.websphere.xs.container-1291179621421-0
   objectgridFile wxs_sample_osgi_root/server/META-INF/protoBufObjectgrid.xml

   osgi> cm put com.ibm.websphere.xs.container-1291179621421-0
   deploymentPolicyFile wxs_sample_osgi_root/server/META-INF/protoBufDeployment.xml
   ```

3. Utilisez la commande suivante pour afficher la configuration :

```
osgi> cm get com.ibm.websphere.xs.container-1291760127968-0
Configuration for service (pid) "com.ibm.websphere.xs.container-1291760127968-0"
(bundle location = null)

key value
------                  ------
deploymentPolicyFile    /opt/wxs/ObjectGrid/samples/OSGiProto/server/META-INF/protoBufDeployment.xml
objectgridFile          /opt/wxs/ObjectGrid/samples/OSGiProto/server/META-INF/protoBufObjectgrid.xml
service.factoryPid    com.ibm.websphere.xs.container
service.pid        com.ibm.websphere.xs.container-1291760127968-0
```

Point de contrôle de la leçon :

Dans cette leçon, vous avez créé un service de configuration qui vous a permis de créer un conteneur eXtreme Scale. Comme les fichiers XML ObjectGrid contiennent la configuration de la grille et sa topologie, vous devez lier le conteneur que vous avez créé pour ces fichiers XML ObjectGrid. Avec cette configuration, le conteneur eXtreme Scale peut reconnaître les ensembles OSGi que vous allez exécuter ultérieurement dans ce tutoriel.

Leçon 2.4 : Installation de Google Protocol Buffers et des ensembles de plug-ins

Suivez ce tutoriel pour installer l'ensemble `protobuf-java-2.4.0a-bundle.jar` et l'ensemble de plug-ins `ProtoBufSamplePlugins-1.0.0.jar` en utilisant la console OSGi Equinox OSGi.

Procédez comme suit pour installer l'ensemble Google Protocol Buffers.

Dans la console OSGI, entrez la commande suivante pour installer l'ensemble :

```
osgi> install file:///wxs_sample_osgi_root/common/lib/com.google.protobuf_2.4.0a.jar
```

La sortie suivante s'affiche :

```
Bundle ID is 21
```

Présentation des exemples d'ensembles de plug-ins :

L'exemple OSGi inclut cinq exemples d'ensembles qui contiennent les plug-ins eXtreme Scale, notamment un plug-in personnalisé ObjectGridEventListener et un plug-in MapSerializerPlugin. Le plug-in MapSerializerPlugin utilise l'exemple Google Protocol Buffers et les messages fournis par l'exemple MapSerializerPlugin.

Les ensembles suivants se trouvent dans le répertoire *wxs_sample_osgi_root*`/lib` : `ProtoBufSamplePlugins-1.0.0.jar` et `ProtoBufSamplePlugins-2.0.0.jar`.

Le fichier blueprint.xml contient ce qui suit sans les commentaires :

```
<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0">
 <bean id="myShardListener" class="com.ibm.websphere.samples.xs.proto.osgi.MyShardListenerFactory"/>
 <service ref="myShardListener" interface="com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactory" ranking="1">
 </service>

 <bean id="myProtoBufSerializer" class="com.ibm.websphere.samples.xs.proto.osgi.ProtoMapSerializerFactory">
  <property name="keyType" value="com.ibm.websphere.samples.xs.serializer.app.proto.DataObjects1$OrderKey" />
  <property name="valueType" value="com.ibm.websphere.samples.xs.serializer.app.proto.DataObjects1$Order" />
 </bean>

 <service ref="myProtoBufSerializer" interface="com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactory"
  ranking="1">
 </service>
</blueprint>
```

Le fichier XML Blueprint exporte les deux services myShardListener et myProtoBufSerializer. Ces deux services sont référencés dans le fichier protoBufObjectgrid.xml.

Installation de l'ensemble de plug-ins :

Procédez comme suit pour installer l'ensemble ProtoBufSamplePlugins-1.0.0.jar.

Exécutez la commande suivante dans la console OSGi Equinox pour installer l'ensemble de plug-ins ProtoBufSamplePlugins-1.0.0.jar :

```
osgi> install file:///wxs_sample_osgi_root/common/lib/ProtoBufSamplePlugins-1.0.0.jar
```

La sortie suivante s'affiche :

```
Bundle ID is 22
```

Point de contrôle de la leçon :

Dans cette leçon, vous avez installé l'ensemble protobuf-java-2.4.0a-bundle.jar et l'ensemble de plug-ins ProtoBufSamplePlugins-1.0.0.jar.

Leçon 2.5 : Démarrage des ensembles OSGi

Le serveur WebSphere eXtreme Scale est modularisé comme ensemble de serveur OSGi. Suivez cette leçon pour installer l'ensemble de serveur eXtreme Scale et d'autres ensembles OSGi que vous avez installés.

1. Démarrer l'exemple d'ensemble de plug-in. Exécutez la commande suivante dans la console OSGi Equinox pour démarrer l'ensemble. Dans cet exemple, l'ID d'ensemble de l'exemple de plug-in est 22.

   ```
   osgi> start 22
   ```

2. Démarrer l'ensemble Google Protocol Buffers. Exécutez la commande suivante dans la console OSGi Equinox pour démarrer l'ensemble. Dans cet exemple, l'ID d'ensemble du plug-in Google Protocol Buffers est 21.

   ```
   osgi> start 21
   ```

3. Démarrer l'ensemble de serveur. Exécutez la commande suivante dans la console OSGi Equinox pour démarrer le serveur. Dans cet exemple, l'ID de l'ensemble de serveur eXtreme Scale est 19.

   ```
   osgi> start 19
   ```

Après avoir démarré le serveur, le programme d'écoute d'événement MyShardListener démarre et il est prêt à insérer ou mettre à jour les enregistrements. Vous pouvez visualiser la sortie suivante sur la console OSGi pour vérifier que l'ensemble de plug-in a démarré correctement :

```
SystemOut O MyShardListener@1253853884(version=1.0.0) order
com.ibm.websphere.samples.xs.serializer.proto.DataObjects1$Order$Builder
@1aba1aba(22) inserted
```

Point de contrôle de la leçon :

Dans cette leçon, vous avez démarré deux ensembles de plug-in et l'ensemble de serveur dans le conteneur eXtreme Scale que vous avez configuré pour l'infrastructure OSGi.

Module 3 : Exécution de l'exemple de client eXtreme Scale

Le serveur WebSphere eXtreme Scale fonctionne maintenant dans un environnement OSGi. Suivez les étapes de ce module pour exécuter un client WebSphere eXtreme Scale qui insère des données dans la grille.

Objectifs d'apprentissage

A la fin des leçons de ce module, vous saurez :

- exécuter une application client qui se connecte à la grille et y insère et en extrait des données ;
- démarrer une commande en utilisant l'application client non-OSGi.

Prerequis

Exécutez le Module 2 : Installation et démarrage des ensembles eXtreme Scale dans l'infrastructure OSGi.

Leçon 3.1 : Configuration d'Eclipse pour exécuter le client et créer les exemples

Effectuez cette leçon pour importer le projet Eclipse que vous utiliserez pour exécuter le client et générer les exemples de plug-ins.

L'exemple inclut un programme client Java SE qui se connecte à la grille, insère des données et en extrait. Il contient également des projets que vous pouvez utiliser pour générer et redéployer les ensembles OSGi.

Le projet fourni a été testé avec Eclipse 3.x et les versions suivantes et ne nécessite que la perspective du projet de développement Java. Procédez comme suit pour configurer votre environnement de développement WebSphere eXtreme Scale.

1. Ouvrez Eclipse dans un espace de travail nouveau ou existant.
2. Dans le menu Fichier, sélectionnez **Importer**.
3. Développez le dossier Général. Sélectionnez **Projets existants dans l'espace de travail** et cliquez sur **Suivant**.
4. Dans la zone **Sélectionner le répertoire racine**, tapez le répertoire *wxs_sample_osgi_root* ou accédez-y. Cliquez sur **Terminer**. Plusieurs nouveaux projets sont affichés dans votre espace de travail. Vous devez corriger plusieurs erreurs de génération en définissant la bibliothèque utilisateur eXtreme Scale. Procédez comme suit pour définir la bibliothèque utilisateur.
5. Dans le menu Fenêtre, sélectionnez **Préférences**.
6. Développez la branche **Java** > **Chemin de génération** et sélectionnez **Bibliothèques utilisateur**.
7. Cliquez sur **Nouveau**.
8. Tapez `eXtremeScale` dans la zone du **nom de bibliothèque utilisateur** et cliquez sur **OK**.

9. Sélectionnez la nouvelle bibliothèque utilisateur et cliquez **Ajouter des fichiers JAR**.
   a. Recherchez et sélectionnez le fichier `objectgrid.jar` dans le répertoire *wxs_install_root*`/lib`. Cliquez sur **OK**.
   b. Pour inclure la documentation d'API ObjectGrid, sélectionnez l'emplacement de la documentation d'API pour le fichier `objectgrid.jar` que vous avez ajouté à l'étape précédente. Cliquez sur **Editer**.
   c. Dans la zone Chemin d'emplacement de la documentation d'API, sélectionnez le fichier `Javadoc.zip` qui se trouve dans le répertoire *wxs_install_root*`/docs/javadoc.zip`.

Point de contrôle de la leçon :

Dans cette leçon, vous avez importé l'exemple de projet Eclipse, défini par la bibliothèque utilisateur eXtreme Scale, et inclus la documentation d'API correspondante pour l'exemple de projet. Maintenant, vous pouvez démarrer l'exemple d'application client.

Leçon 3.2 : Démarrage d'un client et insertion de données dans la grille

Etudiez cette leçon pour démarrer un client et exécuter une application client.

L'application client Java est `com.ibm.websphere.samples.xs.proto.client.Client`.

Ce client utilise une substitution de client, le fichier descripteur XML ObjectGrid pour remplacer la configuration OSGi, afin que le client puisse s'exécuter dans un environnement non-OSGi. Voir le contenu suivant du fichier avec les commentaires et les en-têtes supprimés. Certaines lignes de code sont réparties sur plusieurs lignes à des fins de formatage.

```
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">

    <objectGrids>
        <objectGrid name="Grid" txTimeout="15">
            <bean id="ObjectGridEventListener" className="" osgiService=""/>
            <backingMap name="Map" readOnly="false"
                lockStrategy="PESSIMISTIC" lockTimeout="5"
                copyMode="COPY_TO_BYTES" pluginCollectionRef="serializer"/>

        </objectGrid>
    </objectGrids>

    <backingMapPluginCollections>
        <backingMapPluginCollection id="serializer">

   <bean id="MapSerializer"
   className="com.ibm.websphere.samples.xs.serializer.proto.ProtoMapSerializer"
       osgiService="">
    <property name="keyType" type="java.lang.String"
        value="com.ibm.websphere.samples.xs.serializer.proto.DataObjects2$OrderKey" />
          <property name="valueType" type="java.lang.String"
              value="com.ibm.websphere.samples.xs.serializer.proto.DataObjects2$Order" />
   </bean>
  </backingMapPluginCollection>
    </backingMapPluginCollections>
</objectGridConfig>
```

Procédez comme suit pour démarrer l'application client.

1. Utilisez l'exemple de code suivant pour modifier les attributs de la classe Client pour refléter votre environnement.

```
private String catHost = "localhost";
private int catListenerPort = 2809;
private String clientOGXML = "wxs_sample_osgi_root/client/META-INF/
clientProtoBufObjectgrid.xml";
private String gridName = "Grid";
private String mapName = "Map";
```

2. Exécutez l'application client.

Lorsque vous exécutez l'application, le message suivant est affiché. Le message indique qu'une commande a été insérée :

```
order
com.ibm.websphere.samples.xs.serializer.proto.DataObjects1$Or
der$Builder@5d165d16(5000000) inserted
```

Point de contrôle de la leçon :

Dans cette leçon, vous avez démarré l'application `com.ibm.websphere.samples.xs.proto.client.Client` qui a généré une commande.

Module 4: Interrogation et mise à niveau de l'exemple d'ensemble

Suivez les leçons de ce module pour utiliser la commande **xscmd** pour interroger le classement de services de l'exemple d'ensemble, mettez-le à niveau vers un nouveau classement de services et vérifiez ce dernier.

Un projet Eclipse est fourni pour faciliter l'exécution des exemples d'applications.

Objectifs d'apprentissage

Après avoir suivi les leçons de ce module, vous saurez :

- interroger le classement de services en cours d'un service ;
- interroger le classement en cours de tous les services ;
- interroger tous les classements disponibles d'un service ;
- interroger tous les classements de services disponibles ;
- utiliser l'outil xscmd pour déterminer les classements de services disponibles ;
- mettre à jour les classements des exemples de services OSGi.

Prérequis

Exécutez le Module 3 : Exécution de l'exemple de client eXtreme Scale.

Leçon 4.1 : recherche des classements de services

Etudiez cette leçon pour identifier les classements de services en cours ainsi que les classements de services qui sont disponibles pour la mise à niveau.

- Identifiez le classement de service en cours d'un service. Entrez la commande suivante pour interroger le classement de service en cours utilisé pour le service myShardListener utilisée par la grille ObjectGrid `Grid` et le groupe de mappes `MapSet`.
  1. Accédez au répertoire suivant :
     ```
     cd wxs_home/bin
     ```
  2. Entrez la commande suivante pour interroger le classement en cours du service, myShardListener.
     ```
     ./xscmd.sh -c osgiCurrent -g Grid -ms MapSet -sn myShardListener
     ```

La sortie suivante s'affiche :

```
OSGi Service Name: myShardListener
ObjectGrid Name MapSet Name Server Name        Current Ranking
--------------- ----------- -----------        ---------------
Grid            MapSet      collocatedServer   1

CWXSI0040I: The command osgiCurrent has completed successfully.
```

- Identifiez le classement en cours de tous les services. Entrez la commande suivante pour interroger le classement de service en cours utilisé pour tous les services utilisé par la grille ObjectGrid `Grid` et le groupe de mappes `MapSet`.
  1. Accédez au répertoire suivant :

     ```
     cd wxs_home/bin
     ```

  2. Entrez la commande suivante pour interroger les classements de tous les services.

     ```
     ./xscmd.sh -c osgiCurrent -g Grid -ms MapSet
     ```

     La sortie suivante s'affiche :

```
OSGi Service Name    Current Ranking ObjectGrid Name MapSet Name Server Name
-----------------    --------------- --------------- ----------- -----------
myProtoBufSerializer 1               Grid            MapSet      collocatedServer
myShardListener      1               Grid            MapSet      collocatedServer

CWXSI0040I: The command osgiCurrent has completed successfully.
```

- Identifiez tous les classements disponibles d'un service. Entrez la commande suivante pour interroger les classements du service myShardListener.
  1. Accédez au répertoire suivant :

     ```
     cd wxs_home/bin
     ```

  2. Entrez la commande suivante pour interroger tous les classements d'un service.

     ```
     ./xscmd.sh -c osgiAll -sn myShardListener
     ```

     La sortie suivante s'affiche :

     ```
     Server: collocatedServer
        OSGi Service Name Available Rankings
        ----------------- ------------------
        myShardListener 1

        Summary - All servers have the same service rankings.

     CWXSI0040I: The command osgiAll has completed successfully.
     ```

     La sortie est regroupée en fonction du serveur. Dans cet exemple, seul le serveur collocatedServer existe.

- Identifiez tous les classements de services disponibles. Entrez la commande suivante pour rechercher tous les classements de tous les services.
  1. Accédez au répertoire suivant :

     ```
     cd wxs_home/bin
     ```

  2. Entrez la commande suivante pour rechercher tous les classements de services disponibles.

     ```
     ./xscmd.sh -c osgiAll
     ```

     La sortie suivante s'affiche :

     ```
     Server: collocatedServer
        OSGi Service Name    Available Rankings
        -----------------    ------------------
        myProtoBufSerializer 1
     ```

```
myShardListener 1

Summary - All servers have the same service rankings.
```

- Installez et démarrez la version 2 de l'ensemble de plug-in. Dans la console OSGi du serveur, installez un nouvel ensemble contenant une nouvelle version de la classe Order et le plug-in MapSerializerPlugin. Voir Leçon 2.4 : Installation de Google Protocol et exemples d'ensembles de plug-ins pour plus d'informations sur l'installation de l'ensemble `ProtoBufSamplePlugins-2.0.0.jar`.
  1. Après l'installation, démarrez le nouvel ensemble. Les services du nouvel ensemble sont disponibles, mais ils ne sont pas encore utilisés par le serveur eXtreme Scale. Vous devez exécuter une demande de mise à jour de service pour pouvoir utiliser un service avec une version donnée.
- Maintenant, lorsque vous interrogez de nouveau tous les classements de services disponibles, le classement de service 2 est ajouté dans la sortie.
  1. Accédez au répertoire suivant :
     ```
     cd wxs_home/bin
     ```
  2. Entrez la commande suivante pour rechercher tous les classements de services disponibles.
     ```
     ./xscmd.sh -c osgiAll
     ```

     La sortie suivante s'affiche :
     ```
     Server: collocatedServer
        OSGi Service Name    Available Rankings
        -----------------    ------------------
        myProtoBufSerializer 1, 2
        myShardListener      1, 2

     Summary - All servers have the same service rankings.
     ```

Point de contrôle de la leçon :

Dans ce tutoriel, vous avez interrogé des classements de services spécifiques et les classements de services disponibles. Vous avez également affiché le classement de service d'un nouvel ensemble que vous avez installé et démarré.

Leçon 4.2 : Déterminer si des classements de services spécifiques sont disponibles

Suivez cette leçon pour déterminer si des classements de services spécifiques sont disponibles pour les noms de service que vous spécifiez.

1. Entrez la commande suivante pour déterminer si le service myShardListener, avec le classement de service 2 et le service myProtoBufSerializer avec le classement de service 2 sont disponibles. La liste des classements de service est transmise à l'aide de l'option `-sr`.
   a. Accédez au répertoire suivant :
      ```
      cd wxs_home/bin
      ```
   b. Entrez la commande suivante pour déterminer si les services sont disponibles :
      ```
      ./xscmd.sh -c osgiCheck -g Grid -ms MapSet -sr
      "myShardListener;2,myProtoBufSerializer;2"
      ```

      La sortie suivante s'affiche :
      ```
      CWXSI0040I: The command osgiCheck has completed successfully.
      ```

2. Entrez la commande suivante pour déterminer si le service myShardListener, avec le classement de service 2 et le service myProtoBufSerializer avec le classement de service 3 sont disponibles.
   a. Accédez au répertoire suivant :
      ```
      cd wxs_home/bin
      ```
   b. Entrez la commande suivante pour déterminer si les services sont disponibles :
      ```
      ./xsadmin.sh -c osgiCheck -g Grid -ms MapSet -sr
      "myShardListener;2,myProtoBufSerializer;3"
      ```

      La sortie suivante s'affiche :
      ```
      Server OSGi Service Unavailable Rankings
      ------ ------------ --------------------
      collocatedServer myProtoBufSerializer 3
      ```

Point de contrôle de la leçon :

Dans cette leçon, vous avez spécifié les services myShardListener et myProtoBufSerializer, ainsi que des classements de services spécifiques pour déterminer si ces classements étaient disponibles.

Leçon 4.3 : Mise à jour des classements de services

Suivez cette leçon pour mettre à jour les classements de services en cours que vous avez interrogés.

1. La commande suivante met à jour les classements de services myShardListener et myProtoBufSerializer vers le classement de services 2. La liste des classements de services est envoyée en utilisant l'option `-sr`.
   a. Accédez au répertoire suivant :
      ```
      cd wxs_home/bin
      ```
   b. Entrez la commande suivante pour mettre à jour les classements de services :
      ```
      ./xscmd.sh -c osgiUpdate -g Grid -ms MapSet
      -sr "myShardListener;2,myProtoBufSerializer;2"
      ```

      La sortie suivante s'affiche :
      ```
      Update succeeded for the following service rankings:
      Service Ranking
      ------- -------
      myProtoBufSerializer 2
      myShardListener 2

      CWXSI0040I: The command osgiUpdate has completed successfully.
      ```

      La sortie suivante s'affiche sur la console OSGi :
      ```
      SystemOut O MyShardListener@326505334(version=2.0.0) order
      com.ibm.websphere.samples.xs.serializer.proto.DataObjects2$Order$Builder@
      22342234(34) updated
      ```

      Notez que le service MyShardListener est maintenant au niveau de version 2.0.0 qui a le classement de service 2.
2. Exécutez la commande **xscmd** pour interroger le classement de services en cours utilisé pour tous les services qui sont utilisés par l'ObjectGrid `Grid` et le groupe de mappes `MapSet`.
   a. Accédez au répertoire suivant :
      ```
      cd wxs_home/bin
      ```

b. Entrez la commande suivante pour interroger les classements de tous les services qui sont utilisés par `Grid` et `MapSet`:

```
./xscmd.sh -c osgiCurrent -g Grid -ms MapSet
```

La sortie suivante s'affiche :

```
OSGi Service Name Current Ranking ObjectGrid Name MapSet Name Server Name
----------------- --------------- --------------- ----------- -----------
myProtoBufSerializer 2 Grid MapSet collocatedServer
myShardListener 2 Grid MapSet collocatedServer

CWXSI0040I: The command osgiCurrent has completed successfully.
```

Point de contrôle de la leçon :

Dans cette leçon, vous avez mis à jour les classements des services myShardListener et myProtoBufSerializer.

Chapitre 4. Installation

WebSphere eXtreme Scale est une grille de données en mémoire que vous pouvez utiliser pour partitionner, répliquer et gérer dynamiquement des données d'application et une logique métier entre plusieurs serveurs. Une fois que vous avez déterminé les rôles et exigences de votre déploiement, installez eXtreme Scale sur votre système.

Avant de commencer

- Avant de commencer l'installation, vous devez connaître les architectures de mise en cache, l'intégration de cache et de base de données, la sérialisation, l'évolutivité et la disponibilité WebSphere eXtreme Scale. Voir Présentation du produit pour plus d'informations.
- Planifiez le déploiement WebSphere eXtreme Scale. Pour lus d'informations sur les différentes topologies de cache, le dimensionnement, etc., voir Chapitre 2, «Planification», à la page 9.
- Vérifiez que votre environnement remplit les conditions requises pour installer eXtreme Scale. Pour plus d'informations, voir «Configurations matérielle et logicielle requises», à la page 49.
- Pour plus d'informations sur les environnement et les autres conditions, voir «Planification pour l'installation», à la page 49.
- Si vous installez une mise à niveau sur une version précédente de WebSphere eXtreme Scale, suivez les étapes décrites dans «Mise à jour des serveurs eXtreme Scale», à la page 211.

Présentation de l'installation

Vous pouvez utiliser soit l'installation complète ou client pour installer WebSphere eXtreme Scale dans un environnement autonome ou WebSphere Application Server.

Types d'installations

Le programme d'installation complet et le programme distinct d'installation du client que vous pouvez télécharger à partir du site du support, offrent diverses options d'installation. Lorsque vous utilisez le programme d'installation complet, vous pouvez exécuter des serveurs de catalogue et des serveurs de conteneur. Sur les serveurs qui exécutent des applications client qui accèdent à la grille de données, vous pouvez utiliser une installation client uniquement. Utilisez l'installation serveur ou l'installation serveur et client sur des noeuds qui exécutent des serveurs de catalogue ou des serveurs de conteneurs.

- **Installation complète :**
  - Lorsque vous effectuez l'installation sur WebSphere Application Server, vous pouvez choisir d'installer le client uniquement ou le serveur et le client.
  - Lorsque vous effectuez l'installation dans un environnement autonome, vous pouvez installer à la fois le client et le serveur. Si vous souhaitez installer le client uniquement, utilisez l'installation WebSphere eXtreme Scale Client.
- **Installation client :**

  Vous pouvez utiliser l'installation client uniquement sur les noeuds qui exécutent les applications client. Pour installer le client uniquement, vous pouvez

télécharger le programme d'installation client uniquement pour la plateforme appropriée depuis la section des téléchargements sur le site Support.

Options d'environnement

Vous pouvez installer WebSphere eXtreme Scale dans un environnement autonome ou WebSphere Application Server.

- Environnement **WebSphere Application Server :**

  En installant WebSphere eXtreme Scale sur les noeuds de votre environnement WebSphere Application Server, vous pouvez démarrer automatiquement les serveurs de catalogue et les serveurs de conteneur dans la même cellule que votre gestionnaire de déploiement et d'autres serveurs d'applications.
- **Environnement autonome :**

  Dans une installation autonome, vous installez WebSphere eXtreme Scale dans un environnement ne disposant pas de WebSphere Application Server. Avec un environnement autonome, vous configurez et démarrez manuellement le serveur de catalogue et les processus serveur de conteneur.

Planification pour l'installation

Avant d'installer le produit, vous devez tenir compte de votre environnement.

Topologies d'installation

Avec WebSphere eXtreme Scale, vous pouvez créer des topologies d'installation qui contiennent des serveurs autonomes, WebSphere Application Server ou les deux. Les exemples suivants font partie des topologies possibles que vous pouvez créer.

Noeud de développement

Le scénario d'installation le plus simple consiste à créer un noeud de développement. Dans ce scénario, vous installez le client et le serveur WebSphere eXtreme Scale une fois sur le noeud sur lequel vous voulez développer l'application.

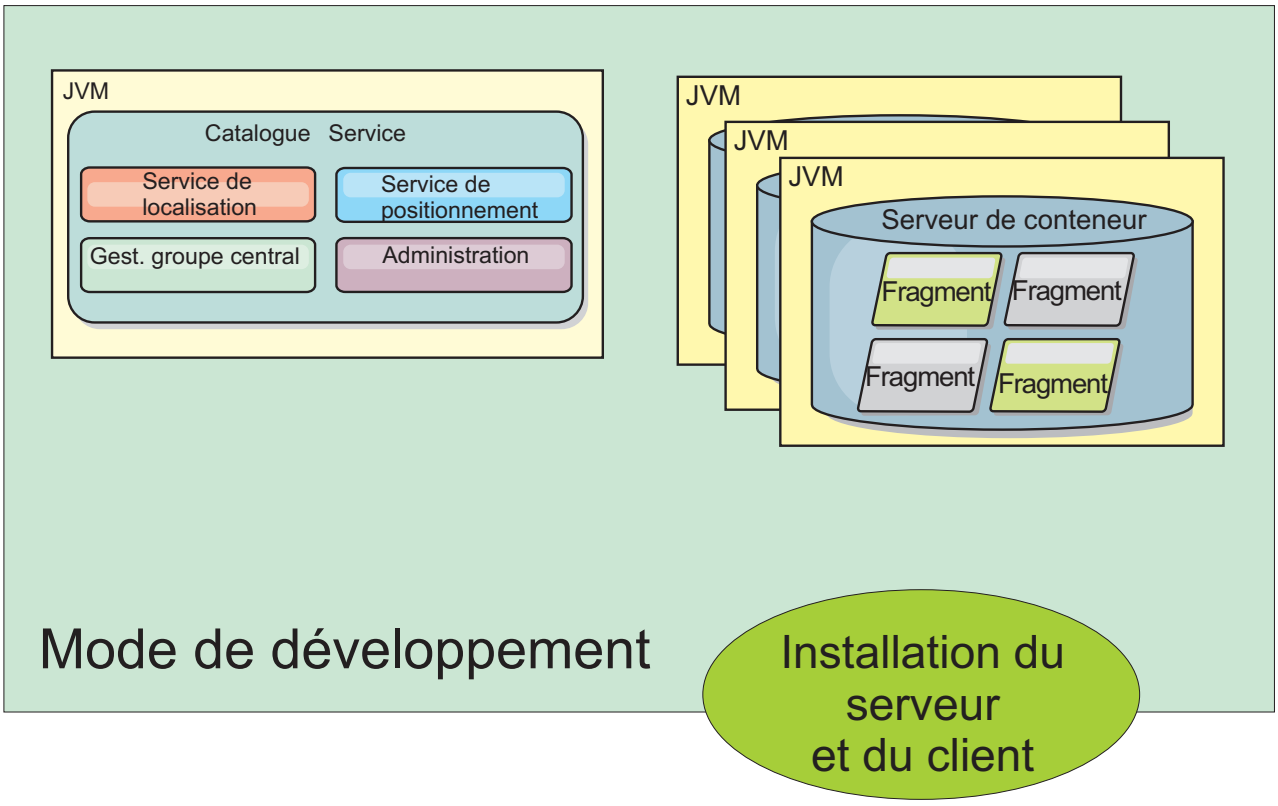


Figure 22. Noeud de développement

Après avoir effectué l'installation sur le noeud de développement, vous pouvez configurer l'environnement de développement et écrire vos applications.

Topologie autonome

Une topologie autonome est constituée de serveurs que vous n'exécutez pas sur WebSphere Application Server. Vous pouvez créez un grand nombre de topologies autonomes différentes, mais la topologie suivante est incluse comme exemple. Dans cette topologie, deux centres de données sont présents. Dans chaque centre de données, les installations complètes WebSphere eXtreme Scale (client et serveur) et les installations uniquement client sont installées sur les serveurs physiques. Les installations uniquement client se trouvent sur les noeuds qui exécutent les applications Web qui utilisent la grille de données. Ces noeuds n'exécutant pas de catalogue ou de serveurs de conteneur, il n'est pas nécessaire d'installer le serveur. Une liaison multimaître connecte deux domaines de services de catalogue dans la configuration. La liaison multimaître permet d'exécuter la réplication entre les fragments dans les serveurs de conteneur dans les différents centres de données.

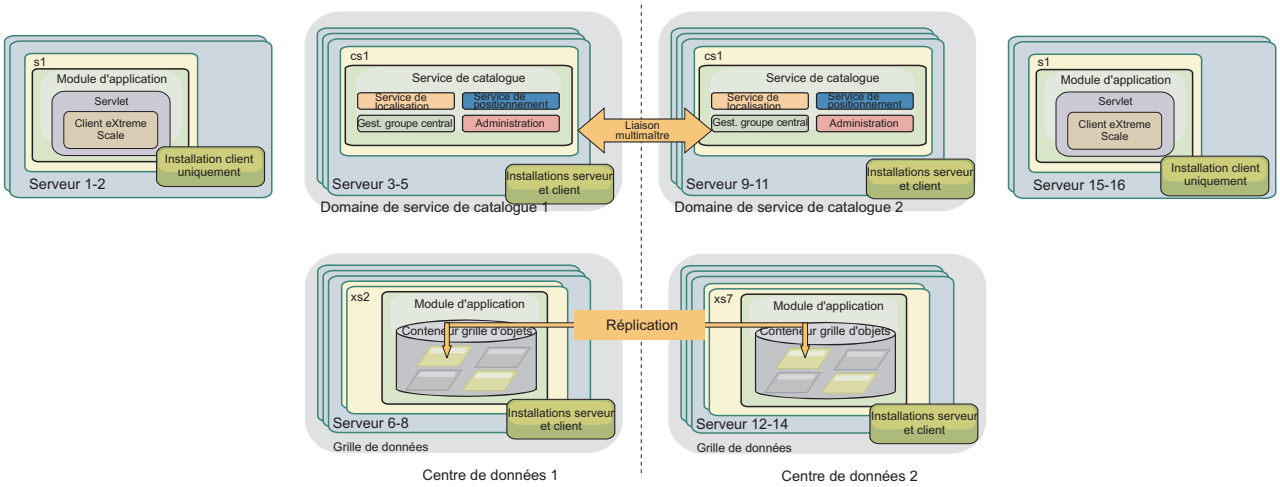


Figure 23. Topologie autonome avec deux centres de données

Avantages de l'utilisation d'une topologie autonome :

- Options d'intégration souple qui peuvent être intégrées aux infrastructure et bibliothèques des fournisseurs.
- Moindre encombrement qu'une topologie WebSphere Application Server.
- Moins d'exigences de licence qu'une topologie WebSphere Application Server.
- Options JRE (Expanded Java Runtime Environment).

Topologie WebSphere Application Server

Vous pouvez aussi créer une installation qui s'exécute entièrement dans une cellule WebSphere Application Server. Les clients, les serveurs de catalogue et les serveurs de conteneur ont chacun un cluster associé. Les noeuds qui exécutent l'application ont l'installation uniquement client. Les autres noeuds ont l'installation client et serveur.

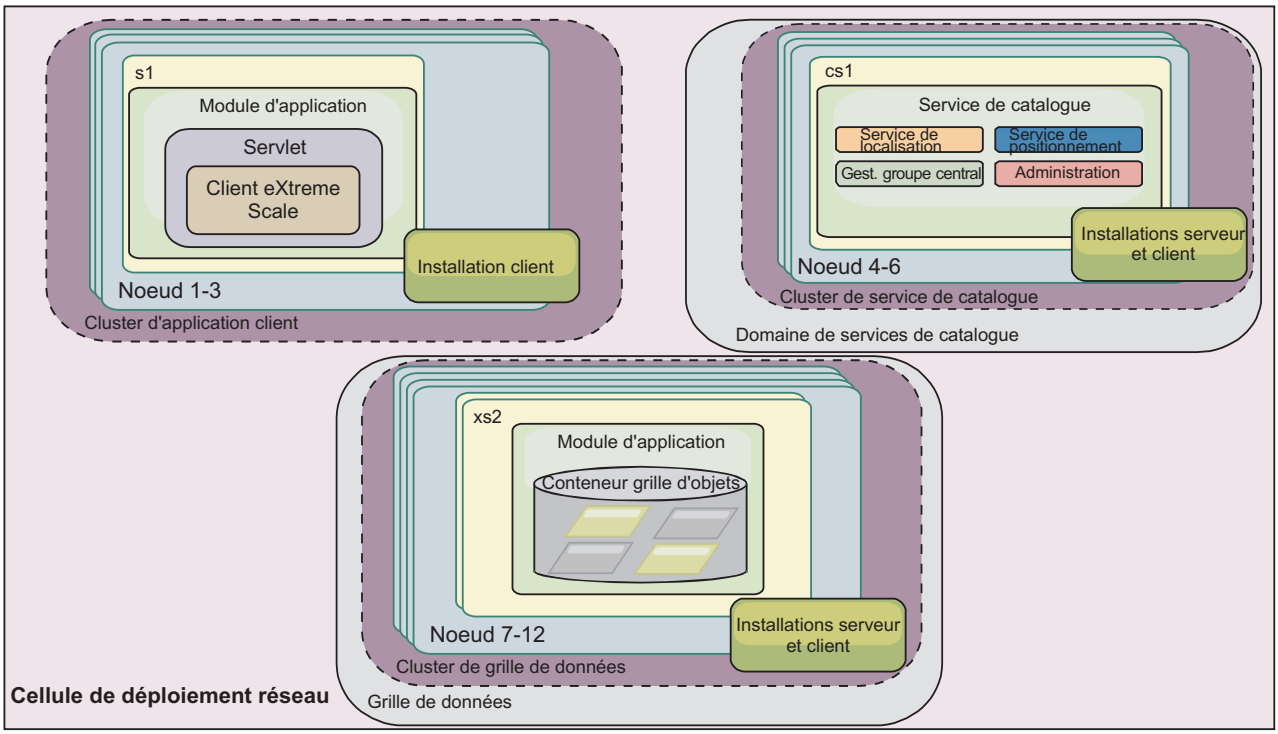


Figure 24. Exemple de topologie WebSphere Application Server

Avantages de la topologie WebSphere Application Server.

- Administration et configuration centralisées et cohérentes.
- Intégration de la sécurité.
- Intégration de l'application Java EE.
- Intégration PMI (Performance monitoring infrastructure).
- Intégration aux composants suivants WebSphere Application Server : cache OpenJPA L2, cache dynamique et persistance de session HTTP.

Topologie mixte

Vous pouvez créer une topologie mixte qui contient des serveurs WebSphere Application Server et autonomes. Dans l'exemple suivant, les applications client s'exécutent dans la cellule WebSphere Application Server, alors que les serveurs de catalogue et de conteneur s'exécutent en mode autonome.

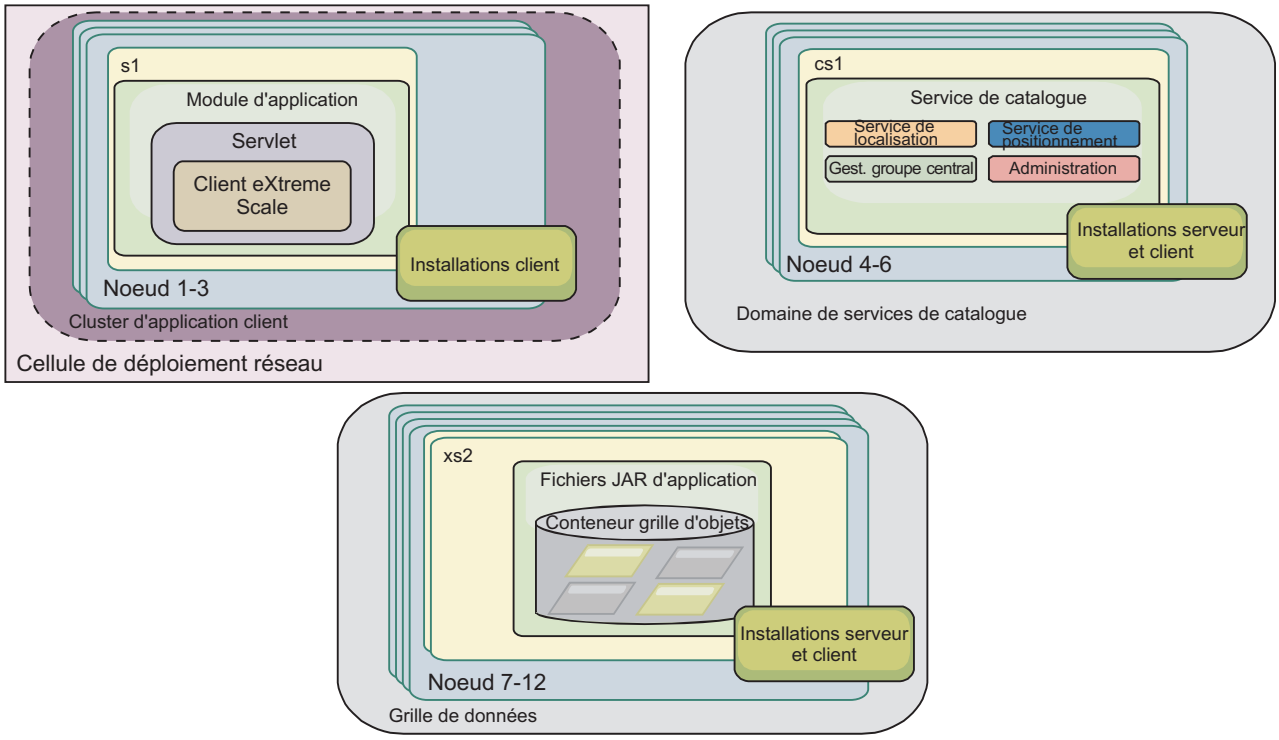


Figure 25. Exemple de topologie mixtes

Configurations matérielle et logicielle requises

Vue d'ensemble des conditions requises en termes de matériels et de systèmes d'exploitation. Bien que vous ne soyez pas tenu d'utiliser un niveau spécifique de matériel ou de système d'exploitation pour WebSphere eXtreme Scale, nous n'en fournissons pas moins sur le site de support du produit (page Configuration requise) une liste détaillée des matériels et logiciels officiellement pris en charge. En cas de conflit entre les informations présentées par le Centre de documentation et celles figurant sur cette page, les informations fournies par le site Web prévalent. Les conditions préalables répertoriées par le Centre de documentation sont fournies à titre informatif uniquement.

Voir la page System Configurations requises pour connaître les configurations matérielles et logicielles officielles.

Vous n'êtes pas obligé d'installer et de déployer eXtreme Scale sur un niveau de système d'exploitation spécifique. Chaque installation Java Platform, Standard Edition (Java SE) et Java Platform, Enterprise Edition (Java EE) requiert des niveaux de système d'exploitation ou des correctifs différents.

Vous pouvez installer et déployer le produit dans les environnements Java EE et Java SE. Vous pouvez également regrouper le composant client avec les applications Java EE directement sans les intégrer à WebSphere Application Server. WebSphere eXtreme Scale prend en charge Java SE 5 et les versions suivantes et WebSphere Application Server Version 6.1 et les versions suivantes.

Configuration matérielle

WebSphere eXtreme Scale ne requiert pas la présence d'un niveau spécifique de matériel. La configuration matérielle requise dépend du matériel pris en charge pour l'installation de Java Platform, Standard Edition que vous utilisez pour exécuter WebSphere eXtreme Scale. Si vous utilisez eXtreme Scale avec WebSphere Application Server ou une autre implémentation Java Platform, Enterprise Edition, la configuration matérielle requise par ces plateformes est suffisante pour WebSphere eXtreme Scale.

Configuration requise en matière de système d'exploitation

- **Sans la console Web**

  eXtreme Scale ne requiert pas la présence d'un système d'exploitation d'un niveau donné. Chaque implémentation Java SE et Java EE requiert un niveau différent du système d'exploitation ou des correctifs pour les problèmes identifiés lors du test de l'implémentation Java. Les niveaux nécessaires à ces implémentations sont suffisants pour eXtreme Scale.
- **Avec la console Web**

  Les conditions suivantes s'appliquent pour chaque système d'exploitation si vous utilisez la console :
  - Linux : JMV 32 bits ou 64 bits
  - Linux PPC : JVM 32 bits uniquement
  - Windows : JVM 32 bits uniquement
  - AIX : JVM 32 bits uniquement

Navigateurs Web requis

La console Web prend en charge les navigateurs Web suivants :

- Mozilla Firefox, version 3.5.x et versions ultérieures
- Mozilla Firefox, version 3.6.x et versions ultérieures
- Microsoft Internet Explorer version 7 ou 8

Configuration requise pour WebSphere Application Server

- WebSphere Application Server Version 6.1.0.39 ou version suivante
- WebSphere Application Server Version 7.0.0.19 ou version suivante
- WebSphere Application Server Version 8.0.0.1 ou version suivante

Pour plus d'informations, consultez la section Recommended fixes for WebSphere Application Server.

Autres conditions requises par le serveur d'applications

Les autres implémentations Java EE peuvent utiliser la phase d'exécution d'eXtreme Scale en tant qu'instance locale ou client pour les serveurs eXtreme Scale. Pour implémenter Java SE, vous devez utiliser la version 5 ou une version suivante.

Java SE : points à prendre en considération

WebSphere eXtreme Scale nécessite Java SE 5 ou une version suivante. En règle générale, les nouvelles versions de Java SE ont des fonctions plus efficaces et sont plus performantes.

Versions prises en charge

Vous pouvez utiliser WebSphere eXtreme Scale avec Java SE 5 et les versions suivantes. La version que vous utilisez doit être actuellement pris en charge par le fournisseur de l'environnement JRE (Java Runtime Environmen).

Un environnement JRE entièrement pris en charge est installé avec les installations autonomes WebSphere eXtreme Scale et WebSphere eXtreme Scale Client dans le répertoire *racine_install_wxs*/java et peut être utilisé par les clients et les serveurs. Si vous installez WebSphere eXtreme Scale dans WebSphere Application Server, vous pouvez utiliser l'environnement JRE inclus dans l'installation WebSphere Application Server.

WebSphere eXtreme Scale tire parti de la fonctionnalité Java Development Kit (JDK) 5 ou d'une version suivante lorsqu'elle devient disponible. Généralement, les nouvelles versions Java Development Kit (JDK) et Java SE sont plus performantes et ont une fonctionnalité plus efficace.

Voir Logiciels pris en charge pour plus d'informations.

Fonctions WebSphere eXtreme Scale dépendantes de Java

Tableau 4. Fonctions nécessitant Java SE 5 ou Java SE 6.

WebSphere eXtreme Scale utilise la fonctionnalité introduite dans Java SE 5 ou Java SE 6 pour fournir les fonctions suivantes du produit.

Caractéristique	Pris en charge dans Java SE 5 et les versions suivantes	Pris en charge dans Java SE 6 et les versions suivantes
Annotations d'API EntityManager (facultatif : vous pouvez également utiliser des fichiers XML)	X	X
Java Persistence API (JPA) : programme de chargement JPA, programme de chargement de client JPA et programme de mise à jour en fonction du temps JPA	X	X
L'expulsion basée sur la mémoire (utilise MemoryPoolMXBean)	X	X
Agents d'instrumentation : • `wxssizeagent.jar` : augmente la précision des mesures de mappe d'octets utilisées. • `ogagent.jar` : augmente la performance des entités d'accès aux zones.	X	X
Console Web de surveillance		X

Java EE : points à prendre en considération

Lors de la préparation de l'intégration WebSphere eXtreme Scale dans un environnement Java Platform, Enterprise Edition, tenez compte de certains

éléments, tels que les versions, les options de configuration, les conditions requises et les limitations, le déploiement et la gestion des applications.

Exécuter des applications eXtreme Scale en environnement Java EE

Une application Java EE peut se connecter à une application eXtreme Scale distante. En outre, l'environnement WebSphere Application Server permet le démarrage d'un serveur eXtreme Scale lorsqu'une application démarre dans le serveur d'applications.

Si vous utilisez un fichier XML pour créer une instance ObjectGrid et que ce fichier XML se trouve dans le module du fichier EAR, accédez à ce fichier à l'aide de la méthode getClass().getClassLoader().getResource("META-INF/objGrid.xml") afin d'obtenir un objet URL permettant de créer une instance ObjectGrid. Dans l'appel à la méthode, remplacez le nom du fichier XML utilisé.

Vous pouvez utiliser des beans de démarrage pour que, à son démarrage, une application amorce une instance ObjectGrid et supprime cette instance lorsqu'elle s'arrête. Un bean de démarrage est un bean de session sans état avec un emplacement distant com.ibm.websphere.startupservice.AppStartUpHome et une interface distante com.ibm.websphere.startupservice.AppStartUp. L'interface distante possède deux méthodes : la méthode start et la méthode stop. Utilisez la méthode start pour amorcer l'instance et la méthode stop pour détruire l'instance. L'application utilise la méthode ObjectGridManager.getObjectGrid pour maintenir la référence à cette instance. Voir les informations relatives à l'accès à un objet ObjectGrid avec ObjectGridManager dans *Guide de programmation* pour plus d'informations.

Utiliser des loaders de classes

Lorsque les modules d'application qui utilisent des chargeurs de classe différents partagent une instance ObjectGrid unique dans une application Java EE, vérifiez que les objets qui sont stockés dans eXtreme Scale et que les plug-ins du produit se trouvent dans un chargeur commun dans l'application.

Gérer dans un servlet le cycle de vie des instances ObjectGrid

Pour gérer le cycle de vie d'une instance ObjectGrid dans un servlet, vous pouvez utiliser la méthode init pour créer l'instance et la méthode destroy pour supprimer l'instance. Si l'instance est mise en cache, elle est extraite et manipulée dans le code du servlet. Voir les informations relatives à l'accès à un objet ObjectGrid avec l'interface ObjectGridManager dans *Guide de programmation* pour plus d'informations.

Conventions relatives aux répertoires

Les conventions de répertoire suivantes sont utilisées dans toute la documentation pour faire référence à des répertoires spéciaux, tels que *wxs_install_root* et *wxs_home*. Vous pouvez accéder à ces répertoires pendant plusieurs scénarios différents, y compris lors de l'installation et de l'utilisation des outils de ligne de commande.

racine_install_wxs
Le répertoire *wxs_install_root* est le répertoire racine où sont installés les fichiers du produit WebSphere eXtreme Scale. Le répertoire *wxs_install_root*

peut être le répertoire dans lequel l'archive d'évaluation est extraite ou depuis lequel le produit est installé WebSphere eXtreme Scale.

- Exemple où la version d'essai a été extraite :

  Exemple : `/opt/IBM/WebSphere/eXtremeScale`
- Exemple où WebSphere eXtreme Scale est installé dans un répertoire autonome :

  Exemple : `/opt/IBM/eXtremeScale`
- Exemple lorsque WebSphere eXtreme Scale est intégré à WebSphere Application Server :

  Exemple : `/opt/IBM/WebSphere/AppServer`

wxs_home
Le répertoire *wxs_home* est le répertoire racine du produit, des bibliothèques, des exemples et des composants WebSphere eXtreme Scale. Ce répertoire est identique au répertoire *wxs_install_root* lorsque l'archive d'évaluation est extraite. Pour les installations autonomes, le répertoire *wxs_home* est le sous-répertoire `ObjectGrid` du répertoire *wxs_install_root*. Pour les installations qui sont intégrées à WebSphere Application Server, ce répertoire est le répertoire `optionalLibraries/ObjectGrid` du répertoire *wxs_install_root*.

- Exemple lorsque la version d'essai a été extraite :

  Exemple : `/opt/IBM/WebSphere/eXtremeScale`
- Exemple lorsque WebSphere eXtreme Scale est installé dans un répertoire autonome :

  Exemple : `/opt/IBM/eXtremeScale/ObjectGrid`
- Exemple lorsque WebSphere eXtreme Scale est intégré à WebSphere Application Server :

  Exemple : `/opt/IBM/WebSphere/AppServer/optionalLibraries/ObjectGrid`

was_root
Le répertoire *was_root* est le répertoire racine d'une installation WebSphere Application Server :

Exemple : `/opt/IBM/WebSphere/AppServer`

restservice_home
Le répertoire *restservice_home* est le répertoire dans lequel se trouvent les bibliothèques et les exemples du service de données REST d'WebSphere eXtreme Scale. Ce répertoire s'appelle `restservice` et il est le sous-répertoire de *wxs_home*.

- Exemple pour les déploiements autonomes :

  Exemple : `/opt/IBM/WebSphere/eXtremeScale/ObjectGrid/restservice`
- Exemple pour les déploiements intégrés à WebSphere Application Server :

  Exemple : `/opt/IBM/WebSphere/AppServer/optionalLibraries/ObjectGrid/restservice`

tomcat_root
Le répertoire *home_tomcat* est le répertoire racine de l'installation d'Apache Tomcat.

Exemple : `/opt/tomcat5.5`

wasce_root
wasce_root est le répertoire racine de l'installation WebSphere Application Server Community Edition.

Exemple : `/opt/IBM/WebSphere/AppServerCE`

java_home
Le répertoire *java_home* est le répertoire racine d'une installation de Java Runtime Environment Kit (JRE).

Exemple : `/opt/IBM/WebSphere/eXtremeScale/java`

samples_home
samples_home est le répertoire dans lequel vous extrayez les exemples de fichiers qui sont utilisés pour les tutoriels.

Exemple : `/wxs-samples/`

dvd_root
dvd_root est le répertoire racine du DVD qui contient le produit.

Exemple : `dvd_root/docs/`

equinox_root
Le répertoire *equinox_root* est le répertoire racine de l'installation de l'infrastructure OSGi Eclipse Equinox.

Exemple :`/opt/equinox`

user_home
Le répertoire *user_home* est l'emplacement de stockage des fichiers utilisateur, tels que les profils de sécurité.

Windows `c:\Documents and Settings\user_name`

UNIX `/home/user_name`

Installation de WebSphere eXtreme Scale avec l'assistant d'installation

Vous pouvez utiliser l'assistant d'installation pour installer WebSphere eXtreme Scale pour les configurations autonomes ou WebSphere Application Server.

Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client avec WebSphere Application Server

Vous pouvez installer WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client dans un environnement dans lequel WebSphere Application Server ou WebSphere Application Server Network Deployment est installé. Vous pouvez utiliser les fonctions existantes de WebSphere Application Server ou WebSphere Application Server Network Deployment pour améliorer vos applications eXtreme Scale.

Avant de commencer

- Installez WebSphere Application Server ou WebSphere Application Server Network Deployment. Pour plus d'informations, voir Installation de l'environnement de traitement des applications.
- En fonction de la version que vous installez, 6.1 ou 7.0, appliquez le dernier correctif de WebSphere Application Server ou WebSphere Application Server Network Deployment pour mettre à jour le niveau du produit. Pour plus d'informations, reportez-vous aux groupes de correctifs les plus récents de WebSphere Application Server.
- Vérifiez que le répertoire d'installation cible ne contient pas une installation de WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client.

- Arrêtez tous les processus en cours d'exécution dans votre environnement WebSphere Application Server ou WebSphere Application Server Network Deployment. Voir Utilitaires de ligne de commande pour plus d'informations sur les commandes **stopManager**, **stopNode** et **stopServer**.

  ATTENTION :
  Vérifiez que les processus en cours d'exécution sont arrêtés. Si les processus en cours d'exécution ne sont pas arrêtés, l'installation se poursuit et crée des résultats imprévisibles, ce qui la laisse dans un état indéterminé sur certaines plateformes.
- Si vous n'installez que le client, vous pouvez utiliser le DVD pour installer le client ou télécharger WebSphere eXtreme Scale Client pour la plateforme spécifique depuis la section des téléchargements sur le site Support .

Important : Lorsque vous installez WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client, il doit se trouver dans le répertoire dans lequel vous avez installé WebSphere Application Server. Par exemple, si vous avez installé WebSphere Application Server dans `C:\`*racine_was*, choisissez également `C:`*racine_was* comme répertoire cible de l'installation WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client.

Pourquoi et quand exécuter cette tâche

Intégrez eXtreme Scale à WebSphere Application Server ou WebSphere Application Server Network Deployment pour appliquer les fonctions d'eXtreme Scale à vos applications Java Platform, Enterprise Edition. Les applications Java EE hébergent les grilles de données et y accèdent à l'aide d'une connexion client.

Procédure

1. Utilisez l'assistant pour effectuer l'installation.
   - Exécutez le script suivant pour démarrer l'assistant de l'installation intégrale de WebSphere eXtreme Scale. Vous pouvez décidez de n'installer que le client ou d'installer le serveur et le client :
     - Linux UNIX *dvd_root*`/install`
     - Windows *dvd_root*`\install.bat`
   - Exécutez le script suivant pour démarrer l'assistant de l'installation WebSphere eXtreme Scale Client. Les fichiers d'installation se trouvent dans le fichier zip que vous téléchargez depuis la section des téléchargements sur le site Support :
     - Linux UNIX `root/WXS_Client/install`
     - Windows `root\WXS_Client\install.bat`

   Avertissement : Si vous utilisez les conventions d'attribution de nom uniforme (UNC) pour identifier les chemins des fichiers dans la commande d'installation, les éléments que vous envisagez d'installer peuvent ne pas être installés à la fin de l'exécution de la commande. Pour éviter ce problème, mappez le chemin des fichiers à une unité de réseau. Exécutez la commande **install** sur l'unité mappée. L'utilisation d'une unité réseau mappée permet d'installer tous les éléments.
2. Suivez les invites de l'assistant.

   Le panneau des fonctions facultatives répertorie les fonctions que vous pouvez choisir d'installer. Toutefois, des fonctions ne peuvent pas être ajoutées de manière incrémentielle à l'environnement du produit une fois que le produit a

été installé. Si vous choisissez de ne pas installer une fonction lors de l'installation initiale du produit, vous devez désinstaller, puis réinstaller le produit pour l'ajouter.

Le panneau Extension de profil répertorie les profils existants que vous pouvez sélectionner pour étendre les fonctions d'eXtreme Scale. Si vous sélectionnez des profils existants déjà en cours d'utilisation, toutefois, un panneau d'avertissement est affiché. Pour poursuivre l'installation, arrêtez les serveurs configurés dans les profils ou cliquez sur **Précédent** pour supprimer les profils de votre sélection.

Résultats

Windows Si vous installez WebSphere eXtreme Scale Client sur Windows, le texte suivant peut s'afficher dans les résultats de l'installation :

```
Success: The installation of the following product was successful:
WebSphere eXtreme Scale Client.  Some configuration steps have errors.
For more information, refer to the following log file:
<WebSphere Application Server install root>\logs\wxs_client\install\log.txt"
Review the installation log (log.txt) and review the deployment manager
augmentation log.
```

S'il existe une erreur associée au fichier `iscdeploy.sh`, vous pouvez l'ignorez. Cette erreur ne pose pas de problème.

Que faire ensuite

- Si vous exécutez WebSphere Application Server Version 6.1 ou 7.0, vous pouvez utiliser le plug-in de l'outil de gestion de profil ou la commande **manageprofiles**. Pour plus d'informations, voir «Création et augmentation de profils pour WebSphere eXtreme Scale», à la page 184.
- Vérifiez l'installation. Pour plus d'informations, voir «Vérification de l'installation», à la page 206.
- Commencez à configurer votre installation WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client. Pour plus d'informations, voir «Premières étapes après l'installation», à la page 208.

Fichiers d'exécution pour WebSphere eXtreme Scale intégré à WebSphere Application Server

Des fichiers archives Java (JAR) sont inclus dans l'installation. Vous pouvez voir les fichiers JAR qui sont inclus et l'emplacement dans lequel ils sont installés.

Tableau 5. Fichiers d'exécution pour WebSphere eXtreme Scale. Le tableau ci-après répertorie les fichiers JAR (archives Java) inclus dans l'installation. L'emplacement d'installation est relatif au répertoire *rép_base_wxs* que vous choisissez lors de l'installation.

Nom de fichier	Environnement	Emplacement de l'installation	Description
`wxsdynacache.jar`	Client et serveur	`lib`	Le fichier `wxsdynacache.jar` contient les classes nécessaires à utiliser avec le fournisseur de cache dynamique.
`wsobjectgrid.jar`	Local et client	`lib`	Le fichier `wsobjectgrid.jar` contient les exécutions local, client et serveur eXtreme Scale.
`ogagent.jar`	Local, client et serveur	`lib`	Le fichier `ogagent.jar` contient les classes d'exécution requises pour exécuter l'agent d'instrumentation Java utilisé avec l'API EntityManager.
`ogsip.jar`	Serveur	`lib`	Le fichier `ogsip.jar` contient l'exécution de gestion de sessions SIP (Session Initiation Protocol eXtreme Scale) qui est compatible avec WebSphere Application Server Version 6.1.x.
`sessionobjectgrid.jar`	Client et serveur	`lib`	Le fichier `sessionobjectgrid.jar` contient l'exécution de la gestion de sessions HTTP eXtreme Scale.

Tableau 5. Fichiers d'exécution pour WebSphere eXtreme Scale (suite). Le tableau ci-après répertorie les fichiers JAR (archives Java) inclus dans l'installation. L'emplacement d'installation est relatif au répertoire *rép_base_wxs* que vous choisissez lors de l'installation.

Nom de fichier	Environnement	Emplacement de l'installation	Description
sessionobjectgridsip.jar	Serveur	lib	Le fichier sessionobjectgridsip.jar contient l'exécution de gestion de sessions SIP eXtreme Scale compatible avec WebSphere Application Server Version 7.x.
wsogclient.jar	Local et client	lib	Fichier wsogclient.jar installé si vous utilisez un environnement qui contient WebSphere Application Server Version 6.0.2 et versions ultérieures. Ce fichier ne contient que les environnements d'exécution local et client.
wxssizeagent.jar	Local, client et serveur	lib	Le fichier wxssizeagent.jar est utilisé pour fournir des informations de dimensionnement d'entrée de cache plus précises lors de l'utilisation de l'environnement Java (JRE) Version 1.5 ou versions suivantes.
oghibernate-cache.jar	Client et serveur	optionalLibraries/ObjectGrid	Le fichier oghibernate-cache.jar contient le plug-in de cache L2eXtreme Scale pour JBoss Hibernate.
ogspring.jar	Local, client et serveur	optionalLibraries/ObjectGrid	Le fichier ogspring.jar contient les classes de support pour l'intégration de l'infrastructure SpringSource Spring.
xsadmin.jar	Utilitaire	optionalLibraries/ObjectGrid	Le fichier xsadmin.jar contient l'exemple d'utilitaire eXtreme Scale.
ibmcfw.jar ibmorb.jar ibmorbapi.jar	Client et serveur	optionalLibraries/ObjectGrid/endorsed	Cet ensemble de fichiers comprend l'exécution ORB (Object Request Broker) utilisée pour exécuter les applications dans les processus Java SE.
wxshyperic.jar	Utilitaire	optionalLibraries/ObjectGrid/hyperic/lib	Le plug-in de détection des serveurs WebSphere eXtreme Scale pour l'agent de surveillance SpringSource Hyperic.
restservice.ear	Client	optionalLibraries/ObjectGrid/restservice/lib	Le fichier restservice.ear contient l'archive d'application d'entreprise de service de données eXtreme Scale REST pour les environnements WebSphere Application Server.
restservice.war	Client	optionalLibraries/ObjectGrid/restservice/lib	Le fichier restservice.war contient l'archive Web de service de données eXtreme Scale REST pour les serveurs d'applications provenant d'un autre fournisseur.
splicerlistener.jar	Utilitaire	optionalLibraries/ObjectGrid/session/lib	Le fichier splicerlistener.jar contient l'utilitaire splicer pour le filtre de gestionnaire de sessions HTTP eXtreme Scale.
splicer.jar	Utilitaire	optionalLibraries/ObjectGrid/legacy/session/lib	Le fichier splicer.jar contient l'utilitaire splicer Version 7.0 pour le filtre de gestionnaire de sessions HTTP eXtreme Scale.

Tableau 6. Fichiers d'exécution pour WebSphere eXtreme Scale Client. Le tableau ci-après répertorie les fichiers JAR (archives Java) inclus dans l'installation. L'emplacement d'installation est relatif au répertoire *rép_base_wxs* que vous choisissez lors de l'installation.

Nom de fichier	Environnement	Emplacement de l'installation	Description
wxsdynacache.jar	Client et serveur	lib	Le fichier wxsdynacache.jar contient les classes nécessaires à utiliser avec le fournisseur de cache dynamique.
ogagent.jar	Local, client et serveur	lib	Le fichier ogagent.jar contient les classes d'exécution requises pour exécuter l'agent d'instrumentation Java utilisé avec l'API EntityManager.
ogsip.jar	Serveur	lib	Le fichier ogsip.jar contient l'exécution de gestion de sessions SIP (Session Initiation Protocol eXtreme Scale) qui est compatible avec WebSphere Application Server Version 6.1.x.
sessionobjectgrid.jar	Client et serveur	lib	Le fichier sessionobjectgrid.jar contient l'exécution de la gestion de sessions HTTP eXtreme Scale.

Tableau 6. Fichiers d'exécution pour WebSphere eXtreme Scale Client (suite). Le tableau ci-après répertorie les fichiers JAR (archives Java) inclus dans l'installation. L'emplacement d'installation est relatif au répertoire *rép_base_wxs* que vous choisissez lors de l'installation.

Nom de fichier	Environnement	Emplacement de l'installation	Description
sessionobjectgridsip.jar	Serveur	lib	Le fichier sessionobjectgridsip.jar contient l'exécution de gestion de sessions SIP eXtreme Scale compatible avec WebSphere Application Server Version 7.x.
wsogclient.jar	Local et client	lib	Fichier wsogclient.jar installé si vous utilisez un environnement qui contient WebSphere Application Server Version 6.0.2 et versions ultérieures. Ce fichier ne contient que les environnements d'exécution local et client.
wxssizeagent.jar	Local, client et serveur	lib	Le fichier wxssizeagent.jar est utilisé pour fournir des informations de dimensionnement d'entrée de cache plus précises lors de l'utilisation de l'environnement d'exécution Java (JRE) Version 1.5 ou version suivante.
oghibernate-cache.jar	Client et serveur	optionalLibraries/ObjectGrid	Le fichier oghibernate-cache.jar contient le plug-in de cache L2 eXtreme Scale pour JBoss Hibernate.
ogspring.jar	Local, client et serveur	optionalLibraries/ObjectGrid	Le fichier ogspring.jar contient les classes de support pour l'intégration de l'infrastructure SpringSource Spring.
xsadmin.jar	Utilitaire	optionalLibraries/ObjectGrid	Le fichier xsadmin.jar contient l'exemple d'utilitaire eXtreme Scale.
ibmcfw.jar ibmorb.jar ibmorbapi.jar	Client et serveur	optionalLibraries/ObjectGrid/endorsed	Cet ensemble de fichiers comprend l'exécution ORB (Object Request Broker) utilisée pour exécuter les applications dans les processus Java SE.
wxshyperic.jar	Utilitaire	optionalLibraries/ObjectGrid/hyperic/lib	Le plug-in de détection des serveurs WebSphere eXtreme Scale pour l'agent de surveillance SpringSource Hyperic.
restservice.ear	Client	optionalLibraries/ObjectGrid/restservice/lib	Le fichier restservice.ear contient l'archive d'application d'entreprise de service de données eXtreme Scale REST pour les environnements WebSphere Application Server.
restservice.war	Client	optionalLibraries/ObjectGrid/restservice/lib	Le fichier restservice.war contient l'archive Web de service de données eXtreme Scale REST pour les serveurs d'applications provenant d'un autre fournisseur.
splicerlistener.jar	Utilitaire	optionalLibraries/ObjectGrid/session/lib	Le fichier splicerlistener.jar contient l'utilitaire splicer pour le filtre de gestionnaire de sessions HTTP eXtreme Scale.
splicer.jar	Utilitaire	optionalLibraries/ObjectGrid/legacy/session/lib	Le fichier splicer.jar contient l'utilitaire splicer Version 7.0 pour le filtre de gestionnaire de sessions HTTP eXtreme Scale.

Utilisation du plug-in Installation Factory pour créer et installer des modules personnalisés

Utilisez le plug-in IBM Installation Factory pour que WebSphere eXtreme Scale crée un module d'installation personnalisée (CIP) ou un module d'installation intégrée (IIP). Un module d'installation personnalisée contient un unique module d'installation du produit et diverses ressources facultatives. Un module d'installation intégrée associe un ou plusieurs modules d'installation dans un même flux de travaux d'installation que vous concevez.

Avant de commencer

Avant de créer et d'installer des modules personnalisés pour eXtreme Scale, vous devez télécharger les produits suivants :

- IBM® Installation Factory for WebSphere Application Server
- Plug-in IBM Installation Factory pour WebSphere eXtreme Scale

Pourquoi et quand exécuter cette tâche

A l'aide d'Installation Factory, vous pouvez créer un module CIP en associant un composant de produit à des modules de maintenance, des scripts de personnalisation et d'autres fichiers. Lorsque vous créez un module IIP, vous regrouper les composants individuels ou les modules d'installation dans un même module d'installation.

Fichier de définition de génération :

Un fichier de définition de génération est un document XML qui spécifie comment créer et installer un module d'installation personnalisée (CIP) ou un module d'installation intégrée (IIP). IBM Installation Factory for WebSphere eXtreme Scale lit les détails relatifs au module dans le fichier de définition de génération pour générer un module CIP ou IIP.

Pour pouvoir créer un module CIP ou IIP, vous devez créer un fichier de définition de génération pour chaque module personnalisé. Le fichier de définition de génération décrit les composants du produit ou les modules d'installation à installer, l'emplacement du module CIP ou IIP, les modules de maintenance à inclure, les scripts installation et les autres fichiers à intégrer. Vous pouvez également spécifier dans le fichier de définition de génération du module IIP l'ordre suivant lequel Installation Factory installe chaque module d'installation.

L'assistant de définition de génération vous guide à travers les étapes de création d'un fichier de définition de génération. Vous pouvez aussi utiliser l'assistant pour modifier un fichier de définition de génération existant. Chaque page de l'assistant vous invite à entrer des informations relatives à un module personnalisé telles que l'identification du module, l'emplacement d'installation de la définition de génération et l'emplacement d'installation du module personnalisé. Toutes ces informations sont enregistrées dans le nouveau fichier de définition de génération ou modifiées et enregistrées dans un fichier de définition de génération existant. Pour plus d'informations, consultez les sections Pages de l'assistant de définition de génération du module CIP et Pages de l'assistant de définition de génération du module IIP.

Pour créer le fichier de définition de génération uniquement, vous pouvez utiliser l'outil de l'interface de ligne de commande pour générer le module personnalisé en dehors de l'interface graphique. Pour plus d'informations, voir «Installation en mode silencieux d'un module CIP ou IIP», à la page 175.

Création d'un fichier de définition de génération et génération d'un module d'installation personnalisée (CIP) :

Le plug-in IBM Installation Factory de WebSphere eXtreme Scale génère un module d'installation personnalisée (CIP) en fonction des détails que vous spécifiez dans le fichier de définition de génération. La définition de génération spécifie le module de produit à installer, l'emplacement du module CIP, les modules de maintenance à inclure dans l'installation, les fichiers du script d'installation et les fichiers supplémentaires à inclure dans le module CIP.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser l'Assistant de définition des générations pour créer un fichier de définition de génération et générer un module CIP.

Procédure

1. Exécutez le script suivant à partir du répertoire *REP_BASE_IF*/bin pour démarrer Installation Factory :
   - UNIX Linux `ifgui.sh`
   - Windows `ifgui.bat`

   Cliquez sur l'icône **Nouvelle définition de génération**.
2. Sélectionnez le produit à inclure dans le fichier de définition de génération et cliquez sur **Terminer** pour démarrer l'assistant de définition de génération.
3. Suivez les invites de l'assistant.

   Dans le panneau Scripts d'installation et de désinstallation, cliquez sur **Ajout des scripts...** pour alimenter la table avec des scripts d'installation personnalisés. Saisissez l'emplacement des fichiers script et désélectionnez la case à cocher permettant de continuer si un message d'erreur s'affiche. L'opération est arrêtée par défaut. Cliquez sur **OK** pour retourner au panneau.

Résultats

Vous avez créé et personnalisé le fichier de définition de génération et généré le module CIP si vous avez choisi de travailler en mode connecté.

Si l'assistant de définition de génération ne vous permet pas de générer le module CIP à partir du fichier de définition de génération, vous pouvez toujours le générer en exécutant le script `ifcli.sh|bat` à partir du répertoire *REP_BASE_IF*/bin.

Que faire ensuite

Installez le module CIP. Pour plus d'informations, voir «Installation d'un module CIP».

Installation d'un module CIP :

Simplifiez la procédure d'installation du produit en installant un module d'installation personnalisée (CIP). Un module CIP est une image d'installation de produit unique qui peut inclure un ou plusieurs modules de maintenance, des scripts de configuration et d'autres fichiers.

Avant de commencer

Avant d'installer un module CIP, vous devez créer un fichier de définition de génération pour spécifier les options à inclure dans le module CIP. Pour plus d'informations, voir «Création d'un fichier de définition de génération et génération d'un module d'installation personnalisée (CIP)», à la page 168.

Pourquoi et quand exécuter cette tâche

Un module CIP associe un composant de produit à des modules de maintenance, des scripts de personnalisation et d'autres fichiers, puis l'installe.

Procédure

1. Arrêtez tous les processus en cours d'exécution sur le poste de travail que vous préparez pour l'installation. Pour arrêter le gestionnaire de déploiement, exécutez le script suivant :
   - Linux UNIX *racine_profil*/bin/stopManager.sh

- Windows *racine_profil*\bin\stopManager.bat

Pour arrêter les noeuds, exécutez le script suivant :

- Linux UNIX *racine_profil*/bin/stopNode.sh
- Windows *racine_profil*\bin\stopNode.bat

2. Exécutez le script suivant pour démarrer l'installation :
   - Linux UNIX *rép_base_CIP*/bin/install
   - Windows *rép_base_CIP*\bin\install.bat
3. Suivez les invites de l'assistant pour effectuer l'installation.

   Le panneau des fonctions facultatives répertorie les fonctions que vous pouvez choisir d'installer. Toutefois, des fonctions ne peuvent pas être ajoutées de manière incrémentielle à l'environnement du produit une fois que le produit a été installé. Si vous choisissez de ne pas installer une fonction lors de l'installation initiale du produit, vous devez désinstaller, puis réinstaller le produit pour l'ajouter.

   Le panneau Extension de profil répertorie les profils existants que vous pouvez sélectionner pour étendre les fonctions d'eXtreme Scale. Si vous sélectionnez des profils existants déjà en cours d'utilisation, toutefois, un panneau d'avertissement est affiché. Pour poursuivre l'installation, arrêtez les serveurs configurés dans les profils ou cliquez sur **Précédent** pour supprimer les profils de votre sélection.

Résultats

Vous avez installé le module CIP.

Que faire ensuite

Si vous exécutez WebSphere Application Server Version 6.1 ou 7.0, vous pouvez utiliser le plug-in de l'outil de gestion de profil ou la commande **manageprofiles** pour créer et étendre des profils. Pour plus d'informations, voir «Création et augmentation de profils pour WebSphere eXtreme Scale», à la page 184.

Si vous avez étendu les profils de eXtreme Scale lors de la procédure d'installation, vous pouvez déployer des applications, démarrer un service de catalogue et démarrer les conteneurs de votre environnement WebSphere Application Server. Pour plus d'informations, voir «Configuration de WebSphere eXtreme Scale avec WebSphere Application Server», à la page 257.

Installation d'un module CIP pour appliquer la maintenance à une installation de produit existante :

Vous pouvez appliquer des modules de maintenance à une installation de produit existante en installant un module d'installation personnalisée (CIP). La procédure d'application de la maintenance à une installation existante avec un module est couramment appelée *installation intermédiaire*.

Avant de commencer

Créez un fichier de définition de génération pour spécifier les options à inclure dans le module CIP. Pour plus d'informations, voir «Création d'un fichier de définition de génération et génération d'un module d'installation personnalisée (CIP)», à la page 168.

Pourquoi et quand exécuter cette tâche

Lorsque vous appliquez la maintenance avec un module CIP qui contient un groupe de mises à jour, un groupe de correctifs ou les deux, tous les correctifs APAR précédemment installés sont désinstallés par l'assistant. Si le module CIP est au même niveau que le produit, les correctifs APAR installés précédemment ne sont conservés que s'ils sont intégrés au module CIP. Pour appliquer correctement la maintenance à une installation existante, vous devez inclure les fonctions installées dans le module CIP.

Procédure

1. Arrêtez tous les processus en cours d'exécution sur le poste de travail que vous préparez pour l'installation. Pour arrêter le gestionnaire de déploiement, exécutez le script suivant :
   - Linux UNIX *racine_profil*/bin/stopManager.sh
   - Windows *racine_profil*\bin\stopManager.bat

   Pour arrêter les noeuds, exécutez le script suivant :
   - Linux UNIX *racine_profil*\bin\stopNode.sh
   - Windows *racine_profil*\bin\stopNode.bat
2. Exécutez le script suivant pour démarrer l'installation :
   - Linux UNIX *rép_base_CIP*/bin/install
   - Windows *rép_base_CIP*\bin\install.bat
3. Suivez les invites de l'assistant pour effectuer l'installation.

   Le récapitulatif de l'aperçu d'installation répertorie la version de produit résultante et les éventuels fonctions et correctifs temporaires applicables. Ensuite, l'assistant applique la maintenance et met à jour les fonctions du produit.

Résultats

Les fichiers binaires du produit sont copiés vers le répertoire *racine_was*/properties/version/nif/backup. Vous pouvez utiliser IBM Update Installer pour désinstaller la mise à jour et restaurer votre poste de travail. Pour plus d'informations, voir «Désinstallation de mises à jour de module CIP d'une installation de produit existante.».

Désinstallation de mises à jour de module CIP d'une installation de produit existante. :

Vous pouvez supprimer des mises à jour de module CIP d'une installation de produit existante sans supprimer l'intégralité du produit. Utilisez IBM Update Installer Version 7.0.0.4 pour désinstaller les mises à jour de module CIP. Cette tâche est également appelée *désinstallation intermédiaire*.

Avant de commencer

Au moins une copie du produit doit être installé sur le système.

Procédure

1. Téléchargez la version 7.0.0.4 du programme d'installation de mises à jour à partir du site FTP suivant :

ftp://ftp.software.ibm.com/software/websphere/cw/process_server/FEP/UPDI/7004

2. Installez le programme d'installation de mises à jour. Pour plus d'informations, voir la rubrique Installation d'Update Installer pour WebSphere Software dans le Centre de documentation de WebSphere Application Server.
3. Désinstallez les éventuels groupes de correctifs, groupes de mises à jour ou correctifs temporaires que vous avez ajoutés à votre environnement après avoir installé le module CIP.
4. Désinstallez les éventuels correctifs que vous avez inclus dans l'installation intermédiaire. Cette procédure est identique à la désinstallation d'un groupe de correctifs ou d'un groupe de mises à jour. Toutefois, la maintenance incluse dans le module CIP est maintenant incluse dans une opération unique.
5. Désinstallez le module CIP à l'aide du programme d'installation de mises à jour. Les niveaux de maintenance retournent à leur état avant mise à jour et le module CIP est dénoté par l'identificateur CIP qui est ajouté comme préfixe à son nom de fichier. L'exemple suivant montre comment un module CIP est affiché de manière différente des autres modules de maintenance standard sur le panneau de sélection des modules de maintenance :

   CIP

   `com.ibm.ws.cip.7000.wxs.primary.ext.pak`

Résultats

Vous avez supprimé les mises à jour de module CIP d'une installation de produit existante.

Création d'un fichier de définition de génération et génération d'un module IIP :

Le plug-in IBM Installation Factory de WebSphere eXtreme Scale génère un module IIP en fonction des propriétés fournies par le fichier de définition de génération. Le fichier de définition de génération contient des informations telles que les modules d'installation à inclure dans le module IIP, l'ordre suivant lequel Installation Factory installe chaque module et l'emplacement du module IIP.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser l'Assistant de définition des générations pour créer un fichier de définition de génération et générer un module IIP.

Procédure

1. Exécutez le script suivant à partir du répertoire *REP_BASE_IF*`/bin` pour démarrer Installation Factory :
   - UNIX Linux `ifgui.sh`
   - Windows `ifgui.bat`
2. Cliquez sur l'icône **Create New Integrated Installation Package** pour démarrer l'assistant de définition de génération.
3. Suivez les invites de l'assistant.
   a. Dans le panneau Construct the IIP, sélectionnez un pris en charge pris en charge dans la liste, puis cliquez sur **Add Installer** pour ajouter le module d'installation au module IIP. Un panneau indiquant le nom du module, son identificateur et ses propriétés s'affiche. Pour afficher des informations spécifiques sur le module sélectionné, cliquez sur **View Installation**

Package Information. Cliquez sur **Modifier** pour entrer le chemin de répertoire du module d'installation pour chaque système d'exploitation. Si vous ajoutez actuellement un module d'installation pour WebSphere Extended Deployment, cochez la case qui vous permet d'utiliser le même module pour tous les systèmes d'exploitation pris en charge. Cliquez sur **OK** et retournez au panneau Construct the IIP. Un appel est créé par défaut.

- Pour modifier le chemin de répertoire d'un module d'installation, sélectionnez le module dans la liste des modules d'installation utilisés dans le module IIP, puis cliquez sur **Modifier**.
- Pour modifier un appel, sélectionnez-le et cliquez sur **Modifier**. Spécifiez l'emplacement d'installation par défaut de l'appel sur chaque système d'exploitation. Spécifiez l'emplacement du fichier de réponses si vous sélectionnez une installation en mode silencieux comme mode d'installation par défaut.
- Cliquez sur **Add Invocation** pour ajouter une contribution d'appel au module d'installation. Un panneau à partir duquel vous pouvez spécifier les propriétés de l'appel s'affiche.
- Cliquez sur **Supprimer** pour supprimer des modules d'installation ou des appels.

4. Vérifiez le récapitulatif de vos sélections, sélectionnez l'option **Enregistrer le fichier de définition de génération et générer le module d'installation personnalisé** et cliquez sur **Terminer**.

   Vous pouvez également sauvegarder le fichier de définition de génération sans générer le module IIP. Avec cette option, vous générez en fait le module IIP en dehors de l'assistant en exécutant le script `ifcli.bat | ifcli.sh` à partir du répertoire *rép_base_IF*/bin/.

Résultats

Vous avez créé et personnalisé le fichier de définition de génération d'un module IIP.

Que faire ensuite

Installez le module IIP.

Installation d'un module IIP :

Utilisez le plug-in IBM Installation Factory pour que WebSphere eXtreme Scale installe un module d'installation intégrée (IIP). Un module d'installation intégrée associe un ou plusieurs modules d'installation dans un même flux de travaux que vous concevez.

Avant de commencer

Avant d'installer un module CIP, vous devez créer un fichier de définition de génération pour spécifier les options à inclure dans le module CIP. Pour plus d'informations, voir «Création d'un fichier de définition de génération et génération d'un module IIP», à la page 172.

Pourquoi et quand exécuter cette tâche

Un module IIP peut inclure un ou plusieurs modules d'installation à disponibilité générale, un ou plusieurs modules CIP et d'autres fichiers et répertoires facultatifs. En installant un module IIP, vous regroupez plusieurs modules d'installation, ou

contributions, dans un même module, puis vous installez ces contributions suivant un ordre spécifique pour effectuer une installation de bout en bout.

Procédure

1. Exécutez le script suivant pour démarrer l'assistant :
   - Linux UNIX *rép_base_IIP*/bin/install
   - Windows *rép_base_IIP*\bin\install.bat
2. Cliquez sur **A propos de** dans le panneau de bienvenue pour afficher les détails du module IIP, tels que l'identificateur du module, les systèmes d'exploitation pris en charge et les modules d'installation inclus.

   Facultatif : Pour modifier les options d'installation de chaque package, cliquez sur **Modifier**.

   Facultatif : Deux boutons **Afficher le journal** sont affichés dans le panneau de l'assistant. Pour afficher le journal de chaque module, cliquez sur le bouton **Afficher le journal** affiché en regard du tableau qui répertorie les modules d'installation. Pour afficher les informations générales du journal du module IIP, cliquez sur le bouton **Afficher le journal** affiché en regard des informations de statut.
3. Sélectionnez les modules d'installation à exécuter, puis cliquez sur **Installer**. Une liste de toutes les contributions suivant leur ordre d'appel que le module IIP contient est affichée. Pour indiquer quels appels de contribution ne doivent pas être exécutés lors de l'installation, désélectionnez la case à cocher en regard de la zone **Nom de l'installation**.

Résultats

Vous avez installé un module IIP.

Modification d'un fichier de définition de génération existant pour un module IIP :

Vous pouvez éditer les propriétés d'un module IIP ou en ajouter pour personnaliser davantage l'installation.

Pourquoi et quand exécuter cette tâche

Pour modifier les propriétés d'un module IIP, modifiez le fichier de définition de génération existant.

Procédure

1. Exécutez le script suivant à partir du répertoire *REP_BASE_IF*/bin pour démarrer Installation Factory :
   - UNIX Linux ifgui.sh
   - Windows ifgui.bat
2. Cliquez sur l'icône d'**ouverture de la définition de génération** et sélectionnez le fichier de définition de génération à modifier.
3. Sélectionnez les propriétés spécifiques du module IIP à modifier. La liste suivante répertorie les modifications possibles que vous pouvez effectuer :
   - Modifiez votre sélection de mode actuelle. En mode connecté, vous créez la définition de génération à utiliser et générez éventuellement le module IIP, à

partir de votre poste de travail actuel. En mode déconnecté, vous créez le fichier de définition de génération à utiliser sur un autre poste de travail.

- Ajoutez ou supprimez les systèmes d'exploitation existants que le module IIP prend en charge.
- Editez l'identificateur et la version existants du module IIP.
- Editez l'emplacement cible du fichier de définition de génération.
- Editez l'emplacement cible du module IIP.
- Choisissez d'afficher ou non un assistant d'installation pour le module IIP. L'assistant fournit des informations sur le module IIP et les options d'installation lorsque le module IIP est exécuté.
- Ajoutez, supprimez et éditez les modules d'installation qui se trouvent dans le module IIP.

  Important : Si vous avez ajouté un système d'exploitation pris en charge et que vous n'avez pas mis à jour les propriétés du module d'installation dans le module IIP, vous recevez un message d'avertissement indiquant que les contributions sélectionnées ne contiennent pas de modules d'installation identifiés pour tous les systèmes d'exploitation pris en charge par le module IIP. Cliquez sur **Oui** pour continuer ou sur **Non** pour éditer le module d'installation.

4. Vérifiez le récapitulatif de vos sélections, sélectionnez **Enregistrer le fichier de définition de génération et générer le module d'installation personnalisé**, puis cliquez sur **Terminer**.

Installation en mode silencieux d'un module CIP ou IIP :

Vous pouvez installer en mode silencieux un module d'installation personnalisée (CIP) ou un module d'installation intégrée (IIP) pour le produit à l'aide d'un fichier de réponses complet, que vous configurez spécifiquement en fonction de vos besoins, ou de paramètres que vous transmettez à la ligne de commande.

Avant de commencer

Créez le fichier de définition de génération du module CIP ou IIP. Pour plus d'informations, voir «Création d'un fichier de définition de génération et génération d'un module d'installation personnalisée (CIP)», à la page 168.

Pourquoi et quand exécuter cette tâche

Une installation en mode silencieux utilise le même programme d'installation que l'interface graphique. Toutefois, au lieu d'afficher une interface d'assistant, elle lit toutes vos réponses dans un fichier que vous personnalisez ou dans les paramètres que vous transmettez à la ligne de commande. Si vous installez en mode silencieux un module IIP, vous pouvez appeler une contribution avec une combinaison d'options que vous spécifiez directement sur la ligne de commande, ainsi que des options que vous spécifiez dans un fichier de réponses. Toutefois, si vous transmettez des options de contribution à la ligne de commande, le programme d'installation du module IIP ignore toutes les options spécifiées dans le fichier de réponses d'une contribution spécifique. Voir les informations détaillées Installation silencieuse d'un module IIP pour plus d'informations.

Remarque : Vous devez spécifier le nom complet du fichier de réponses. Si vous spécifiez le chemin relatif, l'installation échoue sans message d'erreur.

Procédure

1. Facultatif : Si vous choisissez d'installer le module CIP ou IIP à l'aide d'un fichier de réponses, commencez par personnaliser ce fichier.
   a. Copiez le fichier de réponses, `wxssetup.response.txt`, du DVD du produit sur votre unité de disque.
   b. Ouvrez et éditez le fichier de réponses dans l'éditeur de texte de votre choix. Le fichier inclut des commentaires pour faciliter la procédure de configuration et doit inclure ces paramètres :
      - Contrat de licence
      - Emplacement de l'installation du produit

      Conseil : Le programme d'installation utilise l'emplacement que vous sélectionnez pour votre installation pour déterminer où votre instance WebSphere Application Server est installée. Si vous effectuez l'installation sur un noeud qui contient plusieurs instances WebSphere Application Server, définissez précisément votre emplacement.
   c. Exécutez le script suivant pour démarrer votre fichier de réponses personnalisé.
      - Linux UNIX `install -options /chemin_absolu/fichier_réponses.txt -silent`
      - Windows `install.bat -options C:\chemin_unité\fichier_réponses.txt -silent`
2. Facultatif : Si vous choisissez d'installer le module CIP ou IIP en transmettant certains paramètres à la ligne de commande, exécutez le script suivant pour démarrer l'installation :
   - Linux UNIX `install -silent -OPT silentInstallLicenseAcceptance=true -OPT installLocation=emplacement_install`
   - Windows `install.bat -silent -OPT silentInstallLicenseAcceptance=true -OPT installLocation=emplacement_install`

   où *emplacement_install* représente l'emplacement de votre installation WebSphere Application Server existante.
3. Recherchez les éventuelles erreurs ou un incident d'installation dans les journaux résultats.

Résultats

Vous avez installé le module CIP ou IIP.

Que faire ensuite

Si vous exécutez WebSphere Application Server Version 6.1 ou 7.0, vous pouvez utiliser le plug-in de l'outil de gestion de profil ou la commande **manageprofiles** pour créer et étendre des profils.

Si vous avez étendu les profils de eXtreme Scale lors de la procédure d'installation, vous pouvez déployer des applications, démarrer un service de catalogue et démarrer les conteneurs de votre environnement WebSphere Application Server. Pour plus d'informations, voir «Configuration de WebSphere eXtreme Scale avec WebSphere Application Server», à la page 257.

Fichier `wxssetup.response.txt` :

Vous pouvez utiliser un fichier de réponses qualifié complet pour installer WebSphere eXtreme Scale ou or WebSphere eXtreme Scale Client en mode silencieux.

ATTENTION :
N'ajoutez pas de barres obliques (/ ou \) à la fin des adresses d'installation. Les chemins vers ces adresses sont spécifiés avec l'attribut installLocation. Une barre oblique supplémentaire à la fin de l'adresse d'installation risque de provoquer l'échec de celle-ci. Le chemin suivant, par exemple, provoquera l'échec de l'installation :

```
-OPT installLocation="/usr/IBM/WebSphere/eXtremeScale/"
```

Le chemin doit être spécifié ainsi :

```
-OPT installLocation="/usr/IBM/WebSphere/eXtremeScale"
```

Fichier de réponses pour une installation complète de WebSphere eXtreme Scale

```
################################################################################
#
# Fichier d'options IBM WebSphere eXtreme Scale V7.1.1 InstallShield
#
# Nom de l'assistant : Install
# Source de l'assistant : setup.jar
#
# Ce fichier peut être utilisé pour configurer Install avec les options
# spécifiées ci-après lorsque l'assistant est exécuté avec l'option de
# ligne de commande "-options". Consultez la documentation de chacun des
# paramètres pour obtenir des informations sur la manière de modifier sa
# valeur.
# Placez toutes les valeurs entre guillemets.
#
# Une utilisation courante d'un fichier d'options consiste à exécuter
# l'assistant en mode silencieux. Le créateur du fichier d'options peut ainsi
# spécifier les paramètres de l'assistant sans avoir à exécuter l'assistant
# en mode graphique ou console. Pour utiliser ce fichier d'options en vue d'une
# exécution en mode silencieux, utilisez les arguments de ligne de commande
# suivants lorsque vous exécutez l'assistant :
#
#    -options "D:\installImage\WXS\wxssetup.response" -silent
#
# Notez que le nom de fichier de réponses complet doit être utilisé.
#
################################################################################

################################################################################
#
# Acceptation de la licence
#
# Valeurs admises :
#  true - Accepte la licence. Installe le produit.
#  false - Refuse la licence. L'installation n'est pas effectuée.
#
# Si aucune installation n'est effectuée, cela est consigné dans un fichier journal
# temporaire, dans le répertoire temporaire de l'utilisateur.
#
# En spécifiant la valeur "true" pour la propriété silentInstallLicenseAcceptance
# dans ce fichier de réponses, vous certifiez que vous avez lu et que vous
# acceptez les dispositions des Conditions Internationales d'Utilisation des
# Logiciels IBM accompagnant ce programme, lesquelles dispositions sont exposées
# dans le fichier
# RACINE_CD\XD\wxs.primary.pak\repository\legal.xs\license.xs.  Si
# vous n'acceptez pas ces dispositions, ne changez pas la valeur ou
# ne téléchargez pas, n'installez pas, ne copiez pas, n'utilisez pas le programme
# et n'y accédez pas non plus et retournez-le avec l'autorisation
```

```
# d'utilisation au tiers auprès duquel vous l'avez acheté
# pour vous faire rembourser.
#
-OPT silentInstallLicenseAcceptance="false"


################################################################################
# Vérification des prérequis non bloquants
#
# Si vous souhaitez désactiver la vérification des prérequis non bloquants,
# supprimez la mise en commentaire de la ligne ci-après. Cette action indique
# au programme d'installation de poursuivre l'installation et de consigner les
# avertissements bien que la vérification des prérequis ait échoué.
#
#-OPT disableNonBlockingPrereqChecking="true"


################################################################################
#
# Emplacement d'installation
#
# Emplacement d'installation du produit. Spécifiez un répertoire valide dans
# lequel le produit doit être installé. Si le répertoire contient des espaces,
# placez-le entre guillemets, comme indiqué dans l'exemple Windows ci-après. Notez
# que les espaces dans l'emplacement d'installation ne sont pris en charge que
# sur les systèmes d'exploitation Windows. La longueur maximale du chemin d'accès
# est de 60 caractères pour Windows.
#
# Vous trouverez ci-après une liste des emplacements d'installation par défaut
# pour chaque système d'exploitation pris en charge si vous effectuez
# l'installation en tant qu'utilisateur root.  Par défaut, dans ce fichier de
# réponses, l'emplacement d'installation de Windows est utilisé.  Si vous
# souhaitez utiliser l'emplacement d'installation par défaut d'un autre
# système d'exploitation, supprimez la mise en commentaire de l'entrée
# d'emplacement d'installation par défaut (en supprimant le signe '#'), puis
# placez en commentaire (en ajoutant le signe '#') l'entrée de système
# d'exploitation Windows ci-après.
#
# L'emplacement d'installation est utilisé pour déterminer si WebSphere eXtreme
# Scale doit être installé comme déploiement autonome ou s'il doit être intégré
# à une installation WebSphere Application Server existante.
#
# Si l'emplacement spécifié correspond à une installation WebSphere Application
# Server ou WebSphere Network Deployment existante, eXtreme Scale est intégré
# au serveur WebSphere Application Server existant.  Si l'emplacement spécifié
# correspond à un répertoire nouveau ou vide, WebSphere eXtreme Scale est
# installé comme déploiement autonome.
#
# Remarque : Si l'emplacement d'installation spécifié contient une installation de
# WebSphere eXtreme Scale, WebSphere eXtended Deployment DataGrid ou
# ObjectGrid, l'installation échoue.
#
# Emplacement d'installation par défaut sous AIX :
#
#  -OPT installLocation="/usr/IBM/WebSphere/eXtremeScale"
#
# Emplacement d'installation par défaut sous HP-UX, Solaris ou Linux :
#
#  -OPT installLocation="/opt/IBM/WebSphere/eXtremeScale"
#
#
# Emplacement d'installation par défaut sous Windows :
#
-OPT installLocation="C:\Program Files\IBM\WebSphere\eXtremeScale"

#
# Si vous effectuez l'installation en tant qu'utilisateur autre que root sous
```

```
# Unix ou qu'administrateur sous Windows, les emplacements d'installation par
# défaut ci-après sont recommandés. Assurez-vous de disposer des droits d'accès
# en écriture pour l'emplacement d'installation choisi.
#
# Emplacement d'installation par défaut sous AIX :
#
#  -OPT installLocation="<répertoire de base de l'utilisateur>
   /IBM/WebSphere/eXtremeScale"
#
# Emplacement d'installation par défaut sous HP-UX, Solaris ou Linux :
#
#  -OPT installLocation="<répertoire de base de l'utilisateur>
   /IBM/WebSphere/eXtremeScale"
#
# Emplacement d'installation par défaut sous Windows :
#
# -OPT installLocation="C:\IBM\WebSphere\eXtremeScale"


###############################################################################
#  Installation de fonctions facultatives
#
# Spécifiez les fonctions facultatives à installer en affectant à chacune
# d'elles la valeur "true".  Affectez aux fonctions facultatives que vous ne
# souhaitez pas installer la valeur "false".
#
# Les options selectServer, selectClient, selectPF et selectXSStreamQuery ne sont
# valides que si l'option installLocation ci-dessus contient une installation de
# WebSphere Application Server.  Ces options sont ignorées sur une
# installation autonome de WebSphere eXtreme Scale.
#
# Sur l'installation autonome de WebSphere eXtreme Scale, le serveur et le
# client eXtreme Scale sont automatiquement installés.
# Les options pour une installation autonome d'eXtreme Scale sont :
# selectXSConsoleOther et selectXSStreamQueryOther.

#
# Cette option, si elle est sélectionnée, installe les composants
# nécessaires pour exécuter des serveurs WebSphere eXtreme Scale
# et le fournisseur de service de mémoire
# cache dynamique eXtreme Scale. Si cette option est sélectionnée,
# le client WebSphere eXtreme Scale doit également être
# sélectionné en supprimant sa mise en
# commentaire et en lui affectant la valeur "true".
# Sinon, l'installation en mode silencieux ECHOUE.
#
-OPT selectServer="true"

#
# Cette option, si elle est sélectionnée, installe les composants
# nécessaires pour exécuter des applications client
# WebSphere eXtreme Scale. Si l'option Serveur est
# sélectionnée ci-dessus, cette option doit également être
# sélectionnée en supprimant
# sa mise en commentaire et en lui affectant la valeur "true". Sinon,
# l'installation en mode silencieux ECHOUE.
#
-OPT selectClient="true"

#
# Cette option, si elle est sélectionnée, installe les composants
# nécessaires pour exécuter la console WebSphere eXtreme Scale.
# Si cette option est sélectionnée,
# l'emplacement d'installation spécifié ci-dessus doit correspondre
# à un répertoire nouveau ou vide car l'option de la console
# n'est valide que pour un déploiement
# WebSphere eXtreme Scale autonome. Pour installer cette option,
```

```
# la mise en commentaire de la ligne d'option ci-après doit être
# supprimée et vous devez définir cette option comme "true".
#-OPT selectXSConsoleOther="false"

#
# Les options suivantes, si elles sont sélectionnées installeront la
# fonctionnalité OBSOLETE.
#
# Cette option sélectionne WebSphere Partition Facility pour l'installer.
# Cette fonctionnalité est OBSOLETE.  Pour installer cette option,
# la mise en commentaire de la ligne d'option ci-après doit être supprimée
# et vous devez lui affecter la valeur "true".
#
#-OPT selectPF="false"

#
# Cette option sélectionne WebSphere eXtreme Scale StreamQuery for WAS pour
# l'installation. Cette fonctionnalité est OBSOLETE. Pour installer
# cette option, la mise en commentaire de la ligne d'option
# ci-après doit être supprimée et
# vous devez lui affecter la valeur "true".
# Si cette option est sélectionnée, le client WebSphere
# eXtreme Scale doit également être sélectionné en supprimant sa
# mise en commentaire et en lui affectant la valeur "true".
# Sinon, l'installation en mode silencieux ECHOUE.
#
#-OPT selectXSStreamQuery="false"

#
# Cette option sélectionne WebSphere eXtreme Scale StreamQuery for J2SE
# pour l'installation. Cette fonctionnalité est OBSOLETE.
# Pour installer cette option, la mise en commentaire de la ligne d'option
# ci-après doit être supprimée et
# vous devez lui affecter la valeur "true".
# Si cette option est sélectionnée, le client WebSphere
# eXtreme Scale doit également être sélectionné en supprimant
#  sa mise en commentaire et en lui affectant la valeur "true".
# Sinon, l'installation en mode silencieux ECHOUE.
#
#-OPT selectXSStreamQueryOther="false"


###############################################################################
#  Liste des profils à étendre
#
# Spécifiez les profils existants à étendre ou placez en commentaire
# la ligne permettant d'étendre tous les profils existants
# détectés par l'installation.
#
# Pour spécifier plusieurs profils, utilisez une virgule afin de séparer
# les différents noms de profil.
# Par exemple, "AppSrv01,Dmgr01,Custom01". La liste ne doit pas
# contenir d'espace.
#
-OPT profileAugmentList=""


###############################################################################
# Contrôle du traçage
#
# Le format de la sortie de trace peut être contrôlé via l'option
# -OPT traceFormat=ALL
#
# Les choix de format sont 'text' et 'XML'. Par défaut, ces deux formats
# seront générés, dans deux fichiers de trace différents.
#
# Si un seul format est requis, utilisez l'option traceFormat pour le
```

```
# spécifier, comme suit :
#
# Valeurs admises :
#
#  text -  Les lignes du fichier de trace sont au format texte pour une
#          meilleure lisibilité.
#  XML  -  Les lignes du fichier de trace se trouveront au format XML
#          de consignation Java standard qui peut être affiché
#          à l'aide de tout éditeur de texte ou éditeur XML
#          ou à l'aide de l'outil Chainsaw d'Apache, à
#          l'adresse URL suivante :
#          (http://logging.apache.org/log4j/docs/chainsaw.html).
#
# La quantité d'informations de trace capturées peut être
# contrôlée à l'aide de l'option suivante :
# -OPT traceLevel=INFO
#
# Valeurs admises :
#
# Niveau    Niveau
# de trace  numérique  Description
# -------  ---------  ------------------------------------------------------
# OFF         0       Aucun fichier de trace n'est généré
# SEVERE      1       Seules les erreurs graves sont consignées
#                     dans le fichier de trace
# WARNING     2       Les messages relatifs aux exceptions non fatales et aux
#                     avertissements sont
#                     ajoutés au fichier de trace
# INFO        3       Les messages d'information sont ajoutés au fichier de trace
#                     (niveau de trace par défaut)
# CONFIG      4       Les messages liés à la configuration sont ajoutés
#                     au fichier de trace
# FINE        5       Traçage des appels de méthode pour les méthodes
#                     publiques
# FINER       6       Traçage des appels de méthode pour les méthodes
#                     non publiques, exceptées
#                     les méthodes d'accès get et set
# FINEST      7       Traçage de tous les appels de méthode ;
#                     l'entrée/la sortie de trace inclut
#                     les paramètres et la valeur de retour
```

Fichier de réponses pour une installation de WebSphere eXtreme Scale Client

```
################################################################################
#
# Fichier d'options IBM WebSphere eXtreme Scale V7.1.1 InstallShield
#
# Nom de l'assistant : Install
# Source de l'assistant : setup.jar
#
# Ce fichier peut être utilisé pour configurer Install avec
# les options spécifiées ci-après lorsque l'assistant
# est exécuté avec l'option de
# ligne de commande "-options". Consultez la documentation de
# chacun des paramètres pour obtenir des informations sur
# la manière de modifier sa valeur.
# Placez toutes les valeurs entre guillemets.
#
# Une utilisation courante d'un fichier d'options consiste à exécuter
# l'assistant en mode silencieux. Le créateur du fichier d'options peut ainsi
# spécifier les paramètres de l'assistant sans avoir à exécuter
# l'assistant en mode graphique ou console. Pour utiliser ce fichier d'options
# en vue d'une exécution en mode silencieux, utilisez les arguments de
# ligne de commande suivants lorsque vous exécutez l'assistant :
#
#    -options "D:\installImage\WXS_Client\wxssetup.response" -silent
#
```

```
# Notez que le nom de fichier de réponses complet doit être utilisé.
#
################################################################################

################################################################################
#
# Acceptation de la licence
#
# Valeurs admises :
#  true - Accepte la licence. Installe le produit.
#  false - Refuse la licence. L'installation n'est pas effectuée.
#
# Si aucune installation n'est effectuée, cela est consigné dans un
#  fichier journal temporaire, dans le répertoire temporaire de l'utilisateur.
#
# En spécifiant la valeur "true" pour la propriété
# silentInstallLicenseAcceptance
# dans ce fichier de réponses, vous certifiez que vous avez lu et que vous
# acceptez les dispositions des Conditions Internationales d'Utilisation des
# Logiciels IBM accompagnant ce programme, lesquelles dispositions sont exposées
# dans le fichier
# RACINE_CD\WXS_Client\wxs.client.primary.pak\repository\legal.xs.client\license.xs.
# Si vous n'acceptez pas ces conditions, ne changez pas la valeur ou
# ne téléchargez pas, n'installez pas, ne copiez pas, n'utilisez pas
# le programme et n'y accédez pas non plus et
# retournez-le avec l'autorisation d'utilisation au tiers auprès
# duquel vous l'avez acheté pour vous faire rembourser.
#
-OPT silentInstallLicenseAcceptance="false"


################################################################################
# Vérification des prérequis non bloquants
#
# Si vous souhaitez désactiver la vérification des prérequis non bloquants,
# supprimez la mise en commentaire de la ligne ci-après. Cette action indique
# au programme d'installation de poursuivre l'installation et de consigner les
# avertissements bien que la vérification des prérequis ait échoué.
#
#-OPT disableNonBlockingPrereqChecking="true"


################################################################################
#
# Emplacement d'installation
#
# Emplacement d'installation du produit. Spécifiez un répertoire valide dans
# lequel le produit doit être installé. Si le répertoire contient des espaces,
# placez-le entre guillemets, comme indiqué dans l'exemple Windows ci-après. Notez
# que les espaces dans l'emplacement d'installation ne sont pris en charge que
# sur les systèmes d'exploitation Windows. La longueur maximale du chemin d'accès
# est de 60 caractères pour Windows.
#
# Vous trouverez ci-après une liste des emplacements d'installation par défaut
# pour chaque système d'exploitation pris en charge si vous effectuez
# l'installation en tant qu'utilisateur root.  Par défaut, dans ce fichier de
# réponses, l'emplacement d'installation de Windows est utilisé.  Si vous
# souhaitez utiliser l'emplacement d'installation par défaut d'un autre
# système d'exploitation, supprimez la mise en commentaire de l'entrée
# d'emplacement d'installation par défaut (en supprimant le signe '#'), puis
# placez en commentaire (en ajoutant le signe '#') l'entrée de système
# d'exploitation Windows ci-après.
#
# L'emplacement d'installation est utilisé pour déterminer si WebSphere eXtreme
# Scale doit être installé comme déploiement autonome ou s'il doit être
# intégré à une installation WebSphere Application Server existante.
#
```

```
# Si l'emplacement spécifié correspond à une installation WebSphere Application
# Server ou WebSphere Network Deployment existante, eXtreme Scale est intégré
# au serveur WebSphere Application Server existant. Si l'emplacement spécifié
# correspond à un répertoire nouveau ou vide, WebSphere eXtreme Scale est
# installé comme déploiement autonome.
#
# Remarque : Si l'emplacement d'installation spécifié contient une installation
# de WebSphere eXtreme Scale, WebSphere eXtended Deployment DataGrid ou
# ObjectGrid, l'installation échoue.
#
# Emplacement d'installation par défaut sous AIX :
#
#  -OPT installLocation="/usr/IBM/WebSphere/eXtremeScale"
#
# Emplacement d'installation par défaut sous HP-UX, Solaris ou Linux :
#
#  -OPT installLocation="/opt/IBM/WebSphere/eXtremeScale"
#
#
# Emplacement d'installation par défaut sous Windows :
#
-OPT installLocation="C:\Program Files\IBM\WebSphere\eXtremeScale"

#
# Si vous effectuez l'installation en tant qu'utilisateur autre que root sous
# Unix ou qu'administrateur sous Windows, les emplacements d'installation par
# défaut ci-après sont recommandés. Assurez-vous de disposer des droits d'accès
# en écriture pour l'emplacement d'installation choisi.
#
# Emplacement d'installation par défaut sous AIX :
#
#  -OPT installLocation="<répertoire de base de
#  l'utilisateur>/IBM/WebSphere/eXtremeScale"
#
# Emplacement d'installation par défaut sous HP-UX, Solaris ou Linux :
#
#  -OPT installLocation="<répertoire de base de
#  l'utilisateur>/IBM/WebSphere/eXtremeScale"
#
# Emplacement d'installation par défaut sous Windows :
#
# -OPT installLocation="C:\IBM\WebSphere\eXtremeScale"


################################################################################
#  Liste des profils à étendre
#
# Spécifiez les profils existants à étendre ou placez en commentaire la ligne
# permettant d'étendre tous les profils existants détectés par l'installation.
#
# Pour spécifier plusieurs profils, utilisez une virgule afin de séparer les
# différents noms de profil.
# Par exemple, "AppSrv01,Dmgr01,Custom01". La liste ne doit pas contenir d'espace.
#
-OPT profileAugmentList=""


################################################################################
# Contrôle du traçage
#
# Le format de la sortie de trace peut être contrôlé via l'option
# -OPT traceFormat=ALL
#
# Les choix de format sont 'text' et 'XML'. Par défaut, ces deux formats seront
# générés, dans deux fichiers de trace différents.
#
# Si un seul format est requis, utilisez l'option traceFormat pour le spécifier,
```

```
# comme suit :
#
# Valeurs admises :
#
#  text -  Les lignes du fichier de trace sont au format texte pour une meilleure
#          lisibilité.
#  XML  -  Les lignes du fichier de trace se trouveront au format XML
#          de consignation Java
#          standard qui peut être affiché à l'aide de tout éditeur
#          de texte ou éditeur XML
#          ou à l'aide de l'outil Chainsaw d'Apache, à l'adresse URL suivante :
#          (http://logging.apache.org/log4j/docs/chainsaw.html).
#
# La quantité d'informations de trace capturées peut être contrôlée
# à l'aide de l'option suivante :
# -OPT traceLevel=INFO
#
# Valeurs admises :
#
# Niveau    Niveau
# de trace  numérique  Description
# -------  ---------  ------------------------------------------------------
# OFF         0       Aucun fichier de trace n'est généré
# SEVERE      1       Seules les erreurs graves sont consignées
#                     dans le fichier de trace
# WARNING     2       Les messages relatifs aux exceptions non fatales et aux
#                     avertissements sont
#                     ajoutés au fichier de trace
# INFO        3       Les messages d'information sont ajoutés au fichier de trace
#                     (niveau de trace par défaut)
# CONFIG      4       Les messages liés à la configuration sont ajoutés
#                     au fichier de trace
# FINE        5       Traçage des appels de méthode pour les
#                     méthodes publiques
# FINER       6       Traçage des appels de méthode pour les méthodes
#                     non publiques, exceptées
#                     les méthodes d'accès get et set
# FINEST      7       Traçage de tous les appels de méthode ;
#                     l'entrée/la sortie de trace inclut
#                     les paramètres et la valeur de retour
```

Création et augmentation de profils pour WebSphere eXtreme Scale

Une fois que vous avez installé le produit, créez des types de profil uniques et étendez les profils existants de WebSphere eXtreme Scale.

Avant de commencer

Installez WebSphere eXtreme Scale. Pour plus d'informations, voir «Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client avec WebSphere Application Server», à la page 163.

L'extension de profils en vue de leur utilisation avec WebSphere eXtreme Scale est facultative, mais requise dans les cas suivants :

- Pour démarrer automatiquement un service de catalogue ou un conteneur dans un processus WebSphere Application Server. Si vous n'étendez pas les profils des serveurs, les serveurs peuvent être uniquement démarrés à l'aide d'un programme en utilisant l'API ServerFactory ou comme processus distincts à l'aide de scripts **startOgServer**.
- Pour utiliser l'infrastructure PMI (Performance Monitoring Infrastructure) afin de surveiller les mesures de WebSphere eXtreme Scale.

- Pour afficher la version de WebSphere eXtreme Scale dans la console d'administration de WebSphere Application Server.

Pourquoi et quand exécuter cette tâche

Exécution dans WebSphere Application Server Version 6.1 ou 7.0

Si votre environnement contient WebSphere Application Server Version 6.1 ou 7.0, vous pouvez utiliser le plug-in de l'outil de gestion de profil ou la commande **manageprofiles** pour créer et étendre des profils.

Que faire ensuite

Suivant la tâche que vous choisissez d'effectuer, lancez la console Premiers pas pour obtenir de l'aide lors de la configuration et du test de l'environnement de votre produit. La console Premier pas se trouve dans le répertoire *racine_install_wxs*\firststeps\wxs\firststeps.bat. Vous pouvez également créer ou étendre des profils supplémentaires en répétant l'une des tâches précédentes.

Utilisation de l'interface graphique pour créer des profils :

Utilisez l'interface graphique, fournie par le plug-in Profile Management Tool, pour créer des profils pour WebSphere eXtreme Scale. n profil est un ensemble de fichiers qui définissent l'environnement d'exécution.

Avant de commencer

Vous ne pouvez pas utiliser l'interface graphique pour étendre des profils dans les scénarios suivants :

- **Installation 64 bits de WebSphere Application Server** :

  L'outil de gestion de profil n'existe pas pour les installations 64 bits de WebSphere Application Server. Utilisez le script **manageprofiles** à partir de la ligne de commande pour ces installations.

Pourquoi et quand exécuter cette tâche

Pour utiliser les fonctions du produit, le plug-in de l'outil de gestion de profil permet à l'interface graphique de vous aider à configurer des profils, tels qu'un profil WebSphere Application Server, un profil de gestionnaire de déploiement, un profil de cellule et un profil personnalisé. Vous pouvez étendre des profils pendant ou après l'installation de WebSphere eXtreme Scale.

Procédure

Utilisez l'interface graphique de l'outil de gestion des profils pour créer des profils. Choisissez l'une des options suivantes pour démarrer l'assistant :

- Sélectionnez **Outil de gestion de profil** dans la console Premiers pas.
- Accédez à l'outil de gestion de profil à partir du menu **Démarrer**.
- Exécutez le script ./pmt.sh|bat à partir du répertoire *racine_install*/bin/ProfileManagement.

Que faire ensuite

Vous pouvez créer d'autres profils ou étendre des profils existants. Pour redémarrer l'outil de gestion de profil, exécutez la commande **./pmt.sh|bat** à

partir du répertoire *racine_was*/bin/ProfileManagement, ou sélectionnez **Outil de gestion de profil** dans la console Premiers pas.

Démarrez un service de catalogue, démarrez des conteneurs et configurez les ports TCP dans votre environnement WebSphere Application Server. Pour plus d'informations, voir «Configuration de WebSphere eXtreme Scale avec WebSphere Application Server», à la page 257.

Utilisation de l'interface graphique pour étendre des profils :

Après avoir installé le produit, vous pouvez étendre un profil existant pour le rendre compatible avec WebSphere eXtreme Scale.

Pourquoi et quand exécuter cette tâche

Lorsque vous étendez un profil existant, vous modifiez le profil en appliquant un modèle d'extension spécifique à un produit. Par exemple, les serveurs WebSphere eXtreme Scale ne démarrent pas automatiquement à moins que le profil du serveur ne soit étendu avec le modèle xs_augment.

- Etendez le profil avec le modèle xs_augment si vous avez installé le client eXtreme Scale ou le client et le serveur.
- N'étendez le profil avec le modèle pf_augment que si vous avez installé la fonction de partitionnement.
- Appliquez les deux modèles si votre environnement contient le client eXtreme Scale et l'utilitaire de partitionnement.

Procédure

Utilisez l'interface graphique de l'outil de gestion des profils pour étendre les profils pour eXtreme Scale. Choisissez l'une des options suivantes pour démarrer l'assistant :

- Sélectionnez **Outil de gestion de profil** dans la console Premiers pas.
- Accédez à l'outil de gestion de profil à partir du menu **Démarrer**.
- Exécutez le script ./pmt.sh|bat depuis le répertoire *racine_was*/bin/ProfileManagement.

Que faire ensuite

Vous pouvez étendre d'autres profils. Pour redémarrer l'outil de gestion de profil, exécutez la commande **./pmt.sh|bat** à partir du répertoire *racine_was*/bin/ProfileManagement ou sélectionnez **Outil de gestion de profil** dans la console Premiers pas.

Démarrez un service de catalogue, démarrez des conteneurs et configurez les ports TCP dans votre environnement WebSphere Application Server. Pour plus d'informations, voir «Configuration de WebSphere eXtreme Scale avec WebSphere Application Server», à la page 257.

Commande manageprofiles :

Vous pouvez utiliser l'utilitaire **manageprofiles** pour créer des profils à l'aide du modèle WebSphere eXtreme Scale ou étendre et réduire des profils de serveur d'applications existants à l'aide des modèle d'extension de eXtreme Scale. Pour utiliser les fonctions du produit, votre environnement doit contenir au moins un profil étendu pour le produit.

- Pour pouvoir créer et étendre des profils, vous devez installer eXtreme Scale. Pour plus d'informations, voir «Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client avec WebSphere Application Server», à la page 163.

Rôle

La commande **manageprofiles** crée l'environnement d'exécution d'un processus de produit dans un ensemble de fichiers appelé profil. Le profil définit l'environnement d'exécution. Vous pouvez effectuer les actions suivantes à l'aide de la commande **manageprofiles** :

- Création et extension d'un profil de gestionnaire de déploiement
- Création et extension d'un profil personnalisé
- Création et extension d'un profil de serveur d'applications autonome
- Création et extension d'un profil de cellule
- Réduction de tout type de profil

Lorsque vous étendez un profil existant, vous modifiez le profil en appliquant un modèle d'extension spécifique à un produit.

- Etendez le profil avec le modèle xs_augment si vous avez installé le client eXtreme Scale ou le client et le serveur.
- Etendez le profil avec le modèle pf_augment si vous n'avez installé que l'utilitaire de partitionnement.
- Appliquez les deux modèles si votre environnement contient le client eXtreme Scale et l'utilitaire de partitionnement.

Emplacement

Le fichier de commandes se trouve dans le répertoire *racine_install*/bin.

Syntaxe

Pour obtenir une aide détaillée, utilisez le paramètre **-help** :

```
./manageprofiles.sh|bat
-create -templatePath racine_install/profileTemplates/xs_augment/dmgr -help
```

Dans les sections ci-après, chaque tâche que vous pouvez effectuer à l'aide de la commande **manageprofiles** est décrite, avec une liste des paramètres requis. Pour des détails sur les paramètres facultatifs à spécifier pour chaque tâche, reportez-vous à la commande **manageprofiles**, dans le Centre de documentation de WebSphere Application Server.

Création d'un profil de gestionnaire de déploiement

Vous pouvez utiliser la commande **manageprofiles** pour créer un profil de gestionnaire de déploiement. Le gestionnaire de déploiement administre les serveurs d'applications fédérés dans la cellule.

Paramètres

-create
Crée un profil. (Obligatoire)

-templatePath *chemin_modèle*
Indique le chemin du modèle. (Obligatoire)

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/dmgr
```

où *type_modèle* est `xs_augment` ou `pf_augment`.

Exemple

- Utilisation du modèle xs_augment :

  ```
  ./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/xs_augment/dmgr
  ```

- Utilisation du modèle pf_augment :

  ```
  ./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/pf_augment/dmgr
  ```

Création d'un profil personnalisé.

Vous pouvez utiliser la commande **manageprofiles** pour créer un profil personnalisé. Un profil personnalisé est un noeud vide que vous personnalisez via le gestionnaire de déploiement pour inclure des serveurs d'applications, des clusters ou d'autres processus Java.

Paramètres

-create
Crée un profil. (Obligatoire)

-templatePath *chemin_modèle*
Indique le chemin du modèle. (Obligatoire)

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/managed
```

où *type_modèle* est `xs_augment` ou `pf_augment`.

Exemple

- Utilisation du modèle xs_augment :

  ```
  ./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/xs_augment/managed
  ```

- Utilisation du modèle pf_augment :

  ```
  ./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/pf_augment/managed
  ```

Création d'un profil de serveur d'applications autonome

Vous pouvez utiliser la commande **manageprofiles** pour créer un profil de serveur d'applications autonome.

Paramètres

-create
Crée un profil. (Obligatoire)

-templatePath *chemin_modèle*
Indique le chemin du modèle. (Obligatoire)

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/default
```

où *type_modèle* est `xs_augment` ou `pf_augment`.

Exemple

- Utilisation du modèle xs_augment :

```
./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/xs_augment/default
```

- Utilisation du modèle pf_augment :

```
./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/pf_augment/default
```

Création d'un profil de cellule

Vous pouvez utiliser la commande **manageprofiles** pour créer un profil de serveur, qui comprend un gestionnaire de déploiement et un serveur d'applications.

Paramètres

Indiquez les paramètres suivants dans le modèle de gestionnaire de déploiement :

-create
Crée un profil. (Obligatoire)

-templatePath *chemin_modèle*
Indique le chemin du modèle. (Obligatoire)

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/cell/dmgr
```

où *type_modèle* est `xs_augment` ou `pf_augment`.

Indiquez les paramètres suivants avec le modèle de serveur d'applications :

-create
Crée un profil. (Obligatoire)

-templatePath *chemin_modèle*
Indique le chemin du modèle. (Obligatoire)

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/cell/default
```

où *type_modèle* est `xs_augment` ou `pf_augment`.

Exemple

- Utilisation du modèle xs_augment :

```
./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/xs_augment/cell/dmgr
-nodeProfilePath racine_install/profiles/AppSrv01 -cellName cell01dmgr -nodeName node01dmgr
-appServerNodeName node01

./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/xs_augment/cell/default
-dmgrProfilePath racine_install/profiles/Dmgr01 -portsFile
racine_install/profiles/Dmgr01/properties/portdef.props -nodePortsFile
racine_install/profiles/Dmgr01/properties/nodeportdef.props -cellName cell01dmgr
-nodeName node01dmgr -appServerNodeName node01
```

- Utilisation du modèle pf_augment :

```
./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/pf_augment/cell/dmgr
-nodeProfilePath racine_install/profiles/AppSrv01 -cellName cell01dmgr -nodeName node01dmgr
-appServerNodeName node01

./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/pf_augment/cell/default
-dmgrProfilePath racine_install/profiles/Dmgr01 -portsFile
racine_install/profiles/Dmgr01/properties/portdef.props -nodePortsFile
racine_install/profiles/Dmgr01/properties/nodeportdef.props -cellName cell01dmgr
-nodeName node01dmgr -appServerNodeName node01
```

Extension d'un profil de gestionnaire de déploiement

Vous pouvez utiliser la commande **manageprofiles** pour étendre un profil de gestionnaire de déploiement.

Paramètres

-augment
Etend le profil existant. (Obligatoire)

-profileName
Spécifie le nom du profil. (Obligatoire)

-templatePath *chemin_modèle*
Spécifie le chemin des fichiers de modèle qui se trouvent dans le répertoire racine d'installation. (Obligatoire)

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/dmgr
```

où *type_modèle* est `xs_augment` ou `pf_augment`.

Exemple

- Utilisation du modèle xs_augment :

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath racine_install/profileTemplates/xs_augment/dmgr
```

- Utilisation du modèle pf_augment :

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath racine_install/profileTemplates/pf_augment/dmgr
```

Extension d'un profil personnalisé

Vous pouvez utiliser la commande **manageprofiles** pour étendre un profil personnalisé.

Paramètres

-augment
Etend le profil existant. (Obligatoire)

-profileName
Spécifie le nom du profil. (Obligatoire)

-templatePath *chemin_modèle*
Spécifie le chemin des fichiers de modèle qui se trouvent dans le répertoire racine d'installation. (Obligatoire)

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/managed
```

où *type_modèle* est `xs_augment` ou `pf_augment`.

Exemple

- Utilisation du modèle xs_augment :

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath install_root/profileTemplates/xs_augment/managed
```

- Utilisation du modèle pf_augment :

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath install_root/profileTemplates/pf_augment/managed
```

Extension d'un profil de serveur d'applications autonome

Vous pouvez utiliser la commande **manageprofiles** pour étendre un profil de serveur d'applications autonome.

Paramètres

-augment
Etend le profil existant. (Obligatoire)

-profileName
Spécifie le nom du profil. (Obligatoire)

-templatePath *chemin_modèle*
Spécifie le chemin des fichiers de modèle qui se trouvent dans le répertoire racine d'installation. (Obligatoire)

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/default
```

où *type_modèle* est `xs_augment` ou `pf_augment`.

Exemple

- Utilisation du modèle xs_augment :

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath install_root/profileTemplates/xs_augment/default
```

- Utilisation du modèle pf_augment :

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath install_root/profileTemplates/pf_augment/default
```

Extension d'un profil de cellule

Vous pouvez utiliser la commande **manageprofiles** pour étendre un profil de cellule.

Paramètres

Indiquez les paramètres suivants pour le profil de gestionnaire de déploiement :

-augment
Etend le profil existant. (Obligatoire)

-profileName
Spécifie le nom du profil. (Obligatoire)

-templatePath *chemin_modèle*
Spécifie le chemin des fichiers de modèle qui se trouvent dans le répertoire racine d'installation. (Obligatoire)

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/cell/dmgr
```

où *type_modèle* est `xs_augment` ou `pf_augment`.

Indiquez les paramètres suivants pour le profil de serveur d'applications :

-augment
Etend le profil existant. (Obligatoire)

-profileName
Spécifie le nom du profil. (Obligatoire)

-templatePath *chemin_modèle*
Spécifie le chemin des fichiers de modèle qui se trouvent dans le répertoire racine d'installation. (Obligatoire)

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/cell/default
```

où *type_modèle* est `xs_augment` ou `pf_augment`.

Exemple

- Utilisation du modèle xs_augment :

```
./manageprofiles.sh|bat -augment -profileName profile01 -templatePath install_root
    /profileTemplates/xs_augment/cell/dmgr

./manageprofiles.sh|bat -augment -profileName profile01 -templatePath install_root
    /profileTemplates/xs_augment/cell/default
```

- Utilisation du modèle pf_augment :

```
./manageprofiles.sh|bat -augment -profileName profile01 -templatePath install_root
    /profileTemplates/pf_augment/cell/dmgr

./manageprofiles.sh|bat -augment -profileName profile01 -templatePath install_root
    /profileTemplates/pf_augment/cell/default
```

Réduction d'un profil

Pour réduire un profil, spécifiez le paramètre **-ignoreStack** avec le paramètre **-templatePath**, en plus des paramètres **-unaugment** et **-profileName** requis.

Paramètres

-unaugment
Réduit un profil précédemment étendu. (Obligatoire)

-profileName
Spécifie le nom du profil. Le paramètre est généré par défaut si aucune valeur n'est spécifiée. (Obligatoire)

-templatePath *chemin_modèle*
Spécifie le chemin des fichiers de modèle qui se trouvent dans le répertoire racine d'installation. (facultatif).

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/type_profil
```

où *type_modèle* est `xs_augment` ou `pf_augment` et *type_profil* correspond à l'un des quatre types de profil suivants :

- `dmgr` : profil du gestionnaire de déploiement
- `managed` : profil personnalisé
- `default` : profil de serveur d'applications autonome
- `cell` : profil de cellule

-ignoreStack
Utilisé avec le paramètre **-templatePath** pour réduire un profil qui a été étendu. (Facultatif).

Exemple

- Utilisation du modèle xs_augment :

```
./manageprofiles.sh|bat -unaugment -profileName profile01 -ignoreStack
-templatePath racine_install/profileTemplates/xs_augment/type_profil
```

- Utilisation du modèle pf_augment :

```
./manageprofiles.sh|bat -unaugment -profileName profile01 -ignoreStack
-templatePath racine_install/profileTemplates/pf_augment/type_profil
```

Profils non root :

Vous pouvez octroyer à l'utilisateur non root des autorisations pour des fichiers et répertoires afin de permettre à cet utilisateur de créer un profil pour le produit. L'utilisateur non root peut également augmenter un profil qui a été créé par un utilisateur root, par un autre utilisateur non root ou par lui-même.

Dans un environnement WebSphere Application Server, les autorisations de création et d'utilisation de profils des utilisateurs non root (non administrateurs) sont limitées. Dans le plug-in de l'outil de gestion de profil, les noms et les valeurs de port uniques sont désactivés pour les utilisateurs non root. Ces derniers doivent modifier dans l'outil de gestion de profils valeurs par défaut des zones de nom du profil, de nom du noeud, de nom de la cellule et d'affectations des ports. Pensez à affecter aux utilisateurs non root une plage de valeurs pour chacune de ces zones. Vous pouvez attribuer la responsabilité à des utilisateurs non root d'adhérer aux plages de valeurs adéquates et de maintenir l'intégrité de leurs propres définitions.

Par *responsable de l'installation*, l'on entend soit l'utilisateur root, soit des utilisateurs non root. En tant que responsable de l'installation, vous pouvez octroyer aux utilisateurs non root des autorisations de création de profils et d'établissement de leurs propres environnements de produit. Par exemple, un utilisateur non root pourra créer un environnement de produit afin de tester un déploiement d'application avec un profil dont il est le propriétaire. Les autorisations de création de profils accordées aux utilisateurs non root comprennent les éléments suivants :

- création d'un profil et attribution de la propriété du répertoire du profil à un utilisateur non root pour lui permettre de démarrer WebSphere Application Server pour un profil spécifique
- octroi de droits d'accès en écriture aux fichiers et répertoires appropriés à un utilisateur non root pour lui permettre de créer le profil. Cette tâche permet de créer un groupe d'utilisateurs autorisés à créer des profils ou d'accorder à des utilisateurs individuels la possibilité de créer des profils
- installation de packages de maintenance pour le produit, ce qui inclut les services requis pour les profils existants appartenant à un utilisateur non root. En tant que responsable de l'installation, c'est vous le propriétaire de tous les fichiers créés par ce package de maintenance

Pour plus d'informations sur la création de profils pour les utilisateurs non root, reportez-vous à la rubrique Création de profils pour les utilisateurs non root.

En tant que responsable de l'installation, vous pouvez également octroyer aux utilisateurs non root des autorisations d'extension de profils. Par exemple, un utilisateur non root peut étendre un profil créé par un responsable d'installation ou par lui-même. Suivez les extensions de profils réalisées par des utilisateurs non root de WebSphere Application Server Network Deployment.

Toutefois, lorsqu'un utilisateur non root étend un profil créé par le responsable de l'installation, il n'a pas besoin de créer auparavant les fichiers indiqué ci-après. Ces fichiers ont été créés en même temps que le profil.

- *racine_was*/logs/manageprofiles.xml

- *racine_was*/properties/fsdb.xml
- *racine_was*/properties/profileRegistry.xml

Lorsqu'un utilisateur non root étend un profil qu'il crée, il doit modifier les autorisations relatives aux documents situés dans les modèles de profils eXtreme Scale.

Avertissement : Vous pouvez également utiliser un profil non root (non administrateur) pour WebSphere eXtreme Scale dans un environnement autonome, extérieur à WebSphere Application Server. Vous devez remplacer le propriétaire du répertoire ObjectGrid par le profil non root. Vous pouvez ensuite ouvrir une session avec ce profil non root et utiliser eXtreme Scale comme vous le feriez avec un profil root (administrateur).

Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client autonomes

Vous pouvez installer WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client autonomes dans un environnement qui ne contient pas WebSphere Application Server ou WebSphere Application Server Network Deployment.

Avant de commencer

- Vérifiez que le répertoire d'installation cible est vide ou qu'il n'existe pas.

  Important : Si une version précédente d'WebSphere eXtreme Scale ou le composant ObjectGrid se trouve dans le répertoire que vous spécifiez pour installer la version 7.1.1, le produit n'est pas installé. Par exemple, vous pouvez disposer déjà d'un dossier *racine_install_wxs*/ObjectGrid. Vous pouvez sélectionner un autre répertoire d'installation ou annuler l'installation. Ensuite, désinstallez l'installation précédente et exécutez de nouveau l'assistant.
- Un environnement d'exécution IBM Runtime Environment est installé avec l'installation autonome dans le dossier *racine_install_wxs*/java.
- Si vous installez uniquement le client : téléchargez WebSphere eXtreme Scale Client pour la plateforme appropriée depuis le site Support .

Pourquoi et quand exécuter cette tâche

Lorsque vous installez le produit comme produit autonome, vous installez le client et le serveur WebSphere eXtreme Scale simultanément. Avec l'installation de WebSphere eXtreme Scale Client en mode autonome, vous installez un client pour pouvoir accéder aux données de vos grilles. Par conséquent, les processus serveur et client accèdent à toutes les ressources requises en local. Vous pouvez également imbriquer WebSphere eXtreme Scale dans des applications Java Platform, Standard Edition (J2SE) à l'aide de scripts et de fichiers d'archives Java (JAR).

Avertissement : Vous pouvez également utiliser un profil non root (non administrateur) pour WebSphere eXtreme Scale dans un environnement autonome. Pour utiliser un profil non root, vous devez remplacer le propriétaire du répertoire ObjectGrid par le profil non root. Vous pouvez alors vous connecter avec ce profil non root et utiliser eXtreme Scale comme vous le feriez avec un profil root (administrateur).

Procédure

1. Utilisez l'assistant pour installer le serveur et le client depuis le DVD.

- Exécutez le script suivant pour démarrer l'installation intégrale de WebSphere eXtreme Scale :
  - Linux UNIX *dvd_root*/install
  - Windows *dvd_root*\install.bat
- Exécutez le script suivant pour démarrer l'assistant de l'installation de WebSphere eXtreme Scale Client. Les fichiers d'installation se trouvent dans le fichier zip que vous téléchargez depuis la section des téléchargements sur le site Support :
  - Linux UNIX root/WXS_Client/install
  - Windows root\WXS_Client\install.bat

Avertissement : Si vous utilisez les conventions d'attribution de nom uniforme (UNC) pour identifier les chemins des fichiers dans la commande d'installation, les éléments que vous envisagez d'installer peuvent ne pas être installés à la fin de l'exécution de la commande. Pour éviter ce problème, mappez le chemin des fichiers à une unité de réseau. Exécutez la commande **install** sur l'unité mappée. L'utilisation d'une unité réseau mappée permet d'installer tous les éléments.

2. Suivez les invites de l'assistant et cliquez sur **Terminer**.

Restriction : Le panneau des fonctions dispositives répertorie les fonctions que vous pouvez choisir d'installer. Toutefois, des fonctions ne peuvent pas être ajoutées de manière incrémentielle à l'environnement du produit une fois que le produit a été installé. Si vous choisissez de ne pas installer une fonction lors de l'installation initiale du produit, vous devez désinstaller, puis réinstaller le produit pour l'ajouter.

Résultats

Windows Si vous installez WebSphere eXtreme Scale Client sur Windows, le texte suivant peut s'afficher dans les résultats de l'installation :

```
Success: The installation of the following product was successful:
WebSphere eXtreme Scale Client.  Some configuration steps have errors.
For more information, refer to the following log file:
<WebSphere Application Server install root>\logs\wxs_client\install\log.txt"
Review the installation log (log.txt) and review the deployment manager
augmentation log.
```

S'il existe une erreur associée au fichier `iscdeploy.sh`, vous pouvez l'ignorez. Cette erreur ne pose pas de problème.

Que faire ensuite

- Vérifiez l'installation. Pour plus d'informations, voir «Vérification de l'installation», à la page 206.
- Commencez à configurer votre installation WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client installation. Pour plus d'informations, voir «Premières étapes après l'installation», à la page 208.

Fichiers d'exécution de l'installation autonome WebSphere eXtreme Scale

Des fichiers JAR (Java) sont inclus dans l'installation. Vous pouvez voir les fichiers JAR qui sont inclus et leur emplacement d'installation.

Tableau 7. Fichiers d'exécution pour une installation complète de WebSphere eXtreme Scale. WebSphere eXtreme Scale s'appuie sur les processus ObjectGrid et les API associées. Le tableau ci-après répertorie les fichiers JAR inclus dans l'installation. L'emplacement d'installation est relatif au répertoire *rép_base_wxs* que vous choisissez lors de l'installation.

Nom de fichier	Environnement	Emplacement de l'installation	Description
wxsdynacache.jar	Client et serveur	dynacache/lib	Le fichier wxsdynacache.jar contient les classes nécessaires à utiliser avec le fournisseur de cache dynamique. Le fichier est automatiquement inclus dans l'environnement d'exécution du serveur lorsque vous utilisez les scripts fournis.
wxshyperic.jar	Utilitaire	hyperic/lib	Le plug-in de détection des serveurs WebSphere eXtreme Scale pour l'agent de surveillance SpringSource Hyperic.
objectgrid.jar	Local, client et serveur	lib	Le fichier objectgrid.jar est un ensemble OSGi qui est utilisé par l'environnement d'exécution du serveur de Java SE Version 5,0 et versions ultérieures. Le fichier est automatiquement inclus dans l'environnement d'exécution du serveur lorsque vous utilisez les scripts fournis.
ogagent.jar	Local, client et serveur	lib	Le fichier ogagent.jar contient les classes d'exécution requises pour exécuter l'agent d'instrumentation Java utilisé avec l'API EntityManager.
ogclient.jar	Local et client	lib	Le fichier ogclient.jar est un ensemble OSGi qui ne contient que les environnements d'exécution local et client. Vous pouvez utiliser ce fichier avec Java SE 5.0 et les versions suivantes.
ogspring.jar	Local, client et serveur	lib	Le fichier ogspring.jar contient les classes de support pour l'intégration de l'infrastructure SpringSource Spring.
wsogclient.jar	Local et client	lib	Fichier wsogclient.jar installé si vous utilisez un environnement qui contient WebSphere Application Server Version 6.0.2 et versions ultérieures. Ce fichier ne contient que les environnements d'exécution local et client.
wxssizeagent.jar	Local, client et serveur	lib	Le fichier wxssizeagent.jar est utilisé pour fournir des informations de dimensionnement d'entrée de cache plus précises lors de l'utilisation de l'environnement Java (JRE) Version 1.5 et versions suivantes.
ibmcfw.jar ibmorb.jar ibmorbapi.jar	Client et serveur	lib/endorsed	Cet ensemble de fichiers comprend l'exécution ORB (Object Request Broker) utilisée pour exécuter les applications dans les processus Java SE.
restservice.ear	Client	restservice/lib	Le fichier restservice.ear contient l'archive d'application d'entreprise de service de données eXtreme Scale REST pour les environnements WebSphere Application Server.
restservice.war	Client	restservice/lib	Le fichier restservice.war contient l'archive Web de service de données eXtreme Scale REST pour les serveurs d'applications provenant d'un autre fournisseur.
xsadmin.jar	Utilitaire	samples	Le fichier xsadmin.jar contient l'exemple d'utilitaire d'administration eXtreme Scale.
sessionobjectgrid.jar	Client et serveur	session/lib	Le fichier sessionobjectgrid.jar contient l'exécution de la gestion de sessions HTTP eXtreme Scale.
splicerlistener.jar	Utilitaire	session/lib	Le fichier splicerlistener.jar contient l'utilitaire splicer du programme d'écoute de session HTTP eXtreme Scale Version 7.1 et versions suivantes.
xsgbean.jar	Serveur	wasce/lib	Le fichier xsgbean.jar contient le GBean pour l'intégration de serveurs eXtreme Scale dans des serveurs d'applications WebSphere Application Server Community Edition.
splicer.jar	Utilitaire	legacy/session/lib	Utilitaire splicer pour le filtre de gestionnaire de session HTTP WebSphere eXtreme Scale Version 7.0.

Tableau 8. Fichiers d'exécution pour WebSphere eXtreme Scale Client. WebSphere eXtreme Scale Client s'appuie sur les processus ObjectGrid et les API associées. Le tableau ci-après répertorie les fichiers JAR inclus dans l'installation. L'emplacement d'installation est relatif au répertoire *rép_base_wxs* que vous choisissez lors de l'installation.

Nom de fichier	Environnement	Emplacement de l'installation	Description
`wxsdynacache.jar`	Client et serveur	`dynacache/lib`	Le fichier `wxsdynacache.jar` contient les classes nécessaires à utiliser avec le fournisseur de cache dynamique. Le fichier est automatiquement inclus dans l'environnement d'exécution du serveur lorsque vous utilisez les scripts fournis.
`wxshyperic.jar`	Utilitaire	`hyperic/lib`	Le plug-in de détection des serveurs WebSphere eXtreme Scale pour l'agent de surveillance SpringSource Hyperic.
`ogagent.jar`	Local, client et serveur	`lib`	Le fichier `ogagent.jar` contient les classes d'exécution requises pour exécuter l'agent d'instrumentation Java utilisé avec l'API EntityManager.
`ogclient.jar`	Local et client	`lib`	Le fichier `ogclient.jar` est un ensemble OSGi qui ne contient que les environnements d'exécution local et client. Vous pouvez l'utiliser avec Java SE 5 et les versions suivantes.
`ogspring.jar`	Local, client et serveur	`lib`	Le fichier `ogspring.jar` contient les classes de support pour l'intégration de l'infrastructure SpringSource Spring.
`wsogclient.jar`	Local et client	`lib`	Fichier `wsogclient.jar` installé si vous utilisez un environnement qui contient WebSphere Application Server Version 6.0.2 et versions ultérieures. Ce fichier ne contient que les environnements d'exécution local et client.
`wxssizeagent.jar`	Local, client et serveur	`lib`	Le fichier `wxssizeagent.jar` est utilisé pour fournir des informations de taille d'entrée de cache plus précise lorsque vous utilisez l'environnement d'exécution Java (JRE) Version 1.5 ou une version suivante.

`ibmcfw.jar`
`ibmorb.jar`
`ibmorbapi.jar`
	Client et serveur	`lib/endorsed`	Cet ensemble de fichiers comprend l'exécution ORB (Object Request Broker) utilisée pour exécuter les applications dans les processus Java SE.

`restservice.ear`	Client	`restservice/lib`	Le fichier `restservice.ear` contient l'archive d'application d'entreprise de service de données eXtreme Scale REST pour les environnements WebSphere Application Server.
`restservice.war`	Client	`restservice/lib`	Le fichier `restservice.war` contient l'archive Web de service de données eXtreme Scale REST pour les serveurs d'applications provenant d'un autre fournisseur.
`xsadmin.jar`	Utilitaire	`samples`	Le fichier `xsadmin.jar` contient l'exemple d'utilitaire d'administration eXtreme Scale.
`sessionobjectgrid.jar`	Client et serveur	`session/lib`	Le fichier `sessionobjectgrid.jar` contient l'exécution de la gestion de sessions HTTP eXtreme Scale.
`splicerlistener.jar`	Utilitaire	`session/lib`	Le fichier `splicerlistener.jar` contient l'utilitaire splicer du programme d'écoute de session HTTP eXtreme Scale Version 7.1 et versions suivantes.
`splicer.jar`	Utilitaire	`legacy/session/lib`	Utilitaire splicer du filtre de gestionnaire de session HTTP WebSphere eXtreme Scale Version 7.0.

Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client en mode silencieux

Utilisez un fichier de réponses qualifié complet, que vous configurez spécifiquement en fonction de vos besoins, ou transmettez les paramètres à la ligne de commande pour effectuer une installation de WebSphere eXtreme Scale or WebSphere eXtreme Scale Client en mode silencieux.

Avant de commencer

- Arrêtez tous les processus en cours d'exécution dans votre environnement WebSphere Application Server ou WebSphere Application Server Network Deployment. Voir Utilitaires de ligne de commande pour plus d'informations sur les commandes **`stopManager`**, **`stopNode`** et **`stopServer`**.

ATTENTION :
Vérifiez que les processus en cours d'exécution sont arrêtés. Si les processus en cours d'exécution ne sont pas arrêtés, l'installation se poursuit et crée des résultats imprévisibles, ce qui la laisse dans un état indéterminé sur certaines plateformes.

- Vérifiez que le répertoire d'installation cible est vide ou qu'il n'existe pas.

  Important : Si une version précédente d'WebSphere eXtreme Scale ou le composant ObjectGrid se trouve dans le répertoire que vous spécifiez pour installer la version 7.1.1, le produit n'est pas installé. Par exemple, vous pouvez disposer déjà d'un dossier *racine_install_wxs*`/ObjectGrid`. Vous pouvez sélectionner un autre répertoire d'installation ou annuler l'installation. Ensuite, désinstallez l'installation précédente et exécutez de nouveau l'assistant.

Pourquoi et quand exécuter cette tâche

Une installation en mode silencieux utilise le même programme d'installation que l'interface graphique. Toutefois, au lieu d'afficher une interface d'assistant, elle lit toutes vos réponses dans un fichier que vous personnalisez ou dans les paramètres que vous transmettez à la ligne de commande. Consultez un exemple de «Fichier `wxssetup.response.txt`», à la page 176 qui comprend une description de chaque option.

Procédure

1. Facultatif : Si vous choisissez d'installer WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client à l'aide d'un fichier de réponses, commencez par personnaliser le fichier `wxssetup.response.txt`.

   A faire : Vous devez spécifier le nom complet du fichier de réponses. Si vous spécifiez le chemin relatif, l'installation échoue sans message d'erreur.

   a. Créez une copie du fichier de réponses à personnaliser.

      Dans le cas d'une installation complète de WebSphere eXtreme Scale, copiez le fichier de réponses sur votre disque dur à partir du DVD du produit.

      Pour le WebSphere eXtreme Scale Client, décompressez le WebSphere eXtreme Scale Client sur votre disque dur et cherchez le fichier de réponses.

   b. Ouvrez et éditez le fichier de réponses dans l'éditeur de texte de votre choix. L'exemple de fichier de réponses précédent fournit des détails sur la manière de spécifier chacun des paramètres. Vous devez spécifier les paramètres suivants :

      - Contrat de licence
      - Répertoire d'installation

      Conseil : Lorsque vous installez WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client dans un environnement WebSphere Application Server, le programme d'installation utilise le répertoire d'installation pour déterminer où se trouve installée l'instance WebSphere Application Server existante. Si vous effectuez l'installation sur un noeud qui contient plusieurs instances WebSphere Application Server, définissez précisément votre emplacement.

   c. Exécutez le script suivant pour démarrer l'installation.

      Pour l'installation complète de WebSphere eXtreme Scale :

      ```
      ./install.sh|bat -options C:/chemin_unité/fichier_réponses.txt -silent
      ```

      Pour l'installation de WebSphere eXtreme Scale Client :

```
./WXS_Client/install.sh|bat -options C:/chemin_unité/fichier_réponses.txt -silent
```

Vous pouvez également utiliser le fichier de réponses lorsque vous exécutez une installation à l'aide de l'interface graphique. Ce fichier de réponses peut vous servir alors à déboguer les problèmes qui sont masqués dans l'installation en mode silencieux. Lorsque vous spécifiez le fichier `wxssetup.response` pour des installations à l'aide de l'interface graphique ou en mode silencieux, vous devez utiliser le chemin qualifié complet. Exécutez le script suivant pour une installation à l'aide de l'interface graphique avec votre fichier de réponses :

- Linux UNIX `<base_install>/install.sh -options <chemin_complet_install_requis>/wxssetup.response`
- Windows `<base_install>\install.exe -options c:\<chemin_complet_install_requis>\wxssetup.response`

2. Facultatif : Si vous choisissez d'installer eXtreme Scale en transmettant certains paramètres à la ligne de commande, exécutez le script suivant pour démarrer l'installation :

   Pour l'installation complète de WebSphere eXtreme Scale :

   ```
   ./install.sh|bat -silent -OPT silentInstallLicenseAcceptance=true -OPT installLocation=emplacement_install
   ```

   Pour l'installation de WebSphere eXtreme Scale Client :

   ```
   ./WXS_Client/install.sh|bat -silent -OPT silentInstallLicenseAcceptance=true -OPT installLocation=emplacement_install
   ```

Fichier de réponses d'une installation en mode silencieux

Spécifiez des paramètres sur la ligne de commande pour personnaliser et configurer l'installation de votre produit.

Remarque : Vous devez spécifier le nom complet du fichier de réponses. Si vous spécifiez le chemin relatif, l'installation échoue sans message d'erreur.

Paramètres

Vous pouvez transmettre les paramètres suivants lors d'une installation du produit à l'aide de la ligne de commande ou du fichier d'options :

-silent
Supprime l'interface graphique. Spécifiez le paramètre **-options** pour indiquer que le programme d'installation effectue l'installation en fonction d'un fichier d'options personnalisé. Si vous ne spécifiez pas le paramètre **-options**, les valeurs par défaut sont utilisées.

Exemple de syntaxe

```
./install.sh|bat
-silent -options fichier_options.txt
```

-options *nom_chemin/nom_fichier*
Indique un fichier d'options utilisé par le programme d'installation pour effectuer une installation en mode silencieux. Les propriétés sur la ligne de commande sont prioritaires.

Exemple de syntaxe

```
./install.sh|bat -options c:/nom_chemin/fichier_options.txt
```

-log # !file_name @*type_événement*
Génère un fichier journal d'installation qui consigne les types d'événement suivants :

- err

- wrn
- msg1
- msg2
- dbg
- ALL

Exemple de syntaxe

```
./install.sh|bat -log # !c:/temp/logfiles.txt @ALL
```

-is:log *nom_chemin/nom_fichier*
Crée un fichier journal qui contient les recherches JVM (Java Virtual Machine) du programme d'installation lors de la tentative de démarrage de l'interface graphique. Le fichier journal n'est pas créé s'il n'est pas spécifié.

Exemple de syntaxe

```
./install.sh|bat -is:log c:/logs/javalog.txt
```

-is:javaconsole
Affiche une fenêtre de console lors de la procédure d'installation.

Exemple de syntaxe

```
./install.sh|bat -is:javaconsole
```

-is:silent
Supprime la fenêtre d'initialisation Java affichée au démarrage du programme d'installation.

Exemple de syntaxe

```
./install.sh|bat -is:silent
```

-is:tempdir *nom_chemin*
Indique le répertoire temporaire utilisé par le programme d'installation lors de l'installation.

Exemple de syntaxe

```
./install.sh|bat -is:tempdir c:/temp
```

Installation du service de données REST

Nous allons expliquer comment installer sur un serveur Web le service de données REST d'WebSphere eXtreme Scale.

Avant de commencer

Configuration logicielle

Le service de données REST d'WebSphere eXtreme Scale est une application Web Java qui peut être déployée sur tout serveur d'applications prenant en charge la spécification de servlet Java version 2.3 et un environnement d'exécution Java version 5 ou plus récente.

Les logiciels suivants sont requis :

- Java Standard Edition 5 ou version ultérieure
- le conteneur de servlets Web version 2.3 ou ultérieure, qui inclut l'un des éléments suivants :
  - le serveur d'applications WebSphere version 6.1.0.25 ou ultérieure
  - le serveur d'applications WebSphere version 7.0.0.5 ou ultérieure

  - WebSphere Community Edition version 2.1.1.3 ou ultérieure
  - Apache Tomcat version 5.5 ou ultérieure
- WebSphere eXtreme Scale, Version 7.1 ou ultérieure, y compris la version d'évaluation.

Pourquoi et quand exécuter cette tâche

Le service de données REST WebSphere eXtreme Scale inclut un seul fichier `wxsrestservice.war`. Le fichier `wxsrestservice.war` comporte un seul servlet qui fait office de passerelle entre vos applications client WCF Data Services ou tout autre client REST HTTP et une grille de données.

Le service de données REST contient un exemple qui permet de créer rapidement une grille de données et d'interagir avec elle en utilisant un client eXtreme Scale ou le service de données REST. Voir «Configuration des services de données REST», à la page 357 pour plus d'informations sur l'utilisation de l'exemple.

Lors de l'installation de WebSphere eXtreme Scale 7.1 ou de l'extraction de la version d'évaluation d'eXtreme Scale version 7.1, les répertoires et fichiers suivants sont inclus :

- `base_servicerest/lib`

  Le répertoire lib contient ces fichiers :
  - `wxsrestservice.ear` – L'archive d'application d'entreprise de service de données REST à utiliser avec le serveur d'application WebSphere et le serveur d'application CE WebSphere.
  - `wxsrestservice.war` – Le module Web de service de données REST à utiliser avec Apache Tomcat.

  Le fichier `wxsrestservice.ear` inclut le fichier `wxsrestservice.war` et tous deux sont étroitement couplés à l'environnement d'exécution WebSphere eXtreme Scale. En cas de mise à niveau d'WebSphere eXtreme Scale vers une nouvelle version ou si un groupe de correctifs est appliqué, les fichiers `wxsrestservice.war` file ou `wxsrestservice.ear` devront être mis à niveau manuellement vers la version installée dans ce répertoire.
- `base_servicerest/gettingstarted`

  Le répertoire `gettingstarted` contient un exemple simple montrant comment utiliser le service de données REST WebSphere eXtreme Scale avec une grille de données.

Procédure

Packagez et déployez le service de données REST.
Le service de données REST a été conçu en tant que module WAR autonome. Pour configurer le service de données REST, vous devez commencer par packager dans un fichier JAR ou dans un répertoire la configuration du service de données REST et les éventuels fichiers de configuration d'WebSphere eXtreme Scale. Ce package d'application est ensuite référencé par l'environnement d'exécution du serveur de conteneur Web. La figure suivante illustre les fichiers utilisés par le service de données REST d'eXtreme Scale.

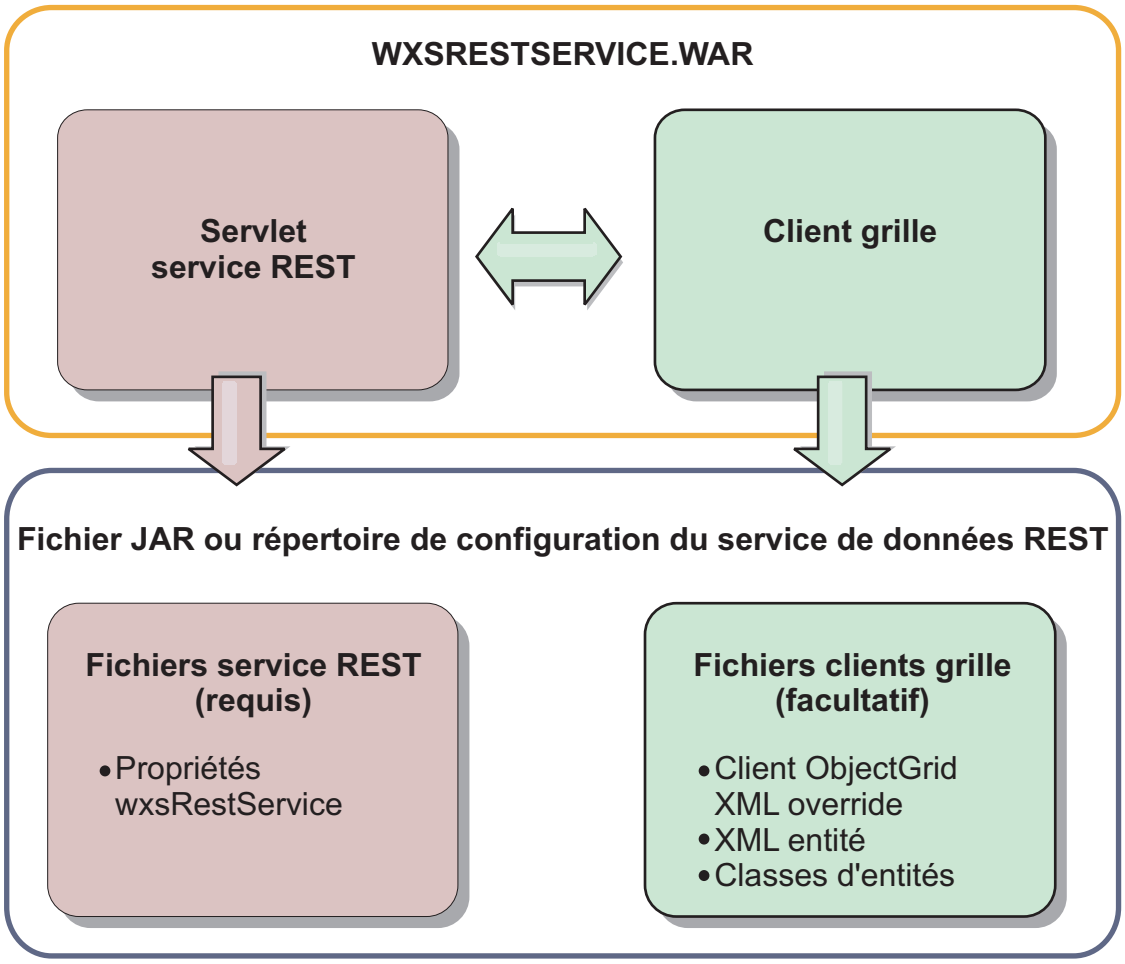


Figure 26. Fichiers du service de données REST d'WebSphere eXtreme Scale

Le fichier JAR de configuration du service REST ou le répertoire doit contenir le fichier suivant :
`wxsRestService.properties` : le fichier `wxsRestService.properties` comprend les options de configuration du service de données REST : points de contact du service de catalogue, noms d'ObjectGrid à exposer, options de suivi, etc. Voir Fichier de propriétés du service de données REST.
Les fichiers suivants du client ObjectGrid sont facultatifs :

- `META-INF/objectGridClient.xml` : le fichier XML de substitution de client ObjectGrid est utilisé pour se connecter à la grille de données distante. Par défaut, ce fichier n'est pas requis. En son absence, le service REST utilise la configuration du serveur en désactivant le cache proche.

  Le nom du fichier peut être remplacé à l'aide de la propriété de configuration du service de données REST objectGridClientXML. S'il est fourni, ce fichier XML doit inclure :

  1. Les ObjectGrids à exposer au service de données REST.
  2. Toute référence au fichier XML descripteur d'entité associé à chaque configuration ObjectGrid.
- `META-INF/`*`fichiers XML de descripteurs d'entités`* : un ou plusieurs fichiers XML de descripteurs d'entités ne sont requis que si le client doit remplacer la définition d'entité du client. Le fichier XML du descripteur d'entité doit être utilisé avec le fichier XML du descripteur d'entité de remplacement ObjectGrid par les clients.

- **Classes entité** Vous pouvez utiliser des classes entité annotées ou un fichier XML descripteur d'entité pour décrire les métadonnées d'entité. Le service REST ne nécessite les classes d'entités dans le chemin d'accès aux classes que si les serveurs eXtreme Scale sont configurés avec les classes de métadonnées d'entités. Aucun fichier XML de descripteur d'entité de remplacement par le client n'est utilisé.

  Voici un exemple avec le fichier de configuration minimum requise, où les entités sont définies dans XML sur les serveurs :

  ```
  restserviceconfig.jar:
  wxsRestService.properties
  ```

  Le fichier de propriétés contient :

  ```
  catalogServiceEndPoints=localhost:2809
  objectGridNames=NorthwindGrid
  ```

  Un exemple avec une entité, des fichiers XML de remplacement et des classes entité :

  ```
  restserviceconfig.jar:
  wxsRestService.properties
  ```

  Le fichier de propriétés contient :

  ```
  catalogServiceEndPoints=localhost:2809
  objectGridNames=NorthwindGrid
  ```

  ```
  com/acme/entities/Customer.class
  META-INF/objectGridClient.xml
  ```

  Le fichier XML descripteur du client ObjectGrid contient :

  ```
  <objectGrid name="CustomerGrid" entityMetadataXMLFile="emd.xml"/>
  META-INF/emd.xml
  ```

  Le fichier XML descripteur des métadonnées d'entité contient :

  ```
  <entity class-name="com.acme.entities.Customer" name="Customer"/>
  ```

Installation de l'infrastructure OSGi Eclipse Equinox avec Eclipse Gemini pour les clients et les serveurs

Si vous souhaitez déployer WebSphere eXtreme Scale dans la structure OSGi, vous devez configurer l'environnement Eclipse Equinox.

Pourquoi et quand exécuter cette tâche

La tâche nécessite que vous téléchargiez et installiez l'infrastructure Blueprint qui permet de configurer ensuite les JavaBeans et de les exposer en tant que services. L'utilisation de services est importante, car vous pouvez exposer des plug-ins en tant que services OSGi pour qu'ils puissent être utilisés par l'environnement d'exécution eXtreme Scale. Le produit prend en charge deux conteneurs Blueprint dans l'infrastructure OSGi principale Eclipse Gemini et Apache Aries. Utilisez cette procédure pour configurer le conteneur Gemini Eclipse.

Procédure

1. Téléchargez Eclipse Equinox SDK Version 3.6.1 ou la version suivante à partir du site Web Eclipse. Créez un répertoire pour l'infrastructure Equinox, par exemple, `/opt/equinox`. Ces instructions font référence à ce répertoire sous la forme `equinox_root`. Extrayez le fichier compressé dans le répertoire `equinox_root`.
2. Téléchargez le fichier compressé gemini-plan d'incubation 1.0.0 depuis le site Web Eclipse. Extrayez le contenu du fichier dans un répertoire temporaire et copiez les fichiers extraits suivants vers le répertoire `equinox_root/plugins` :

```
dist/gemini-blueprint-core-1.0.0.jar
dist/gemini-blueprint-extender-1.0.0.jar
dist/gemini-blueprint-io-1.0.0.jar
```

3. Téléchargez Spring Framework Version 3.0.5 à partir de la page Web SpringSource http://www.springsource.com/download/community. Extrayez le contenu du fichier dans un répertoire temporaire et copiez les fichiers extraits suivants vers le répertoire `equinox_root/plugins` :

   ```
   org.springframework.aop-3.0.5.RELEASE.jar
   org.springframework.asm-3.0.5.RELEASE.jar
   org.springframework.beans-3.0.5.RELEASE.jar
   org.springframework.context-3.0.5.RELEASE.jar
   org.springframework.core-3.0.5.RELEASE.jar
   org.springframework.expression-3.0.5.RELEASE.jar
   ```

4. Téléchargez le fichier AOP Alliance Java archive (JAR) depuis la page Web SpringSource. Copiez com.springsource.org.aopalliance-1.0.0.jar vers le répertoire `equinox_root/plugins` .
5. Téléchargez le fichier JAR Apache commons logging 1.1.1 JAR depuis la page Web SpringSource. Copiez le fichier `com.springsource.org.apache.commons.logging-1.1.1.jar` vers le répertoire `equinox_root/plugins`.
6. Téléchargez le client de ligne de commande Luminis OSGi Configuration Admin. Utilisez cet ensemble pour gérer les configurations d'administration OSGi. Vous pouvez télécharger le fichier JAR depuis la page Web https://opensource.luminis.net/wiki/display/SITE/OSGi+Configuration+Admin+command+line+client. Copiez le fichier `net.luminis.cmc-0.2.5.jar` vers le répertoire `equinox_root/plugins`.
7. Téléchargez l'ensemble Apache Felix file installation Version 3.0.2 depuis la page Web http://felix.apache.org/site/index.html. Copiez le fichier `org.apache.felix.fileinstall-3.0.2.jar` vers le répertoire `equinox_root/plugins`.
8. Créez un répertoire de configuration dans le répertoire `equinox_root/plugins`, par exemple :

   ```
   mkdir equinox_root/plugins/configuration
   ```

9. Créez le fichier `config.ini` suivant dans le répertoire `equinox_root/plugins/configuration` en remplaçant *equinox_root* par le chemin absolu dans le chemin du répertoire `equinox_root` en supprimant tous les espaces après la barre oblique inverse dans chaque ligne. Vous devez placer une ligne blanche à la fin du fichier, par exemple :

   ```
   osgi.noShutdown=true
   osgi.java.profile.bootdelegation=none
   org.osgi.framework.bootdelegation=none
   eclipse.ignoreApp=true
   osgi.bundles=\
   org.eclipse.osgi.services_3.2.100.v20100503.jar@1:start, \
   org.eclipse.osgi.util_3.2.100.v20100503.jar@1:start, \
   org.eclipse.equinox.cm_1.0.200.v20100520.jar@1:start, \
   com.springsource.org.apache.commons.logging-1.1.1.jar@1:start, \
   com.springsource.org.aopalliance-1.0.0.jar@1:start, \
   org.springframework.aop-3.0.5.RELEASE.jar@1:start, \
   org.springframework.asm-3.0.5.RELEASE.jar@1:start, \
   org.springframework.beans-3.0.5.RELEASE.jar@1:start, \
   org.springframework.context-3.0.5.RELEASE.jar@1:start, \
   org.springframework.core-3.0.5.RELEASE.jar@1:start, \
   org.springframework.expression-3.0.5.RELEASE.jar@1:start, \
   org.apache.felix.fileinstall-3.0.2.jar@1:start, \
   net.luminis.cmc-0.2.5.jar@1:start, \
   gemini-blueprint-core-1.0.0.jar@1:start, \
   gemini-blueprint-extender-1.0.0.jar@1:start, \
   gemini-blueprint-io-1.0.0.jar@1:start
   ```

   Si vous avez déjà configuré l'environnement, vous pouvez nettoyer le référentiel de plug-in Equinox en supprimant le répertoire `equinox_root\plugins\configuration\org.eclipse.osgi`.
10. Exécutez la commande suivante pour démarrer la console Equinox.

Si vous exécutez une version différente d'Equinox, le nom du fichier JAR est différent de celui de l'exemple ci-dessous :

```
java -jar plugins\org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
```

Installation des ensembles eXtreme Scale

WebSphere eXtreme Scale inclut des ensembles qui peuvent être installés dans une infrastructure OSGi Eclipse Equinox. Ces ensembles sont nécessaires pour démarrer les serveurs eXtreme Scale ou utiliser les clients eXtreme Scale dans OSGi.

Avant de commencer

Cette tâche suppose que les produits suivants ont été installés :

- Infrastructure OSGi Eclipse Equinox
- Client ou serveur autonome eXtreme Scale

Pourquoi et quand exécuter cette tâche

eXtreme Scale inclut deux ensembles. Un seul des ensembles suivants est nécessaire dans une infrastructure OSGi :

objectgrid.jar
L'ensemble de serveur est le fichier `objectgrid.jar`. Il est installé avec l'installation de serveur autonome eXtreme Scale et il est nécessaire pour exécuter les serveurs eXtreme Scale servers. Il peut être aussi utilisé pour exécuter les clients eXtreme Scale ou les mémoires caches internes locales. L'ID d'ensemble du fichier `objectgrid.jar` est com.ibm.websphere.xs.server_`<version>`, où la version a le format `<Version>.<Release>.<Modification>`. Par exemple, l'ensemble de serveur pour eXtreme Scale version 7.1.1 est com.ibm.websphere.xs.server_7.1.1.

ogclient.jar
L'ensemble `ogclient.jar` est installé avec les installations client et autonomes eXtreme Scale et il est utilisé pour exécuter les clients eXtreme Scale ou les mémoires caches internes locales. L'ID d'ensemble du fichier `ogclient.jar` est com.ibm.websphere.xs.client_`<version>`, où la version a le format `<Version>_<Release>_<Modification`. Par exemple, l'ensemble client pour eXtreme Scale Version 7.1.1 est com.ibm.websphere.xs.client_7.1.1.

Pour plus d'informations sur le développement de plug-ins eXtreme Scale, voir la rubrique API système et plug-ins.

Procédure

Pour installer l'ensemble client ou serveur eXtreme Scale dans l'infrastructure OSGi Eclipse Equinox en utilisant la console OSGi :

1. Démarrez l'infrastructure Eclipse Equinox avec la console activée. Par exemple :
   rép_base_java`/bin/java -jar <equinox_root>/plugins/org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console`
2. Installez l'ensemble serveur ou client eXtreme Scale dans la console Equinox :
   `osgi> install file:///<path to bundle>`
3. Equinox affiche l'ID d'ensemble du nouvel ensemble installé :
   `Bundle id is 25`
4. Démarrez l'ensemble dans la console Equinox, où `<id>` est l'ID affecté à l'ensemble lors de son installation :

```
osgi>  start <id>
```

5. Extrayez l'état du service dans la console Equinox pour vérifier que l'ensemble a démarré. Par exemple :

   ```
   osgi> ss
   ```

   Lorsque l'ensemble a démarré correctement, il affiche l'état ACTIVE, par exemple :

   ```
   25      ACTIVE      com.ibm.websphere.xs.server_7.1.1
   ```

Installez l'ensemble client ou serveur eXtreme Scale dans l'infrastructure OSGi Eclipse Equinox en utilisant le fichier `config.ini` :

6. Copiez l'ensemble client ou serveur eXtreme Scale (objectgrid.jar ou ogclient.jar) de `<wxs_install_root>/ObjectGrid/lib` vers le répertoire Eclipse Equinox, par exemple : <equinox_root>/plugins
7. Modifiez le fichier de configuration Eclipse Equinox `config.ini` et ajoutez l'ensemble à la propriété osgi.bundles, par exemple :

   ```
   osgi.bundles=\
   org.eclipse.osgi.services_3.2.100.v20100503.jar@1:start, \
   org.eclipse.osgi.util_3.2.100.v20100503.jar@1:start, \
   org.eclipse.equinox.cm_1.0.200.v20100520.jar@1:start, \
   objectgrid.jar@1:start
   ```

   Important : Vérifiez qu'une ligne blanche existe après le dernier nom d'ensemble. Chaque ensemble est séparé par une virgule.
8. Démarrez l'infrastructure Eclipse Equinox avec la console activée. Par exemple :

   ```
   rép_base_java/bin/java -jar <equinox_root>/plugins/
   org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
   ```
9. Extrayez l'état du service dans la console Equinox pour vérifier que l'ensemble a démarré :

   ```
   osgi> ss
   ```

   Lorsque l'ensemble a démarré correctement, il affiche l'état ACTIVE, par exemple :

   ```
   25      ACTIVE      com.ibm.websphere.xs.server_7.1.1
   ```

Résultats

L'ensemble client ou serveur eXtreme Scale est installé et démarré dans l'infrastructure OSGi Eclipse Equinox.

Vérification de l'installation

Une fois l'assistant d'installation exécuté, vous pouvez vérifier l'installation en vérifiant plusieurs aspects de l'installation.

Procédure

- **Pour une installation autonome ou une installation qui est intégrée à WebSphere Application Server:**

  Utilisez l'une des méthodes suivantes pour vérifier que votre installation s'est terminée avec succès :

  – Exécutez la commande version info pour WebSphere eXtreme Scale :

  ```
  racine_was/lib/> java -jar wsobjectgrid.jar version
  ```

Le nom et le numéro de version du produit et le numéro de version s'affichent.

– Vérifiez les fichiers de propriétés pour le numéro de version approprié.
  - Fichiers de signature : les fichiers de signature se trouvent dans le répertoire *racine_was*/properties/version. Si un groupe de correctifs a été installé, d'autres fichiers fxtg sont également inclus. Voici quelques exemples de noms de fichier de signature :

    ```
    WebSphere_eXtreme_Scale.7.1.1..swtag
    WebSphere_eXtreme_Scale.7.1.0.2.fxtag
    WebSphere_eXtreme_Scale.7.1.0.3.fxtag
    ```

  - Fichier du produit WebSphere eXtreme Scale :

    Le fichier du produit se trouve dans le répertoire *racine_was*/properties/version. Recherchez le fichier WXS.product. Exemple de contenu dans ce fichier :

    ```
    <?xml version="1.0" encoding="UTF-8"?>
      <!DOCTYPE product SYSTEM "product.dtd">
      <product name="IBM WebSphere eXtreme Scale">
           <id>WXS</id>
           <version>7.1.1.0</version>
           <build-info
       date="8/5/11"
       level="a1132.68720"/>
      </product>
    ```

– Vérifiez que les fichiers d'exécution sont installés. Les fichiers d'exécution de chaque type d'installation sont décrits dans les sections suivantes :
  - «Fichiers d'exécution de l'installation autonome WebSphere eXtreme Scale», à la page 195
  - «Fichiers d'exécution pour WebSphere eXtreme Scale intégré à WebSphere Application Server», à la page 165

• **Pour une installation intégrée à WebSphere Application Server, vous disposez des méthodes supplémentaires suivantes pour vérifier que l'installation s'est déroulée avec succès :**

– Exécutez la commande version info pour WebSphere Application Server:

  ```
  racine_was/bin/> versionInfo.sh|.bat
  ```

  La sortie affiche la liste des produits installés, y compris les répertoires d'installation, les produits installés, les versions, le niveau de génération, la date de génération, etc.

  Conseil : Ajoutez le paramètre **-maintenancePackages** pour afficher des informations supplémentaires :

  ```
  racine_was/bin/> versionInfo.sh|.bat -maintenancePackages
  ```

– Vérifiez le panneau d'accueil de la console d'administration WebSphere Application Server. Accédez à http://localhost:9060/ibm/console. Connectez-vous à la console. La version de WebSphere eXtreme Scale affiche le panneau d'accueil.

– Utilisez la console Premiers pas pour décrire l'installation WebSphere Application Server avec WebSphere eXtreme Scale :

  ```
  racine_was/firststeps/WXS> firststeps.sh|.bat
  ```

  Pour plus d'informations, voir «Création et augmentation de profils pour WebSphere eXtreme Scale», à la page 184.

Que faire ensuite

Si vous constatez que l'installation ne s'est pas déroulée correctement, traitez les problèmes d'installation. Pour plus d'informations, voir «Traitement des problèmes d'installation».

Premières étapes après l'installation

Une fois l'installation terminée et vérifiée, vous pouvez commencer à utiliser WebSphere eXtreme Scale pour créer votre grille de données.

Procédure

1. Mettez à jour votre installation en appliquant la maintenance.

   Informations complémentaires : «Mise à jour des serveurs eXtreme Scale», à la page 211.
2. Si vous utilisez WebSphere eXtreme Scale pour la première fois, vous pouvez utiliser les informations du guide de démarrage pour en savoir plus sur l'utilisation du produit.

   Informations complémentaires : Chapitre 1, «Mise en route», à la page 1
3. Configurez le produit. Créez des propriétés et des fichiers XML pour définir la configuration pour les grilles de données, des serveurs et des clients. Vous pouvez également configurer l'intégration du cache ou de la base de données, les services de données REST ou les plug-ins OSGi.

   Informations complémentaires : Chapitre 6, «configuration», à la page 223
4. Développez une application qui accède à la grille de données.

   Informations complémentaires : Développement d'applications
5. Démarrez et administrez les serveurs de conteneur et de catalogue avec vos fichiers de configuration et les données d'application de la grille.

   Informations complémentaires : Chapitre 7, «Administration», à la page 395
6. Surveillez les performances de votre configuration à l'aide de divers outils de surveillance.

   Informations complémentaires : Chapitre 8, «Contrôle», à la page 441

Traitement des problèmes d'installation

Utilisez ces informations pour traiter les problèmes d'installation.

Procédure

- **Problème :** lorsque vous exécutez la commande d'installation à partir d'un ordinateur distant, tel que `\\mymachine\downloads\`, le message suivant s'affiche : `CMD.EXE was started with the above path as the current directory. UNC paths are not supported. Defaulting to Windows directory.` En conséquence, l'installation ne peut pas se terminer correctement.

  Solution : mappez l'ordinateur distant à une unité réseau. Par exemple, dans Windows, vous pouvez cliquer avec le bouton droit de la souris sur **Ordinateur**, choisir **Connecter un lecteur réseau** et inclure le chemin UNC (uniform naming conventions) vers l'ordinateur distant. Ensuite, vous pouvez exécuter le script d'installation depuis le lecteur réseau avec succès. Par exemple, `y:\mymachine\downloads\WXS\install.bat`.
- **Problème :** l'installation n'aboutit pas.

Solution : vérifiez les fichiers journaux pour savoir où l'installation a échoué. Lorsque l'installation n'aboutit pas, les fichiers journaux se trouvent dans le répertoire *racine_install_wxs*/logs/wxs.

- **Problème :** échec catastrophique lors de l'installation.

  Solution : vérifiez les fichiers journaux pour savoir où l'installation a échoué. Lorsque l'installation a été partiellement exécutée, le journaux se trouvent généralement dans le répertoire *user_root*/wxs_install_logs/.

- Windows **Problème :** si vous installez WebSphere eXtreme Scale Client sur Windows, le texte suivant peut s'afficher dans les résultats de l'installation :

  ```
  Success: The installation of the following product was successful:
  WebSphere eXtreme Scale Client.  Some configuration steps have errors.
  For more information, refer to the following log file:
  <WebSphere Application Server install root>\logs\wxs_client\install\log.txt"
  Review the installation log (log.txt) and review the deployment manager
  augmentation log.
  ```

  Solution : si vous identifiez une erreur avec le fichier iscdeploy.sh, vous pouvez l'ignorer. Cette erreur ne pose pas de problème.

Désinstallation de WebSphere eXtreme Scale

Pour supprimer WebSphere eXtreme Scale de votre environnement, vous pouvez utiliser l'assistant ou procéder à une désinstallation en mode silencieux.

Avant de commencer

Avertissement : Le programme de désinstallation supprime tous les fichiers binaires et la maintenance, telle que les groupes de correctifs et les correctifs temporaires, en même temps.

Procédure

1. Arrêtez tous les processus exécutant eXtreme Scale.

   ATTENTION :
   Assurez-vous que tous les processus en cours d'exécution sont bien arrêtés. Si les processus ne sont pas arrêtés, la désinstallation s'effectue, créant des résultats imprévisibles et laissant sur quelques plateformes la désinstallation dans un état indéterminé.

   - Si vous avez installé un serveur eXtreme Scale autonome, lisez la rubrique sur l'arrêt des serveurs autonomes pour arrêter les processus.
   - Si vous avez installé eXtreme Scale avec une installation existante de WebSphere Application Server, lisez le document sur les utilitaires de ligne de commande pour savoir comment arrêter les processus de WebSphere Application Server.
   - Si vous exécutez la console Web, utilisez la commande stopConsoleServer pour arrêter le serveur de la console Web. Le script stopConsoleServer se trouve dans le répertoire *racine_install_wxs*/ObjectGrid/bin. Si vous n'arrêtez pas le serveur avant d'exécuter la désinstallation, le processus est automatiquement arrêté lors du processus de désinstallation.

2. Désinstallez le produit. La désinstallation peut s'effectuer dans une interface graphique ou en mode silencieux.

   Remarque : Lorsqu'on spécifie le fichier de réponses wxssetup.response pour une désinstallation ou pour des installations dans l'interface graphique ou en

mode silencieux, le chemin qualifié complet du fichier doit être spécifié. Le fichier de réponses n'est pas obligatoire dans le cas d'une désinstallation à partir de l'interface graphique.

- **Pour exécuter la désinstallation à partir de l'interface graphique :**
  - Linux UNIX `<rép_install>/uninstall_wxs/uninstall`
  - Windows `<rép_install>\uninstall_wxs\uninstall.exe`

  Pour exécuter la désinstallation à partir de l'interface graphique et avec le fichier `wxssetup.response`, utilisez l'une des commandes suivantes :
  - Linux UNIX

    ```
    <rép_install>/uninstall_wxs/uninstall -options
    <chemin_complet_install_requis>/wxssetup.response
    ```
  - Windows

    ```
    <rép_install>\uninstall_wxs\uninstall.exe -options
    <chemin_complet_install_requis>\wxssetup.response
    ```
- **Pour exécuter la désinstallation en mode silencieux avec le script `wxssetup.response` de fichier de réponses :**
  - Linux UNIX

    ```
    <rép_install>/uninstall_wxs/uninstall -options
    <chemin_complet_install_requis>/wxssetup.response -silent
    ```
  - Windows

    ```
    <rép_install>\uninstall_wxs\uninstall.exe -options
    <chemin_complet_install_requis>\wxssetup.response -silent
    ```

Résultats

Vous avez supprimé eXtreme Scale de votre environnement.

Chapitre 5. Mise à niveau et migration de WebSphere eXtreme Scale

Vous pouvez migrer vers la version 7.1.1 à partir de versions précédentes et appliquer des packages de maintenance à la version 7.1.1. Pour éviter les indisponibilités, vous devez prendre en compte l'ordre dans lequel vous appliquez les mises à jour aux serveurs dans votre configuration.

- Pour mettre à niveau une installation version 7.1.0.x, voir «Mise à jour des serveurs eXtreme Scale» et «Utilisation du programme d'installation de mises à jour pour installer des modules de maintenance», à la page 215.
- Pour mettre à niveau une installation version 7.0.x, voir «Mise à jour des serveurs eXtreme Scale» et «Migration vers WebSphere eXtreme Scale Version 7.1.1», à la page 214.

Mise à jour des serveurs eXtreme Scale

Vous pouvez mettre à niveau WebSphere eXtreme Scale vers une nouvelle version, soit en appliquant la maintenance ou en installant une nouvelle version, sans interrompre le service.

Avant de commencer

Vous devez disposer du fichier binaire pour la version majeure ou de maintenance à appliquer. Vous pouvez obtenir les dernières informations sur les versions disponibles et les packages de maintenance depuis le portail du support IBM pour WebSphere eXtreme Scale.

Pourquoi et quand exécuter cette tâche

Pour mettre à niveau sans interruption du service, mettez à niveau préalablement les serveurs de catalogue. Ensuite, mettez à niveau les serveurs de conteneur et les clients.

Procédure

1. Mettez à jour le niveau de service de catalogue, en répétant les étapes suivantes pour chaque serveur de catalogue dans la grille de données. Mettez à jour le niveau de service de catalogue avant de mettre à niveau les serveurs de conteneur ou les clients. Les serveurs de catalogue individuels peuvent interopérer avec la compatibilité de version de sorte que vous pouvez appliquer des mises à niveau à un seul serveur de catalogue à la fois sans interrompre le service.
   a. Recherchez un état de quorum sain. Exécutez la commande suivante :

      ```
      xsadmin -quorumStatus
      xscmd -c showQuorumStatus
      ```

      Ce résultat indique que tous les serveurs de catalogue sont connectés.
   b. Si vous utilisez la réplication multimaître entre deux domaine de services de catalogue, supprimez la liaison entre les deux domaine de services de catalogue lorsque vous mettez à niveau les serveurs de catalogue.

      ```
      xsadmin –ch host -p 1099 -dismissLink domain_name
      ```

7.1.1+

```
xscmd –c dismissLink –cep host:2809 -fd domain_name
```

Il suffit d'exécuter cette commande depuis l'une des domaine de services de catalogue pour supprimer la liaison entre deux domaine de services de catalogue.

c. Arrêtez l'un des serveurs de catalogue. Vous pouvez utiliser la commande **stopOgserver**, la commande **xscmd -c teardown** ou arrêtez le serveur d'application qui exécute le service de catalogue dans WebSphere Application Server. Il n'existe pas d'ordre d'arrêt spécifique des serveurs de catalogue, mais l'arrêt du serveur de catalogue principal en dernier réduit la rotation. Pour identifier le serveur de catalogue principal, recherchez le message CWOBJ8106 dans les fichiers journaux. Dans des conditions normales, le quorum est maintenu lorsqu'un serveur de catalogue est arrêté, mais il est recommandé d'identifier l'état du quorum après chaque arrêt avec la commande **xscmd -c showQuorumStatus**.

   Si vous utilisez la commande **xscmd -c teardown**, vous pouvez filtrer les noms de serveur. La commande **stopOgServer** nécessite d'entrer un nom de serveur exact ou une liste de noms de serveur à arrêter en parallèle. Vous devez regrouper le processus d'arrêt au lieu d'appeler le processus d'arrêt ou le désassemblage pour de nombreux serveurs en parallèle. En groupant les serveurs à arrêter, la grille de données peut réagir aux serveurs qui sont en cours d'arrêt en déplaçant les fragments sur la grille de données. Vous pouvez utiliser l'une des commandes suivantes pour arrêter vos serveurs :

   Vous pouvez fournir une liste spécifique de serveurs à arrêter à la commande **stopOgServer** ou **xscmd -c teardown** :

   ```
   stopOgServer <server_name>[,<server_name>]
   xsadmin –teardown <server_name>[,<server_name>]
   ```

   7.1.1+

   ```
   xscmd –c teardown -sl <server_name>[,<server_name>]
   ```

   Avec les exemples précédents, la commande **stopOgServer** ou **xscmd -c teardown** exécute les mêmes tâches d'arrêt. Toutefois, vous pouvez filtrer les serveurs à arrêter avec la commande **xscmd -c teardown**. Voir «Arrêt propre des serveurs avec l'utilitaire **xscmd**», à la page 409 pour plus d'informations sur le filtrage des serveurs en fonction de la zone ou du nom d'hôte. La commande teardown filtre les serveurs correspondants et demande si les serveurs sélectionnés sont corrects.

d. Installez les mises à jour sur le serveur de catalogue. Vous pouvez soit migrer le serveur de catalogue vers une nouvelle version majeure du produit ou appliquer un package de maintenance. Voir les rubriques suivantes pour plus d'informations :

   - Pour migrer depuis une installation Version 7.0.x : «Migration vers WebSphere eXtreme Scale Version 7.1.1», à la page 214
   - Pour mettre à niveau une installation version 7.1.0.x : «Utilisation du programme d'installation de mises à jour pour installer des modules de maintenance», à la page 215

e. Redémarrez le serveur de catalogue.

   Si vous utilisez un environnement autonome, voir «Démarrage d'un service de catalogue autonome», à la page 395 pour plus d'informations. Si vous utilisez un environnement autonome WebSphere Application Server, voir

«Démarrage et arrêt des serveurs dans un environnement WebSphere Application Server», à la page 409 pour plus d'informations.

Le serveur de catalogue s'exécute en mode de compatibilité jusqu'à ce que tous les serveurs de catalogue soient amenés au même niveau. Le mode de compatibilité s'applique principalement aux migrations de versions majeures, car les nouvelles fonctions ne sont pas disponibles sur les serveurs qui ne sont pas migrés. Il n'existe aucune restriction sur la durée d'exécution des serveurs de catalogue en mode de compatibilité, mais il est recommandé de migrer tous les serveurs de catalogue vers le même niveau dès que possible.

f. Appliquez les mises à jour aux serveurs de catalogue restants de la configuration.

2. Mettez à niveau les serveurs de conteneur en répétant les étapes suivantes pour chaque serveur de conteneur dans la grille de données. Vous pouvez mettre à niveau les serveurs de conteneur dans n'importe quel ordre. Toutefois, envisagez de mettre à jour les serveurs d'abord, puis les clients, si vous utilisez des fonctions nouvelles dans la mise à niveau.

   a. Arrêtez les serveurs de conteneur à mettre à niveau. Vous pouvez arrêter les groupes de serveurs de conteneur dans les groupes avec la commande **stopOgserver** ou **teardown**. En créant des lots d'opérations teardown et en exécutant des opérations de démarrage de serveur en parallèle, le mécanisme de placement déplace de plus grands groupes de fragments.

   ```
   xsadmin –teardown -fz DefaultZone
   ```

   7.1.1+

   ```
   xscmd -c teardown -z DefaultZone

   Connecting to Catalog service at localhost:1099

   Processing filter options for Server teardown

   The following servers will be torn down:

     container00
     container01
     container02
     container03
     container04


   Do you want to tear down the listed servers? (Y/N)
   ```

   b. Installez les mises à jour sur le serveur de conteneur. Vous pouvez soit migrer les serveurs de conteneur vers une nouvelle version majeure du produit ou appliquer un module de maintenance. Voir les rubriques suivantes pour plus d'informations :

      - Pour migrer depuis une installation Version 7.0.x : «Migration vers WebSphere eXtreme Scale Version 7.1.1», à la page 214
      - Pour mettre à niveau une installation version 7.1.0.x : «Utilisation du programme d'installation de mises à jour pour installer des modules de maintenance», à la page 215

   c. Redémarrez les serveurs de conteneur.

   d. Mettez à niveau les serveurs de conteneur restants de la configuration.

3. Si vous utilisez la réplication multimaître, reconnectez vos domaine de services de catalogue. Utilisez la commande **inattendue -c establishLink** pour reconnecter les domaine de services de catalogue. 7.1.1+

```
xsadmin –ch host –p 1099 –establishLink dname fdHostA:2809,fdHostB:2809
xscmd –c establishLink -cep host:2809 -fd dname -fe fdHostA:2809,fdHostB:2809
```

Que faire ensuite

Vous pouvez également utiliser ces étapes pour revenir à une version antérieure ou pour désinstaller les modules de maintenance. Toutefois, si vous revenez à la version 7.1.0 lorsque vous utilisez la réplication multimaître, la réplication bidirectionnelle peut échouer lorsque vous rétablissez les liaisons. Dans ce cas, redémarrez les domaines de services de catalogue et liez-les de nouveau avec la commande **establishLink**.

Migration vers WebSphere eXtreme Scale Version 7.1.1

Le programme d'installation de WebSphere eXtreme Scale ne vous permet pas de mettre à niveau ou de modifier une précédente installation. Vous devez désinstaller la version précédente avant d'installer la nouvelle version. Vous n'avez pas besoin de faire migrer vos fichiers de configuration car leur compatibilité est ascendante. Mais, si vous avez modifié l'un des scripts qui sont livrés avec le produit, vous devrez réappliquer ces changements aux scripts modifiés.

Avant de commencer

Vérifiez que vos systèmes disposent de la configuration minimale requise pour les versions du produit que vous avez l'intention de migrer et d'installer. Pour plus d'informations, voir «Configurations matérielle et logicielle requises», à la page 49.

Pourquoi et quand exécuter cette tâche

Fusionnez les fichiers script modifiés du produit avec les nouveaux fichiers scripts du produit dans le répertoire `/bin` pour conserver vos modifications.

Conseil : Si vous n'avez pas modifié les fichiers script installés avec le produit, vous n'avez pas besoin d'effectuer les étapes de migration ci-après. En revanche, vous pouvez passer à la version 7.1.1 en désinstallant la version précédente et en installant la nouvelle version dans le même répertoire.

Procédure

1. Arrêtez tous les processus qui utilisent eXtreme Scale.
   - Pour arrêter tous les processus exécutés dans votre environnement eXtreme Scale autonome, consultez la rubrique sur l'arrêt des serveurs autonomes.
   - Pour arrêter tous les processus en cours d'exécution dans votre environnement WebSphere Application Server ou WebSphere Application Server Network Deployment, reportez-vous aux utilitaires de ligne de commande.
2. Sauvegardez les scripts modifiés de votre répertoire d'installation actuel dans un répertoire temporaire.
3. Désinstallez le produit.
4. Installez eXtreme Scale Version 7.1.1. Pour plus d'informations, voir «Installation de WebSphere eXtreme Scale avec l'assistant d'installation», à la page 163.
5. Fusionnez vos modifications apportées aux fichiers du répertoire temporaire avec les nouveaux fichiers scripts du produit, dans le répertoire `/bin`.

6. Démarrez tous vos processus eXtreme Scale pour commencer à utiliser le produit. Pour plus d'informations, voir Chapitre 7, «Administration», à la page 395 les explications sur l'administration de votre environnement.

Utilisation du programme d'installation de mises à jour pour installer des modules de maintenance

Utilisez IBM Update Installer pour mettre à jour votre environnement WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client avec divers types de maintenance, tels que des correctifs temporaires, des groupes de correctifs et des kits d'actualisation.

Pourquoi et quand exécuter cette tâche

Utilisez IBM Update Installer pour installer et appliquer plusieurs types de modules de maintenance pour WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client. Ce programme d'installation de mises à jour étant soumis à des opérations de maintenance régulières, veuillez utiliser sa dernière version.

Procédure

1. Arrêtez tous les processus en cours d'exécution dans votre environnement.
   - Pour arrêter tous les processus en cours d'exécution dans votre environnement eXtreme Scale autonome, voir «Arrêt des serveurs autonomes», à la page 406.
   - Pour arrêter tous les processus en cours d'exécution dans votre environnement WebSphere Application Server, voir le document sur les utilitaires de ligne de commande.
2. Téléchargez la dernière version du programme d'installation de mises à jour. Voir Correctifs recommandés pour plus d'informations.
3. Installez le programme d'installation de mises à jour. Voir l'installation Update Installer for WebSphere Software dans WebSphere Application Server le Centre de documentation pour plus d'informations.
4. Téléchargez dans le répertoire *updi_root*/maintenance les modules de maintenance à installer. Voir Site de support pour plus d'informations.
5. Utilisez le programme d'installation de mises à jour pour installer le correctif temporaire, le groupe de correctifs ou le groupe de mises à jour. Vous pouvez installer le module de maintenance en exécutant l'interface graphique ou le programme d'installation de mises à jour en mode silencieux.

   Exécutez la commande suivante à partir du répertoire *updi_root* pour démarrer l'interface graphique :
   - Linux UNIX `update.sh`
   - Windows `update.bat`

   Exécutez la commande suivante à partir du répertoire *updi_root* pour lancer le programme d'installation de mises à jour en mode silencieux :
   - Linux UNIX `./update.sh -silent -options` *responsefile/file_name*
   - Windows `update.bat -silent -options` *responsefile\file_name*

   En cas d'échec du processus d'installation, consultez le fichier journal temporaire, situé dans le répertoire *updi_root*/logs/update/tmp. Le programme d'installation de mises à jour crée le répertoire *install_root*/logs/update/*maintenance_package*.install qui contient les journaux d'installation.

Migration de l'outil xsadmin vers l'outil xscmd

Dans les versions précédentes, l'outil **xsadmin** était un exemple d'utilitaire de ligne de commande pour surveiller l'état de l'environnement. L'outil **inattendue** a été introduit comme outil officiel de ligne de commande d'administration et de surveillance. Si vous utilisiez l'outil **xsadmin**, migrez les commandes vers le nouvel outil **xscmd**.

Equivalents des commandes xsadmin et xscmd

Tableau 9. Arguments de l'utilitaire **xsadmin** *et commandes équivalentes* **xscmd**. Certaines commandes **xscmd** ont une forme courte et une forme longue. La forme courte des commandes a un tiret (-) et la forme longue, deux (--). Vous pouvez utiliser l'une à la place de l'autre et inversement.

Arguments de ligne de commande xsadmin	Commande équivalente xscmd	Paramètres de commande xscmd
-bp	• **-cep** *hostname:listener_port* • **--catalogEndpoint** *hostname:listener_port*	s/o
-ch	• **-cep** *hostname:listener_port* • **--catalogEndpoint** *hostname:listener_port*	s/o
-clear	**-c clearGrid**	**-g, -ms, -v, -m, (-cep)**
-containers	• **-c listCoreGroups** • **-c listCoreGroupMembers -cg** *core_group*	**-e, -I, , -st, -snp, -ct, -s, -p, -hf, -z, -g, -m, -ms**
-continuous	• **-cnt** • **--continuous**	s/o
-coregroups	• **-c listCoreGroups** • **-c listCoreGroupMembers -cg** *core_group*	s/o
-dismissLink *<catalog_service_domain>*	**-c dismissLink**	• **-fd** *<foreignCatalogServiceDomain>* • **--foreignCatalogServiceDomain** *<foreignCatalogServiceDomain>*
-dmgr	s/o - Cet argument est déterminé automatiquement avec **xscmd**	s/o
-empties	arg spécifique d'une nouvelle commande	s/o
-establishLink *<foreign_domain_name> <host1:port1,host2:port2...>*	**-c establishLink**	• **-fd** *<foreignCatalogServiceDomain>* **-fe** *<host1:port1,host2:port2...>* • **--foreignCatalogServiceDomain** *<foreignCatalogServiceDomain>* **-foreignEndPoints** *<host1:port1,host2:port2...>*
-fc	• **-ct** • **--container**	s/o
-fh	• **-hf** • **--hostFilter**	s/o

Tableau 9. Arguments de l'utilitaire **xsadmin** *et commandes équivalentes* **xscmd** *(suite).* Certaines commandes **xscmd** ont une forme courte et une forme longue. La forme courte des commandes a un tiret (-) et la forme longue, deux (--). Vous pouvez utiliser l'une à la place de l'autre et inversement.

Arguments de ligne de commande xsadmin	Commande équivalente xscmd	Paramètres de commande xscmd
-fm	• **-m** • **--map**	s/o
-fnp	• **-snp** • **--serversWithNoPrimaries**	s/o
-fp	• **-p** • **--partitionId**	s/o
-fs	• **-s** • **--server**	s/o
-fst	• **-st** <*shard_type*> • **--shardType** <*shard_type*> Shard values: P=primary A=asyncReplica S=syncReplica	s/o
-fz	• **-z** • **--zone**	s/o
-force	arg spécifique d'une nouvelle commande	
-g	• **-g** • **--objectGrid**	s/o
-getstatsspec	**-c getStatsSpec**	s/o
-getTraceSpec	**-c getTraceSpec**	s/o
-h	Vous pouvez exécuter l'aide avec ou sans un nom de commande spécifique : • **-h** • **--help** • **-h** <*command_name*> • **--help** <*command_name*>	s/o
-hosts	**-c listHosts**	**-g, -ms, -st, -c, -s, -hf, -z**
-jmxUrl	• **-cep** *hostname:listener_port* • **--catalogEndpoint** *hostname:listener_port*	s/o
-l	**-c listObjectGridNames**	s/o
-m	• **-ms** • **--mapSet**	s/o
-mapsizes	**-c showMapSizes**	**-g, -ms, -cnt, -i, [-ct, -z, -s, -hf, sht [P,A,S], -p]**
-mbeanservers	**-c listAllJMXAddresses**	s/o
-overridequorum	**-c overrideQuorum**	s/o

Tableau 9. Arguments de l'utilitaire **xsadmin** *et commandes équivalentes* **xscmd** *(suite).* Certaines commandes **xscmd** ont une forme courte et une forme longue. La forme courte des commandes a un tiret (-) et la forme longue, deux (--). Vous pouvez utiliser l'une à la place de l'autre et inversement.

Arguments de ligne de commande **xsadmin**	Commande équivalente **xscmd**	Paramètres de commande **xscmd**
-password	• **-pwd** • **--password**	s/o
-p	• **-cep** *hostname:listener_port* • **--catalogEndpoint** *hostname:listener_port*	s/o
-placementStatus	**-c placementServiceStatus**	**-g, -ms**
-primaries	**-c showPlacement -sf P**	**-e, -I, , -st, -snp, -ct, -s, -p, -hf, -z, -g, -m, -ms**
-profile	Pour enregistrer les paramètres de sécurité actuels dans un profil de sécurité : • **-ssp** *profile_name* • **--saveSecProfile** *profile_name* Pour utiliser un profil de sécurité spécifié : • **-sp** *profile_name* • **--securityProfile** *profile_name*	
-quorumstatus	**-c showQuorumStatus**	s/o
-releaseShard <*container_server_name*> <*objectgrid_name*> <*map_set_name*> <*partition_name*>	**-c releaseShard**	**-c, -g, -ms, -p**
-reserved	• **-sf** *[R,U]* • **--shardFilter** *[R,U]* R=reserved, U=unassigned	s/o
-reserveShard <*container_server_name*> <*objectgrid_name*> <*map_set_name*> <*partition_name*>	**-c reserveShard**	**-c, -g, -ms, -p**
-resumeBalancing <*objectgrid_name*> <*map_set_name*>	**-c resumeBalancing**	**-g, -ms**
-revisions	**-c revisions**	**-s,-p,-g,-m**
-routetable	**-c routetable**	**-z, -hf,-p,-g,-ms**
-settracespec <*trace_string*>	**-c setTraceSpec**	**-spec** <*trace_string*>
-swapShardWithPrimary <*container_server_name*> <*objectgrid_name*> <*map_set_name*> <*partition_name*>	**-c swapShardWithPrimary**	**-c -g, -ms, -p**
-setstatsspec <*stats_spec*>	**-c setStatsSpec**	**-spec** <*stats_spec*>
-suspendBalancing <*objectgrid_name*> <*map_set_name*>	**-c suspendBalancing**	**-g, -ms**
-ssl	• **-ssl** • **--enableSSL**	s/o

Tableau 9. Arguments de l'utilitaire **xsadmin** *et commandes équivalentes* **xscmd** *(suite).* Certaines commandes **xscmd** ont une forme courte et une forme longue. La forme courte des commandes a un tiret (-) et la forme longue, deux (--). Vous pouvez utiliser l'une à la place de l'autre et inversement.

Arguments de ligne de commande xsadmin	Commande équivalente xscmd	Paramètres de commande xscmd
`-teardown`	`-c teardown`	`-f, , -st, -snp, -c, -s, -p, -hf, -z, -g, -ms, -m`
`-triggerPlacement`	`-c triggerPlacement`	`-g, -ms`
`-trustPass`	• `-tsp` • `--trustStorePassword`	s/o
`-trustPath`	• `-ts` • `--trustStore`	s/o
`-trustType`	• `-tst` • `--trustStoreType`	s/o
`-unassigned`	`-c showPlacement -sf U`	`-e, -I, , -st, -snp, -ct, -s, -p, -hf, -z, -g, -m, -ms`
`-username`	• `-user` • `--username`	s/o
`-v`	• `-v` • `--verbose`	s/o
`-xml`	`-c showPlacement`	s/o

Propriétés et API obsolètes

Les propriétés et API suivantes sont obsolètes dans la version 7.1.1. Utilisez l'action de migration recommandée pour déterminer comment mettre à jour votre configuration.

7.1.1+ Eléments obsolètes dans la version 7.1.1

Tableau 10. Propriétés et API obsolètes

Dépréciation	Action de migration recommandée
Classe com.ibm.websphere.objectgrid.plugins.builtins.TranPropListener Cette classe était utilisée pour propager les processus de validation de la transaction ObjectGrid ayant abouti aux autres serveurs d'applications WebSphere hébergeant la même instance ObjectGrid en fonction du nom ObjectGrid.	7.1.1+ L'interface TranPropListener a été remplacée par l'interface JMSObjectGridEventListener qui est une implémentation JMS de l'interface ObjectGridEventListener. Elle prend en charge l'invalidation de cache local côté client et la réplication entre homologues.
Classe com.ibm.websphere.objectgrid.plugins.OptimisticCallback Cette classe était utilisée pour fournir des opérations de comparaison optimistes pour les valeurs d'une mappe.	7.1.1+ Le plug-in OptimisticCallback a été remplacé par l'interface ValueDataSerializer.Versionable que vous pouvez implémenter lorsque vous utilisez le plug-in DataSerializer avec le mode de copie COPY_TO_BYTES ou lorsque vous utilisez l'annotation @Version avec l'API EntityManager. Voir la documentation d'API pour plus d'informations.
Plug-in com.ibm.websphere.objectgrid.plugins.NoVersioningOptimisticCallback Ce plug-in était utilisé pour le verrouillage optimiste sans vérification de version. Avec ce gestionnaire OptimisticCallback intégré, le chargeur effectuait la vérification de version, mais le verrouillage optimiste était utilisé pour que les données validées soient toujours retournées sur une lecture.	7.1.1+ L'interface NoVersioningOptimisticCallback étend l'interface OptimisticCallback. Par conséquent, utilisez la stratégie de verrouillage optimiste avec l'isolement de transaction par défaut READ_COMMITTED ou inférieur. Voir Optimisation des performances de verrouillage pour plus d'informations.

Tableau 10. Propriétés et API obsolètes (suite)

Dépréciation	Action de migration recommandée
Classe com.ibm.websphere.objectgrid.plugins.ObjectTransformer Ce plug-in était utilisé pour sérialiser, désérialiser et copier des objets dans le cache.	7.1.1+ L'interface ObjectTransformer a été remplacée par les plug-ins DataSerializer que vous pouvez utiliser pour stocker efficacement les données arbitraires dans WebSphere eXtreme Scale pour que les API de produit existantes puissent interagir efficacement avec vos données.
Méthode com.ibm.websphere.objectgrid.BackingMap.setMapEventListeners Cette méthode était utilisée pour définir la liste des objets MapEventListener.	7.1.1+ Utilisez la méthode addMapEventListener(EventListener) ou removeMapEventListener(EventListener) ajouter ou supprimer des programmes d'écoute d'événement à partir dans une mappe de sauvegarde.
Méthode com.ibm.websphere.objectgrid.ObjectGrid.setEventListeners Cette méthode était utilisée pour remplacer la liste en cours des objets ObjectGridEventListener par la liste fournie des objets ObjectGridEventListeners.	7.1.1+ Utilisez la méthode addEventListener(EventListener) ou removeEventListener(EventListener) pour ajouter ou supprimer des programmes d'écoute d'événement et de cycle de vie dans la grille de données.

7.1.1+ Fonctions stabilisées dans la version 7.1.1

Si une fonction est listée comme étant stabilisée, IBM n'envisage pas de la rendre obsolète ou de la supprimer dans une version suivante du produit, mais les efforts porteront sur la fonction alternative. Les utilisateurs n'ont pas besoin de modifier les applications et les scripts existants qui utilisent une fonction stabilisée, mais doivent envisager d'utiliser l'alternative stratégique pour les nouvelles applications.

Tableau 11. Propriétés et API obsolètes

Fonction stabilisée	Action de migration recommandée
xsadmin L'utilitaire xsadmin est fourni comme exemple pour monter comment créer des utilitaires personnalisés pour votre déploiement.	7.1.1+ Utilisez l'utilitaire **xscmd** effectuer des tâches d'administration dans l'environnement, telles qu'établir des liens de réplication multimaître, remplacer un quorum et arrêter des groupes de serveurs avec la commande teardown.

Eléments obsolètes dans la version 7.1

Tableau 12. Propriétés et API obsolètes

Dépréciation	Action de migration recommandée
Propriété catalog.services.cluster de cellule et de serveur : cette propriété personnalisée servait à définir un groupe de serveurs de catalogue dans la configuration de WebSphere Application Server.	Cette propriété personnalisée est obsolète depuis la version 7.1. Dans la console d'administration de WebSphere Application Server, créez un domaine de services de catalogue qui crée la même configuration qu'avec la propriété personnalisée. Pour plus d'informations, voir .
Bean géré et interface CoreGroupServicesMBean	Ce bean géré est obsolète depuis la version 7.1. Utilisez plutôt le bean géré CatalogServiceManagementMBean.
Utilisation du bean géré ServerMBean.updateTraceSpec()	Cette opération est obsolète depuis la version 7.1. Utilisez plutôt l'attribut TraceSpec du bean géré DynamicServerMBean.
Bean géré CoreGroupServicesMBean	Ce bean géré est obsolète depuis la version 7.1. Utilisez le bean géré CatalogServiceManagementMbean à la place.
Exception ServiceUnavailableException	Cette exception est obsolète depuis la version 7.1. Utilisez plutôt l'exception TargetNotAvailableException.

Tableau 12. Propriétés et API obsolètes (suite)

Dépréciation	Action de migration recommandée
Partitioning Facility (WPF) : l'utilitaire de partitionnement est un ensemble d'API de programmation qui permettent aux applications Java EE de prendre en charge la mise en clusters asymétrique.	Les fonctionnalités de WPF peuvent également être utilisées dans WebSphere eXtreme Scale.
StreamQuery : Requête continue sur les données en cours stockées dans les mappes ObjectGrid.	Néant
Configuration de grille statique : Topologie statique basée sur les clusters, qui utilise le fichier XML de déploiement des clusters.	Remplacée par la topologie de déploiement dynamique, améliorée, pour la gestion des grilles de données de grande taille.
Propriétés système dépréciées : Les propriétés système permettant de spécifier les fichiers de propriétés des serveurs et des clients sont dépréciées.	Vous pouvez toujours utiliser ces arguments, mais vous devrez remplacer vos propriétés par les nouvelles valeurs. **-Dcom.ibm.websphere.objectgrid.CatalogServerProperties** Cette propriété est dépréciée depuis la version 7.0 de WebSphere eXtreme Scale. Utilisez la propriété **-Dobjectgrid.server.props**. **-Dcom.ibm.websphere.objectgrid.ClientProperties** Cette propriété est dépréciée depuis la version 7.0 de WebSphere eXtreme Scale. Utilisez la propriété **-Dobjectgrid.client.props**. **-Dobjectgrid.security.server.prop** Cette propriété est dépréciée depuis la version 6.1.0.3 de WebSphere eXtreme Scale. Utilisez la propriété **-Dobjectgrid.server.prop**. **-serverSecurityFile** Cet argument n'est plus utilisé dans WebSphere eXtreme Scale Version 6.1.0.3. Cette option est transmise dans le script `startOgServer`. Utilisez l'argument **-serverProps**.

Chapitre 6. configuration

Vous pouvez configurer WebSphere eXtreme Scale pour qu'il soit exécuté dans un environnement autonome ou configurer eXtreme Scale pour qu'il soit exécuté dans un environnement avec WebSphere Application Server ou WebSphere Application Server Network Deployment. Pour qu'un déploiement WebSphere eXtreme Scale sélectionne les modifications de configuration sur le serveur de la grille de données, vous devez redémarrer les processus pour que ces modifications entrent en vigueur au lieu d'être appliquées de manière dynamique. Toutefois, côté client, vous ne pouvez pas modifier les paramètres de configuration d'une instance de client existante, mais vous pouvez créer un client avec les paramètres dont vous avez besoin à l'aide d'un fichier XML ou d'un programme. Lorsque vous créez un client, vous pouvez remplacer les paramètres par défaut provenant de la configuration de serveur actuelle.

Méthodes de configuration

Vous pouvez configurer la plupart des éléments du produit avec des fichiers XML et des fichiers de propriétés. Vous pouvez également utiliser des méthodes de programmation, y compris des interfaces de programmation d'applications et système, des plug-ins et des beans gérés.

Pourquoi et quand exécuter cette tâche

Utilisez les fichiers suivants pour créer une configuration de base :

Fichier de propriétés du serveur
Utilisez le fichier de propriétés de serveur pour définir les paramètres des serveurs de catalogue et de conteneur (trace, consignation, sécurité, ports, etc.). Vous pouvez envoyer un fichier de propriétés au script **startOgServer**, placer le fichier dans le chemin d'accès aux classes ou définir le fichier avec des propriétés système.

Fichier de propriétés du client
Utilisez le fichier de propriétés de client pour définir les propriétés dans les clients, y compris les ports et les paramètres de sécurité. Vous pouvez spécifier le fichier des propriétés de client à utiliser avec une propriété système en le plaçant dans le chemin d'accès aux classes ou en utilisant la méthode ClientClusterContext.getClientProperties.

Fichier XML descripteur ObjectGrid
Le fichier XML descripteur ObjectGrid décrit la grille de données et la configuration de mappe. Spécifiez le fichier à utiliser avec le script **startOgServer** pour les configurations autonomes ou ajoutez le fichier au module d'application pour les configurations WebSphere Application Server.

Fichier XML descripteur de la stratégie de déploiement
Le fichier XML de stratégie de déploiement contrôle la segmentation et le placement des données sur divers serveurs de conteneur dans la configuration. Spécifiez le fichier à utiliser avec le script **startOgServer** pour les configurations autonomes ou ajoutez le fichier au module d'application pour les configurations WebSphere Application Server.

Configuration des grilles de données

Utilisez un fichier descripteur XML ObjectGrid pour configurer des mappes de données, des mappes de sauvegarde, des plug-ins, etc. Pour configurer WebSphere eXtreme Scale, utilisez un fichier XML de descripteur d'ObjectGrid et l'API ObjectGrid. Dans le cas d'une topologie répartie, vous avez besoin d'un fichier descripteur XML d'ObjectGrid et d'un fichier XML de stratégie de déploiement.

Configuration de déploiements locaux

Une configuration eXtreme Scale en mémoire locale peut être créée à l'aide d'un fichier XML descripteur d'ObjectGrid ou des API.

Pourquoi et quand exécuter cette tâche

Pour créer un déploiement local, vous créez un fichier XML descripteur d'ObjectGrid et transmettez le fichier createObjectGrid aux méthodes de l'interface ObjectGridManager.

Vous pouvez également créer le déploiement complet à l'aide d'un programme avec l'interface ObjectGridManager.

Procédure

1. Créez un fichier XML descripteur d'ObjectGrid.

   Le fichier `companyGrid.xml` ci-après est un exemple de XML de descripteur d'ObjectGrid. Les premières lignes de ce fichier incluent l'en-tête requis de chaque fichier XML ObjectGrid. Le fichier définit une instance ObjectGrid nommée "CompanyGrid" et plusieurs mappes de sauvegarde intitulées "Customer," "Item," "OrderLine" et "Order."

   fichier
   companyGrid.xml
   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">

    <objectGrids>
     <objectGrid name="CompanyGrid">
      <backingMap name="Customer" />
      <backingMap name="Item" />
      <backingMap name="OrderLine" />
      <backingMap name="Order" />
     </objectGrid>
    </objectGrids>

   </objectGridConfig>
   ```

2. Transmettez le fichier XML à l'une des méthodes createObjectGrid de l'interface ObjectGridManager.

   L'exemple de code ci-après valide le fichier `companyGrid.xml` par rapport au schéma XML et crée l'instance ObjectGrid intitulée "CompanyGrid." L'instance ObjectGrid nouvellement créée n'est pas placée en cache.

   ```
   ObjectGridManager objectGridManager = ObjectGridManagerFactory.getObjectGridManager();
   ObjectGrid companyGrid = objectGridManager.createObjectGrid("CompanyGrid",
    new URL("file:etc/test/companyGrid.xml"), true, false);
   ```

Que faire ensuite

Voir Création d'instances avec l'interface ObjectGrid ObjectGridManager pour plus d'informations sur la définition toutes les mappes à l'aide d'un programme avec les méthodes createObjectGrid sur l'interface ObjectGridManager.

Activation des expulseurs avec la configuration XML

Au lieu d'utiliser l'interface BackingMap pour définir à l'aide d'un programme les attributs BackingMap qui doivent être utilisés par l'expulseur basé sur la durée de vie, vous pouvez utiliser un fichier XML pour configurer chaque instance de BackingMap. Le code suivant démontre comment définir ces attributs pour trois mappes BackingMap différentes :

Avant de commencer

Avant de commencer, choisissez le type d'expulseur que vous allez utiliser :

- **Expulseur TTL basé sur le temps par défaut :** l'expulseur par défaut utilise une règle d'expulsion TTL (time-to-live) pour chaque instance BackingMap.
- **Mécanisme d'expulsion enfichable :** les expulseurs enfichables utilisent généralement une règle d'expulsion basée sur le nombre de tentatives et non pas sur le temps.

La plupart des paramètres d'expulseur doivent être définis avant l'initialisation d'ObjectGrid.

Procédure

- Pour définir l'expulseur TTL par défaut, ajoutez l'attribut ttlEvictorType dans le fichier XML descripteur d'ObjectGrid.

  L'exemple suivant montre que l'instance BackingMap map1 utilise le type d'expulseur TTL NONE. L'instance BackingMap map2 utilise le ttype d'expulseur TTL LAST_ACCESS_TIME ou LAST_UPDATE_TIME. Indiquez que l'un ou l'autre de ces deux paramètres. L'instance BackingMap map2 a une valeur de durée de vie de 1800 seconds, soit 30 minutes. L'instance BackingMap map3 est définie pour utiliser le type d'expulseur basé sur la durée de vie CREATION_TIME et a une durée de vie de 1200 secondes (20 minutes).

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
<objectGrids>
    <objectGrid name="grid1">
        <backingMap name="map1" ttlEvictorType="NONE" />
        <backingMap name="map2" ttlEvictorType="LAST_ACCESS_TIME|LAST_UPDATE_TIME"
     timeToLive="1800" />
        <backingMap name="map3" ttlEvictorType="CREATION_TIME"
     timeToLive="1200" />
    </objectgrid>
</objectGrids>
```

Figure 27. Enable TimeToLive evictor with XML

- Pour définir un expulseur enfichable, utilisez l'exemple suivant.

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
<objectGrids>
    <objectGrid name="grid">
        <backingMap name="map1" ttlEvictorType="NONE" pluginCollectionRef="LRU" />
        <backingMap name="map2" ttlEvictorType="NONE" pluginCollectionRef="LFU" />
    </objectGrid>
</objectGrids>
<backingMapPluginCollections>
    <backingMapPlugincollection id="LRU">
        <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
            <property name="maxSize" type="int" value="1000" description="set max size
       for each LRU queue" />
            <property name="sleepTime" type="int" value="15" description="evictor
       thread sleep time" />
            <property name="numberOfLRUQueues" type="int" value="53" description="set number
       of LRU queues" />
        </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="LFU">
        <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LFUEvictor">
            <property name="maxSize" type="int" value="2000" description="set max size for each LFU heap" />
            <property name="sleepTime" type="int" value="15" description="evictor thread sleep time" />
            <property name="numberOfHeaps" type="int" value="211" description="set number of LFU heaps" />
        </bean>
    </backingMapPluginCollection>
</backingMapPluginCollections>
</objectGridConfig>
```

Figure 28. Connexion d'un expulseur en utilisant XML

Configuration d'une stratégie de verrouillage

Vous pouvez définir une stratégie optimiste, pessimiste ou sans verrouillage sur chaque BackingMap de la configuration de WebSphere eXtreme Scale.

Pourquoi et quand exécuter cette tâche

Chaque instance de mappe de sauvegarde BackingMap peut être configurée pour utiliser l'une de ces stratégies de verrouillage :

1. mode de verrouillage optimiste
2. mode de verrouillage pessimiste
3. aucun

La stratégie de verrouillage OPTIMISTIC est le mode par défaut. Utilisez le verrouillage optimiste lorsque les données sont modifiées rarement. Les verrous sont uniquement maintenus pendant un laps de temps limité tandis que les données sont lues depuis le cache et copiées dans la transaction. Lorsque le cache de transaction est synchronisé avec le cache principal, tous les objets mis en cache qui ont été mis à jour sont vérifiés avec la version d'origine. Si la vérification échoue, la transaction est annulée et une exception OptimisticCollisionException est provoquée.

La stratégie de verrouillage PESSIMISTIC obtient des verrous pour les entrées de cache et doit être utilisée lorsque les données sont modifiées fréquemment. A chaque lecture d'une entrée de cache, un verrou est obtenu et maintenu de façon conditionnelle jusqu'à la fin de la transaction. La durée de certains verrous peut être paramétrée à l'aide des niveaux d'isolement de transaction pour la session.

Si le verrouillage n'est pas obligatoire car les données ne sont jamais mises à jour ou le sont au cours de période calmes, vous pouvez le désactiver à l'aide de la

stratégie de verrouillage NONE. Cette stratégie est très rapide car un gestionnaire de verrou n'est pas requis. La stratégie de verrouillage NONE est idéale pour les tables de recherche et les mappes en lecture seule.

Pour plus d'informations sur les stratégies de verrouillage, voir les Stratégies de verrouillageinformations relatives aux stratégies de verrouillage dans *Présentation du produit*.

Procédure

- **Configurez une stratégie de verrouillage optimiste**
  - Utilisation de la méthode setLockStrategy à l'aide d'un programme :

```
import com.ibm.websphere.objectgrid.BackingMap;
import com.ibm.websphere.objectgrid.LockStrategy;
import com.ibm.websphere.objectgrid.ObjectGrid;
import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
...
ObjectGrid og =
 ObjectGridManagerFactory.getObjectGridManager().createObjectGrid("test");
BackingMap bm = og.defineMap("optimisticMap");
bm.setLockStrategy( LockStrategy.OPTIMISTIC );
```

  - Utilisation de l'attribut lockStrategy dans Fichier XML du descripteur d'ObjectGrid :

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">
    <objectGrids>
        <objectGrid name="test">
            <backingMap name="optimisticMap"
                lockStrategy="OPTIMISTIC"/>
        </objectGrid>
    </objectGrids>
</objectGridConfig>
```

- **Configurez une stratégie de verrouillage pessimiste**
  - Utilisation de la méthode setLockStrategy à l'aide d'un programme :

```
specify pessimistic strategy programmatically
import com.ibm.websphere.objectgrid.BackingMap;
import com.ibm.websphere.objectgrid.LockStrategy;
import com.ibm.websphere.objectgrid.ObjectGrid;
import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
...
ObjectGrid og =
 ObjectGridManagerFactory.getObjectGridManager().createObjectGrid("test");
BackingMap bm = og.defineMap("pessimisticMap");
bm.setLockStrategy( LockStrategy.PESSIMISTIC);
```

  - Utilisation de l'attribut lockStrategy dans Fichier XML du descripteur d'ObjectGrid.

```
specify pessimistic strategy using XML
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">

    <objectGrids>
        <objectGrid name="test">
            <backingMap name="pessimisticMap"
                lockStrategy="PESSIMISTIC"/>
        </objectGrid>
    </objectGrids>
</objectGridConfig>
```

- **Configurez une stratégie sans verrouillage**
  - Utilisation de la méthode setLockStrategy à l'aide d'un programme :

```
import com.ibm.websphere.objectgrid.BackingMap;
import com.ibm.websphere.objectgrid.LockStrategy;
import com.ibm.websphere.objectgrid.ObjectGrid;
import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
...
ObjectGrid og =
 ObjectGridManagerFactory.getObjectGridManager().createObjectGrid("test");
BackingMap bm = og.defineMap("noLockingMap");
bm.setLockStrategy( LockStrategy.NONE);
```

  - Utilisation de l'attribut lockStrategy dans Fichier XML du descripteur d'ObjectGrid :

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">

    <objectGrids>
        <objectGrid name="test">
            <backingMap name="noLockingMap"
                lockStrategy="NONE"/>
        </objectGrid>
    </objectGrids>
</objectGridConfig>
```

Que faire ensuite

Pour éviter une exception java.lang.IllegalStateException, vous devez appeler la méthode setLockStrategy avant d'appeler les méthodes initialize ou getSession sur l'instance ObjectGrid.

Configuration de la réplication entre homologues avec JMS

Le mécanisme de réplication entre homologues basé sur JMS (Java Message Service) est utilisé dans les environnements WebSphere eXtreme Scale réparti et local. JMS est un processus de réplication de coeur à coeur qui permet aux mises à jour de données de circuler parmi les ObjectGrid locaux et les ObjectGrid répartis. Par exemple, avec ce mécanisme, vous pouvez transférer les mises à jour de données d'une grille de données eXtreme Scale vers une grille eXtreme Scale locale ou d'une grille vers une autre dans domaine système différent.

Avant de commencer

Le mécanisme JMS de réplication entre homologues repose sur l'ObjectGridEventListener JMS pré-intégré, com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener. Pour des informations plus détaillées sur l'activation du mécanisme de réplication entre homologues, voir «Programme d'écoute d'événement JMS», à la page 232.

Pour plus d'informations, voir «Activation du mécanisme d'invalidation de client», à la page 296.

Vous trouverez ci-après un exemple de configuration XML permettant d'activer un mécanisme de réplication entre homologues sur une configuration eXtreme Scale :

Configuration de réplication entre homologues - Exemple de XML

```
<bean id="ObjectGridEventListener"
className="com.ibm.websphere.objectgrid.plugins.JMSObjectGridEventListener">
 <property name="replicationRole" type="java.lang.String" value="DUAL_ROLES" description="" />
  <property name="replicationStrategy" type="java.lang.String" value="PUSH" description="" />
```

```
 <property name="jms_topicConnectionFactoryJndiName" type="java.lang.String"
 value="defaultTCF" description="" />
 <property name="jms_topicJndiName" type="java.lang.String" value="defaultTopic" description="" />
 <property name="jms_topicName" type="java.lang.String" value="defaultTopic" description="" />
 <property name="jms_userid" type="java.lang.String" value="" description="" />
 <property name="jms_password" type="java.lang.String" value="" description="" />
 <property name="jndi_properties" type="java.lang.String"
value="java.naming.factory.initial=org.apache.activemq.jndi.ActiveMQInitialContextFactory;
java.naming.provider.url=tcp://localhost:61616;connectionFactoryNames=defaultTCF;
topic.defaultTopic=defaultTopic"
 description="jndi properties" />
    </bean>
```

Répartition de modifications entre des machines virtuelles Java homologues

Les objets LogSequence et LogElement répartissent les modifications entre des machines virtuelles Java homologues et, à l'aide d'un plug-in ObjectGridEventListener, ils communiquent les modifications intervenues dans une transaction eXtreme Scale.

Pour plus d'informations sur l'utilisation de JMS (Java Message Service) pour répartir les modifications transactionnelles, voir Répartition des transactions.

Il est impératif au préalable que l'instance ObjectGrid soit mise en cache par ObjectGridManager. Voir les méthodes createObjectGrid pour plus d'informations à ce sujet. La valeur booléenne de cacheInstance doit être définie comme true.

Il n'est pas nécessaire que vous implémentiez vous-même ce mécanisme. Il existe en effet un mécanisme pré-intégré de réplication entre homologues qui se chargera de cette fonction. Voir «Configuration de la réplication entre homologues avec JMS», à la page 228.

Les objets fournissent aux applications le moyen de publier facilement les modifications qui sont intervenues dans un ObjectGrid, à savoir un transport de messages vers des ObjectGrids homologues situés sur des machines virtuelles Java pour appliquer ces modifications sur ces JVM. La classe LogSequenceTransformer est indispensable pour cette prise en charge. Nous allons expliquer ici comment écrire, pour la propagation des messages, un programme d'écoute utilisant un système de messagerie Java Message Service (JMS). En fait un plug-in IBM permet à eXtreme Scale de prendre en charge la transmission de LogSequences qui résultent de la validation d'une transaction eXtreme Scale entre des membres d'un cluster WebSphere Application Server. Cette fonction n'est pas activée par défaut, mais il est possible de la configurer pour la rendre opérationnelle. Mais, lorsque l'utilisateur ou le producteur ne sont pas WebSphere Application Server, le recours à un système externe de messagerie JMS peut s'avérer nécessaire.

Implémenter le mécanisme

La classe LogSequenceTransformer et les API ObjectGridEventListener, LogSequence et LogElement permettent l'utilisation de n'importe quel mécanisme de publication/abonnement pour la répartition des modifications et le filtrage des mappes à répartir. Les fragments de code utilisés ici montrent comment exploiter ces API avec JMS pour construire un ObjectGrid d'égal à égal, partagé par des applications hébergées sur plusieurs sortes de plateformes partageant un transport commun de messages.

Initialisation du plug-in

L'ObjectGrid appelle la méthode initialize du plug-in, qui fait partie du contrat de l'interface ObjectGridEventListener, lorsque l'ObjectGrid démarre. La méthode

initialize doit obtenir ses ressources JMS (connexions, sessions et diffuseurs de publications) et elle doit démarrer l'unité d'exécution qu'est le programme d'écoute JMS.

Les exemples suivants illustrent la méthode initialize :

exemple de méthode initialize

```
public void initialize(Session session) {
        mySession = session;
        myGrid = session.getObjectGrid();
        try {
            if (mode == null) {
                throw new ObjectGridRuntimeException("No mode specified");
            }
            if (userid != null) {
                connection = topicConnectionFactory.createTopicConnection(userid,
        password);
            } else
                connection = topicConnectionFactory.createTopicConnection();

            // need to start the connection to receive messages.
            connection.start();

            // the jms session is not transactional (false).
            jmsSession = connection.createTopicSession(false,
       javax.jms.Session.AUTO_ACKNOWLEDGE);
            if (topic == null)
                if (topicName == null) {
                    throw new ObjectGridRuntimeException("Topic not specified");
                } else {
                    topic = jmsSession.createTopic(topicName);
                }
            publisher = jmsSession.createPublisher(topic);
            // start the listener thread.
            listenerRunning = true;
            listenerThread = new Thread(this);
            listenerThread.start();
        } catch (Throwable e) {
            throw new ObjectGridRuntimeException("Cannot initialize", e);
        }
    }
```

L'unité d'exécution lancée par le code est une unité Java 2 Platform, Standard Edition (Java SE). Pour une exécution sur WebSphere Application Server version 6.x ou sur WebSphere Application Server version 5.x Enterprise, vous devrez utiliser l'API de bean asynchrone pour lancer cette unité d'exécution de démon. Vous pouvez également utiliser les API communes. Voici un exemple de fragment de code montrant la même action effectuée à l'aide d'un gestionnaire de travaux :

```
// start the listener thread.
listenerRunning = true;
workManager.startWork(this, true);
```

Par ailleurs, le plug-in doit implémenter l'interface Work et non l'interface Runnable. Vous devez également ajouter une méthode release pour définir comme false la variable listenerRunning. Le plug-in doit être fourni avec une instance WorkManager dans son constructeur ou par injection si l'on utilise un conteneur IoC (Inversion of Control).

Transmission des modifications

Voici un exemple de méthode transactionEnd pour la publication des modifications locales apportées à un ObjectGrid. Cet exemple utilise JMS, bien qu'il soit possible d'utiliser n'importe quel transport de messages capable de publication/abonnement fiable.

Exemple de méthode transactionEnd

```
// This method is synchronized to make sure the
    // messages are published in the order the transaction
    // were committed. If we started publishing the messages
    // in parallel then the receivers could corrupt the Map
    // as deletes may arrive before inserts etc.
    public synchronized void transactionEnd(String txid, boolean isWriteThroughEnabled,
    boolean committed,
            Collection changes) {
        try {
            // must be write through and commited.
            if (isWriteThroughEnabled && committed) {
                // write the sequences to a byte []
                ByteArrayOutputStream bos = new ByteArrayOutputStream();
                ObjectOutputStream oos = new ObjectOutputStream(bos);
                if (publishMaps.isEmpty()) {
                    // serialize the whole collection
                    LogSequenceTransformer.serialize(changes, oos, this, mode);
                } else {
                    // filter LogSequences based on publishMaps contents
                    Collection publishChanges = new ArrayList();
                    Iterator iter = changes.iterator();
                    while (iter.hasNext()) {
                        LogSequence ls = (LogSequence) iter.next();
                        if (publishMaps.contains(ls.getMapName())) {
                            publishChanges.add(ls);
                        }
                    }
                    LogSequenceTransformer.serialize(publishChanges, oos, this, mode);
                }
                // make an object message for the changes
                oos.flush();
                ObjectMessage om = jmsSession.createObjectMessage(bos.toByteArray());
                // set properties
                om.setStringProperty(PROP_TX, txid);
                om.setStringProperty(PROP_GRIDNAME, myGrid.getName());
                // transmit it.
                publisher.publish(om);
            }
        } catch (Throwable e) {
            throw new ObjectGridRuntimeException("Cannot push changes", e);
        }
    }
```

Cette méthode utilise plusieurs variables d'instance :

- La variable `jmsSession` : session JMS servant à publier les messages. Elle est créée lors de l'initialisation du plug-in.
- La variable `mode` : le mode de répartition.
- La variable `publishMaps` : ensemble contenant le nom de chacune des mappes dont les modifications sont à publier. La variable vide signifie que la totalité des mappes sont à publier.
- La variable `publisher` : objet TopicPublisher qui est créé durant l'initialisation du plug-in.

Réception et application des messages d'actualisation

Vient à présent la méthode run. Cette méthode s'exécute en boucle jusqu'à ce que l'application arrête la boucle. Chaque itération de la boucle tente de réceptionner un message JMS et de l'appliquer à l'ObjectGrid.

Exemple de méthode run de message JMS

```
private synchronized boolean isListenerRunning() {
        return listenerRunning;
    }
```

```
public void run () {
    try {
        System.out.println("Listener starting");
        // get a jms session for receiving the messages.
        // Non transactional.
        TopicSession myTopicSession;
        myTopicSession = connection.createTopicSession(false, javax.jms.
   Session.AUTO_ACKNOWLEDGE);

        // get a subscriber for the topic, true indicates don't receive
        // messages transmitted using publishers
        // on this connection. Otherwise, we'd receive our own updates.
        TopicSubscriber subscriber = myTopicSession.createSubscriber(topic,
   null, true);
        System.out.println("Listener started");
        while (isListenerRunning()) {
            ObjectMessage om = (ObjectMessage) subscriber.receive(2000);
            if (om != null) {
                // Use Session that was passed in on the initialize...
                // very important to use no write through here
                mySession.beginNoWriteThrough();
                byte[] raw = (byte[]) om.getObject();
                ByteArrayInputStream bis = new ByteArrayInputStream(raw);
                ObjectInputStream ois = new ObjectInputStream(bis);
                // inflate the LogSequences
                Collection collection = LogSequenceTransformer.inflate(ois,
       myGrid);
                Iterator iter = collection.iterator();
                while (iter.hasNext()) {
                    // process each Maps changes according to the mode when
                    // the LogSequence was serialized
                    LogSequence seq = (LogSequence) iter.next();
                    mySession.processLogSequence(seq);
                }
                mySession.commit();
            } // if there was a message
        } // while loop
        // stop the connection
        connection.close();
    } catch (IOException e) {
        System.out.println("IO Exception: " + e);
    } catch (JMSException e) {
        System.out.println("JMS Exception: " + e);
    } catch (ObjectGridException e) {
        System.out.println("ObjectGrid exception: " + e);
        System.out.println("Caused by: " + e.getCause());
    } catch (Throwable e) {
        System.out.println("Exception : " + e);
    }
    System.out.println("Listener stopped");
}
```

Programme d'écoute d'événement JMS

Le programme JMSObjectGridEventListener est conçu pour prendre en charge l'invalidation du cache local côté client et un mécanisme de réplication entre homologues. Il s'agit d'une implémentation JMS (Java Message Service) de l'interface ObjectGridEventListener.

Le mécanisme d'invalidation de client peut être utilisé dans un environnement eXtreme Scale réparti pour garantir la synchronisation des données du cache local avec les serveurs ou les autres clients. Sans cette fonction, le cache local du client pourrait contenir des données obsolètes. Toutefois, même avec ce mécanisme d'invalidation de client JMS, vous devez prendre en compte le délai de mise à jour d'un cache local client en raison du retard de la publication des mises à jour.

Le mécanisme de réplication entre homologues peut être utilisé dans les environnements eXtreme Scale répartis et locaux. Il s'agit d'un processus de réplication de coeur à coeur qui permet aux mises à jour de données de circuler parmi les ObjectGrid locales et les ObjectGrid réparties. Par exemple, avec ce mécanisme, vous pouvez transférer des mises à jour de données d'une grille répartie vers une grille locale ou d'une grille vers une autre grille d'un autre domaine système.

Le programme JMSObjectGridEventListener exige de l'utilisateur la configuration des informations JMS et JNDI (Java Naming and Directory Interface) pour obtenir les ressources JMS requises. En outre, les propriétés de réplication doivent être définies correctement. Dans un environnement JEE, les informations JNDI doivent être disponibles dans les conteneurs Web et EJB (Enterprise JavaBean). Dans ce cas, la propriété JNDI est facultative à moins que vous ne souhaitiez obtenir des ressources JMS externes.

Ce programme d'écoute d'événement comporte des propriétés que vous pouvez configurer avec le langage XML ou à l'aide de programmes et pouvant être utilisées pour l'invalidation de client, pour la réplication entre homologues ou les deux. La plupart des propriétés sont facultatives afin de personnaliser le comportement permettant d'obtenir les fonctionnalités dont vous avez besoin.

.

Pour plus d'informations, consultez l'API JMSObjectGridEventListener.

Extension du plug-in JMSObjectGridEventListener

Le plug-in JMSObjectGridEventListener permet aux instances ObjectGrid homologues de recevoir des mises à jour lorsque les données de la grille sont modifiées ou expulsées. Il permet également aux clients d'être avertis lors de la mise à jour ou de l'expulsion d'entrées d'une grille eXtreme Scale. Cette rubrique décrit l'extension du plug-in JMSObjectGridEventListener pour permettre aux applications d'obtenir une notification à réception d'un message JMS. Cette fonction est particulièrement utile lors de l'utilisation du paramètre CLIENT_SERVER_MODEL pour l'invalidation de client.

Lors d'une exécution avec le rôle récepteur, la méthode JMSObjectGridEventListener.onMessage substituée est automatiquement appelée par l'exécution eXtreme Scale lorsque l'instance JMSObjectGridEventListener reçoit des mises à jour du message JMS de la grille. Ces messages incluent une collection d'objets LogSequence. Les objets LogSequence sont transmis à la méthode onMessage et l'application utilise l'objet LogSequence pour identifier les entrées de cache qui ont été insérées, supprimées, mises à jour ou invalidées.

Pour utiliser le point d'extension onMessage, les applications suivent les étapes ci-dessous.

1. Crée une classe, en étendant la classe JMSObjectGridEventListener et en substituant la méthode onMessage.
2. Configure la classe étendue JMSObjectGridEventListener de la même manière que la classe ObjectGridEventListener pour l'ObjectGrid.

La classe étendue JMSObjectGridEventListener est un enfant de la classe JMSObjectGridEventListener et peut uniquement se substituer à deux méthodes : les méthodes initialize (facultative) et onMessage. Si une classe enfant de la classe

JMSObjectGridEventListener doit utiliser des artefacts ObjectGrid tels que ObjectGrid ou Session dans la méthode onMessage, elle peut obtenir ces artefacts dans la méthode initialize et les mettre en cache en tant que variables d'instance. De même, dans la méthode onMessage, les artefacts ObjectGrid mis en cache doivent être utilisés pour traiter une collection de LogSequences transmise.

Remarque : La méthode initialize remplacée doit appeler la méthode super.initialize pour initialiser la classe parent JMSObjectGridEventListener de manière appropriée.

Voici un exemple de classe étendue JMSObjectGridEventListener.

```
package com.ibm.websphere.samples.objectgrid.jms.price;

import java.util.*;
import com.ibm.websphere.objectgrid.*;
import com.ibm.websphere.objectgrid.plugins.LogElement;
import com.ibm.websphere.objectgrid.plugins.LogSequence;
import com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener;

public class ExtendedJMSObjectGridEventListener extends JMSObjectGridEventListener{
 protected static boolean debug = true;

    /**
     * Grille associée à ce programme d'écoute d'événement.
     */
    ObjectGrid grid;

    /**
     * Session associée à ce programme d'écoute d'événement.
     */
    Session session;

    String objectGridType;

    public List receivedLogSequenceList = new ArrayList();


 /* (non-Javadoc)
  * @see com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener
    #initialize(com.ibm.websphere.objectgrid.Session)
  */
 public void initialize(Session session) {
  // Remarque : si l'utilisation d'un artefact ObjectGrid est requise,
            // cette classe doit obtenir l'ObjectGrid
  // de l'instance Session transmise et obtenir l'ObjectMap de l'instance Session
  // pour toutes les opérations de mappe ObjectGrid transactionnelles.

  super.initialize(session);  // doit appeler la méthode initialize super.
  this.session = session; // mettez en cache l'instance de session si son utilisation est
  //   requise pour effectuer une opération de mappe.
  this.grid = session.getObjectGrid(); // obtenez ObjectGrid, si vous devez obtenir
  //   des informations d'ObjectGrid.

  if (grid.getObjectGridType() == ObjectGrid.CLIENT)
   objectGridType = "CLIENT";
  else if (grid.getObjectGridType() == ObjectGrid.SERVER)
   objectGridType = "Server";

  if (debug)
   System.out.println("ExtendedJMSObjectGridEventListener[" +
     objectGridType + "].initialize() : grid = " + this.grid);
 }

 /* (non-Javadoc)
  * @see com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener
    #onMessage(java.util.Collection)
  */
 protected void onMessage(Collection logSequences) {
  System.out.println("ExtendedJMSObjectGridEventListener[" +
    objectGridType + "].onMessage(): ");

  Iterator iter = logSequences.iterator();

  while (iter.hasNext()) {
            LogSequence seq = (LogSequence) iter.next();
```

```
      StringBuffer buffer = new StringBuffer();
      String mapName = seq.getMapName();
      int size = seq.size();
      buffer.append("\nLogSequence[mapName=" + mapName + ", size=" + size + ",
     objectGridType=" + objectGridType
        + "]: ");

      Iterator logElementIter = seq.getAllChanges();
      for (int i = seq.size() - 1; i >= 0; --i) {
       LogElement le = (LogElement) logElementIter.next();
       buffer.append(le.getType() + " -> key=" + le.getCacheEntry().getKey() + ", ");
      }
      buffer.append("\n");

      receivedLogSequenceList.add(buffer.toString());

      if (debug) {
       System.out.println("ExtendedJMSObjectGridEventListener["
      + objectGridType + "].onMessage(): " + buffer.toString());
      }
  }
 }

 public String dumpReceivedLogSequenceList() {
  String result = "";
  int size = receivedLogSequenceList.size();
  result = result + "\nExtendedJMSObjectGridEventListener[" + objectGridType
    + "]: receivedLogSequenceList size = " + size + "\n";
  for (int i = 0; i < size; i++) {
   result = result + receivedLogSequenceList.get(i) + "\n";
  }
  return result;
 }

 public String toString() {
  return "ExtendedJMSObjectGridEventListener["
    + objectGridType + " - " + this.grid + "]";
  }
 }
```

Configuration

La classe étendue JMSObjectGridEventListener doit être configurée de la même manière pour le mécanisme d'invalidation de client que pour le mécanisme de réplication entre homologues. L'exemple suivant illustre l'approche de la configuration XML.

```
<objectGrid name="PRICEGRID">
   <bean id="ObjectGridEventListener"
    className="com.ibm.websphere.samples.objectgrid.jms.
      price.ExtendedJMSObjectGridEventListener">
    <property name="invalidationModel" type="java.lang.String"
     value="CLIENT_SERVER_MODEL" description="" />
    <property name="invalidationStrategy" type="java.lang.String"
     value="INVALIDATE" description="" />
    <property name="jms_topicConnectionFactoryJndiName" type="java.lang.String"
     value="jms/TCF" description="" />
    <property name="jms_topicJndiName" type="java.lang.String"
     value="GRID.PRICEGRID" description="" />
    <property name="jms_topicName" type="java.lang.String"
     value="GRID.PRICEGRID" description="" />
    <property name="jms_userid" type="java.lang.String" value=""
     description="" />
    <property name="jms_password" type="java.lang.String" value=""
     description="" />
   </bean>
   <backingMap name="PRICE" pluginCollectionRef="PRICE"></backingMap>
      </objectGrid>
```

Remarque : Le nom de classe du bean ObjectGridEventListener est configuré à l'aide de la classe étendue JMSObjectGridEventListener avec les mêmes propriétés que la classe générique JMSObjectGridEventListener.

Configuration de règles de déploiement

Le fichier XML du descripteur de la règle de déploiement et le fichier XML du descripteur d'ObjectGrid permettent de gérer une topologie répartie. La stratégie de déploiement est codée sous la forme d'un fichier XML qui est fourni au serveur de conteneur. La règle de déploiement fournit des informations concernant les mappes, les groupe de mappes, les partitions, les fragments réplique, etc. Elle contrôle également les comportements pour le positionnement des fragments.

Configuration de déploiements répartis

Utilisez le fichier XML descripteur de la stratégie de déploiement et le fichier XML descripteur d'ObjectGrid pour gérer votre topologie.

La stratégie de déploiement est codée dans un fichier XML qui est fourni au serveur de conteneur eXtreme Scale. Le fichier XML spécifie les informations suivantes :

- Les mappes appartenant à chaque ensemble de mappes
- Le nombre de partitions
- Le nombre de fragments réplique synchrones et asynchrones

La règle de déploiement contrôle également les comportements de positionnement ci-après.

- Le nombre minimal de serveurs de conteneurs actifs avant le placement
- Le remplacement automatique des fragments perdus
- Le positionnement de chaque fragment d'une partition sur une autre machine

Les informations sur les points de contact ne sont pas préconfigurées dans l'environnement dynamique. La règle de déploiement ne contient pas de noms de serveur ou d'informations sur la topologie physique. Tous les fragments dans une grille de données sont placés automatiquement dans les serveurs de conteneur par le service de catalogue. Ce dernier utilise les contraintes définies par la règle de déploiement pour gérer automatiquement le positionnement des fragments. Ce placement automatique des fragments facilite la configuration des grilles de données volumineuses. Vous pouvez également ajouter des serveurs à votre environnement, si nécessaire.

Restriction : Dans un environnement WebSphere Application Server, une taille de groupe central de plus de 50 membres n'est pas prise en charge.

Un fichier XML descripteur de stratégie de déploiement est transmis au serveur de conteneur lors du démarrage. Une règle de déploiement doit être utilisée avec un fichier XML d'ObjectGrid. La stratégie de déploiement n'est pas nécessaire pour démarrer un serveur de conteneur, mais elle est recommandée. La règle de déploiement doit être compatible avec le fichier XML d'ObjectGrid qui lui est associé. Pour chaque élément objectgridDeployment de la règle de déploiement, vous devez inclure un élément objectGrid correspondant dans votre fichier XML d'ObjectGrid. Les mappes de l'élément objectgridDeployment doivent être cohérentes avec les éléments backingMap du XML d'ObjectGrid. Chaque mappe de sauvegarde backingMap doit être référencée sans un seul élément mapSet.

Dans l'exemple ci-après, le fichier companyGridDpReplication.xml doit être associé au fichier companyGrid.xml correspondant.

companyGridDpReplication.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org./2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

 <objectgridDeployment objectgridName="CompanyGrid">
  <mapSet name="mapSet1" numberOfPartitions="11"
   minSyncReplicas="1" maxSyncReplicas="1"
   maxAsyncReplicas="0" numInitialContainers="4">
   <map ref="Customer" />
   <map ref="Item" />
   <map ref="OrderLine" />
   <map ref="Order" />
  </mapSet>
 </objectgridDeployment>

</deploymentPolicy>
```

companyGrid.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">

 <objectGrids>
  <objectGrid name="CompanyGrid">
   <backingMap name="Customer" />
   <backingMap name="Item" />
   <backingMap name="OrderLine" />
   <backingMap name="Order" />
  </objectGrid>
 </objectGrids>

</objectGridConfig>
```

Le fichier companyGridDpReplication.xml contient un élément mapSet divisé en 11 partitions. Chaque partition doit posséder exactement une réplique synchrone. Le nombre de fragments réplique synchrones est spécifié par les attributs minSyncReplicas et maxSyncReplicas. L'attribut minSyncReplicas ayant la valeur 1, chaque partition de l'élément mapSet doit disposer d'au moins une réplique synchrone disponible pour traiter les transactions d'écriture. Etant donné que l'attribut maxSyncReplicas a la valeur 1, chaque partition ne peut dépasser une réplique synchrone. Les partitions de cet élément mapSet ne possèdent pas de fragments réplique asynchrones.

L'attribut numInitialContainers indique au service de catalogue de retarder le placement jusqu'à ce que quatre serveurs soient disponibles pour prendre en charge cette instance ObjectGrid. L'attribut numInitialContainers est ignoré lorsque le nombre de serveurs de conteneur est atteint.

7.1.1+ Vous pouvez également utiliser la propriété **placementDeferralInterval** et la commande **xscmd -c suspendBalancing** pour retarder le placement des fragments sur les serveurs de conteneur.

Bien que le fichier companyGridDpReplication.xml soit un exemple simple, une règle de déploiement peut vous permettre de contrôler intégralement votre environnement.

Topologie répartie

Les mémoires cache cohérentes réparties permettent d'améliorer les performances, la disponibilité et l'évolutivité du système, que vous pouvez configurer.

WebSphere eXtreme Scale équilibre automatiquement les serveurs. Vous pouvez inclure des serveurs supplémentaires sans redémarrer WebSphere eXtreme Scale. L'ajout de serveurs supplémentaires sans avoir à redémarrer eXtreme Scale permet

d'avoir des déploiements simples, mais également des déploiements de grande taille se chiffrant en téraoctets et comptant plusieurs milliers de serveurs.

Cette topologie de déploiement est flexible. A l'aide du service de catalogue, vous pouvez ajouter et supprimer des serveurs afin de mieux utiliser les ressources sans supprimer l'intégralité du cache. Vous pouvez utiliser les commandes **startOgServer** et **stopOgServer** pour démarrer et arrêter les serveurs de conteneur. Ces deux commandes nécessitent de spécifier l'option **-catalogServiceEndPoints**. Tous les clients d'une topologie répartie communiquent avec le service de catalogue via le protocole IIOP (Internet Interoperability Object Protocol). Tous les clients utilisent l'interface ObjectGrid pour communiquer avec les serveurs.

La fonctionnalité de configuration dynamique de WebSphere eXtreme Scale facilite l'ajout de ressources au système. Les conteneurs hébergent les données et le service de catalogue permet aux clients de communiquer avec la grille de serveurs de conteneur. Le service de catalogue transmet les demandes, alloue de l'espace dans les serveurs de conteneur hôtes et gère l'état et la disponibilité de l'ensemble du système. Les clients se connectent à un service de catalogue, extraient la description de la topologie des serveurs de conteneur et communiquent directement avec chaque serveur. Lorsque la topologie des serveurs change suite à l'ajout de serveurs ou de la défaillance d'autres serveurs, le service de catalogue achemine automatiquement les demandes client au serveur approprié qui héberge les données.

Un service de catalogue existe dans sa propre grille de machines virtuelles Java. Un même serveur de catalogues peut gérer plusieurs serveurs. Vous pouvez démarrer un serveur de conteneur dans une machine virtuelle Java seule ou le charger dans une machine virtuelle Java arbitraire avec d'autres serveurs de conteneur pour différents serveurs. Un client peut exister dans une machine virtuelle Java et communiquer avec un ou plusieurs serveurs. Un client peut également exister dans la même machine virtuelle Java qu'un serveur de conteneur.

Vous pouvez également créer une stratégie de déploiement à l'aide d'un programme lorsque vous intégrez un serveur de conteneur dans un processus ou une application Java. Pour plus d'informations, consultez la documentation de l'API DeploymentPolicy.

Contrôle du placement avec des zones

Utilisez votre stratégie de déploiement pour définir des zones. Les zones vous permettent de contrôler le placement des fragments dans WebSphere eXtreme Scale. Les zones sont un concept logique défini par l'utilisateur. Elles permettent de représenter les regroupements logiques des serveurs physiques.

Configuration de zones pour le positionnement de fragments réplique

La prise en charge des zones permet la configuration avancée du positionnement de réplique pour plusieurs centres de données. Grâce à cette fonctionnalité, des grilles de milliers de partitions peuvent être facilement gérées à l'aide de plusieurs règles de positionnement facultatives. Un centre de données peut correspondre à plusieurs étages d'un bâtiment, plusieurs bâtiments, plusieurs villes ou d'autres distinctions, selon la configuration des règles de zone.

Flexibilité des zones

Il est possible de placer des fragments dans des zones. Cette fonction permet de mieux contrôler la manière dont eXtreme Scale place les fragments dans une grille.

Les machines virtuelles Java qui hébergent un serveur eXtreme Scale peuvent être marquées à l'aide d'un identificateur de zone. Le fichier de déploiement peut désormais inclure une ou plusieurs règles de zone, qui sont associées à un type de fragment. La section suivante présente l'utilisation des zones. Pour plus d'informations, voir les informations sur la surveillance du placement des fragments avec des zones dans le *Guide d'administration*.

Les zones de positionnement contrôlent la manière dont eXtreme Scale assigne les fragments primaires et les fragments réplique pour configurer les topologies avancées.

Une machine virtuelle Java peut posséder plusieurs conteneurs mais un seul serveur. Un conteneur peut héberger plusieurs fragments d'un seul objet ObjectGrid.

Cette fonctionnalité permet de s'assurer que les fragments réplique et les fragments primaires sont placés dans différents emplacements ou zones et que leur haute disponibilité est optimale. Généralement, eXtreme Scale ne place pas de fragment principal et de fragment réplique dans les machines virtuelles Java possédant une adresse IP identique. Cette règle simple empêche généralement deux serveurs eXtreme Scale d'être placés sur le même ordinateur physique. Toutefois, vous pouvez avoir besoin d'un mécanisme plus flexible. Par exemple, vous souhaitez utiliser deux châssis lame sur lesquels *segmentés* les fragments primaires et vous voulez que le fragment réplique de chaque fragment primaire soit positionné sur le châssis de l'autre fragment primaire.

Les fragments primaires *à bandes* désignent les fragments primaires placés dans chaque zone. La réplique de chacun de ces fragments primaires est située dans la zone opposée. Par exemple, le fragment primaire 0 se trouve dans la zone A, et le fragment réplique synchronisé 0 dans la zone B. Le fragment primaire 1 se trouve dans la zone B, et le fragment réplique synchronisé 1 dans la zone A.

Dans ce cas, le nom du châssis correspond à celui de la zone. Vous pouvez également nommer les zones en fonction des étages d'un bâtiment et utiliser les zones pour vous assurer que les fragments primaires et les fragments réplique correspondant aux mêmes données se situent à des étages différents. Vous pouvez également utiliser des bâtiments et des centres de données. Des tests effectués sur les centres de données à l'aide de zones ont permis de s'assurer que les données sont répliquées de manière appropriée entre les centres de données. Si vous utilisez HTTP Session Manager pour eXtreme Scale, vous pouvez également utiliser des zones. Cette fonction vous permet de déployer une seule application Web sur les trois centres de données et de vous assurer que les sessions HTTP des utilisateurs sont répliquées dans les centres de données afin que les sessions puissent être récupérées en cas de défaillance d'un centre de données entier.

WebSphere eXtreme Scale prend en compte la nécessité de gérer une grille volumineuse dans plusieurs centres de données. Il est possible de s'assurer que les sauvegardes et les fragments primaires de la même partition sont situés dans des centres de données différents, le cas échéant. Il permet de placer tous les fragments primaires dans le centre de données 1 et tous les fragments réplique dans le centre de données 2. Il peut également permuter de manière circulaire les fragments primaires et les fragments réplique entre les deux centres de données. Les règles sont flexibles de sorte que de nombreux scénarios sont possibles. eXtreme Scale peut également gérer des milliers de serveurs. Cette fonctionnalité, combinée au positionnement automatique en fonction des centres de données, rend ces grilles

volumineuses plus économiques d'un point de vue administratif. Les administrateurs peuvent spécifier ce qu'ils veulent faire de manière simple et efficace.

En tant qu'administrateur, utilisez les zones de positionnement pour définir les emplacements des fragments primaires et des fragments réplique. Vous pouvez ainsi configurer des topologies avancées hautes performances et haute disponibilité. Vous pouvez définir une zone dans tout groupement logique de processus eXtreme Scale, comme indiqué ci-dessus : ces zones peuvent correspondre à des emplacements de stations de travail physiques, tels qu'un centre de données, un étage d'un centre de données ou un châssis lame. Vous pouvez segmenter les données dans les zones, afin de bénéficier d'une disponibilité accrue. Vous pouvez également diviser les fragments primaires et les fragments réplique en zones distinctes si un secours automatique est nécessaire.

Association d'un serveur eXtreme Scale à une zone qui n'utilise pas WebSphere Extended Deployment

Si eXtreme Scale est utilisé avec Java Standard Edition ou un serveur d'application qui n'est pas basé sur WebSphere Extended Deployment version 6.1, il est possible d'associer une JVM utilisée comme conteneur de fragments à une zone, si vous utilisez les méthodes suivantes.

Applications utilisant le script startOgServer

Le script startOgServer permet de démarrer une application eXtreme Scale lorsqu'elle n'est pas en cours d'intégration dans un serveur existant. Le paramètre **-zone** permet de spécifier la zone à utiliser pour tous les conteneurs du serveur.

Spécification de la zone lors du démarrage d'un conteneur à l'aide d'API

Association de nœuds WebSphere Extended Deployment à des zones

Si vous utilisez eXtreme Scale avec des applications WebSphere Extended Deployment Java EE, vous pouvez optimiser les groupes de nœuds WebSphere Extended Deployment pour placer les serveurs dans des zones spécifiques.

Dans eXtreme Scale, une JVM ne peut être membre que d'une seule zone. Toutefois, WebSphere autorise un nœud à faire partie de plusieurs groupes. Vous pouvez utiliser cette fonctionnalité des zones eXtreme Scale si vous vous assurez que chacun des nœuds se trouve uniquement dans un groupe de nœuds de zone.

Utilisez la syntaxe suivante pour nommer un groupe de nœuds afin de le déclarer en tant que zone : `ReplicationZone<UniqueSuffix>`. Les serveurs exécutés sur un nœud faisant partie d'un tel groupe sont inclus dans la zone spécifiée par le nom du groupe. Vous trouverez ci-dessous la description d'un exemple de topologie.

Tout d'abord, vous devez configurer quatre nœuds : node1, node2, node3 et node4, chaque nœud possédant deux serveurs. Ensuite, vous créez deux groupes de nœuds que vous nommez ReplicationZoneA et ReplicationZoneB. Vous ajoutez node1 et node2 à ReplicationZoneA , et node3 et node4 à ReplicationZoneB.

Lors du démarrage des serveurs de node1 et node2, ils font partie de ReplicationZoneA. De la même manière, les serveurs de node3 et node4 font partie de ReplicationZoneB.

Une machine virtuelle Java membre de grille vérifie l'appartenance aux zones uniquement lors du démarrage. L'ajout d'un nouveau groupe de nœuds ou la modification de l'appartenance a une incidence uniquement sur les machines virtuelles Java démarrées ou redémarrées.

Règles de zone

Une partition eXtreme Scale possède un fragment primaire et aucune réplique ou plus. Pour cet exemple, considérons les conventions d'attribution de nom suivantes pour les fragments. `P` est le fragment primaire ; `S` est une réplique synchrone et `A` une réplique asynchrone. Une règle de zone comporte trois composants :

- un nom de règle
- une liste de zones
- un indicateur inclusif ou exclusif

Le nom de zone d'un conteneur peut être spécifié comme décrit dans la documentation de «API de serveurs intégrés», à la page 413. Une règle de zone spécifie l'ensemble de zones dans lequel un fragment peut être placé. L'indicateur inclusif indique que le positionnement d'un fragment dans une zone de la liste entraîne le positionnement de tous les autres fragments dans la même liste. Un paramètre exclusif indique que chaque fragment d'une partition est placé dans une zone différente de la liste. Par exemple, si vous utilisez un paramètre exclusif et s'il existe trois fragments (un fragment primaire et deux répliques synchrones), la liste doit alors contenir trois zones.

Chaque fragment peut être associé à une règle de zone. Une règle de zone peut être partagée par deux fragments. Lorsqu'une règle est partagée, l'indicateur inclusif ou exclusif s'applique aux fragments de tout type qui partagent une règle unique.

Exemples

Vous trouverez ci-dessous des exemples illustrant les différents scénarios et la configuration de déploiement permettant d'implémenter ces derniers.

Segmentation des fragments primaires et des fragments réplique dans les zones

Vous disposez de trois châssis lame et souhaitez y répartir les fragments primaires, en plaçant une réplique synchrone dans un châssis différent du fragment primaire. Définissez chaque châssis en tant que zone en les nommant `ALPHA`, `BETA` et `GAMMA`. Exemple de syntaxe XML de déploiement :

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
 xsi:schemaLocation=
 "http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
    xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
  <objectgridDeployment objectgridName="library">
   <mapSet name="ms1" numberOfPartitions="37" minSyncReplicas="1"
    maxSyncReplicas="1" maxAsyncReplicas="0">
   <map ref="book" />
   <zoneMetadata>
    <shardMapping shard="P" zoneRuleRef="stripeZone"/>
    <shardMapping shard="S" zoneRuleRef="stripeZone"/>
    <zoneRule name ="stripeZone" exclusivePlacement="true" >
     <zone name="ALPHA" />
     <zone name="BETA" />
     <zone name="GAMMA" />
    </zoneRule>
   </zoneMetadata>
  </mapSet>
 </objectgridDeployment>
</deploymentPolicy>
```

Cette syntaxe de déploiement XML contient une grille appelée "library" (bibliothèque) qui contient une mappe unique appelée "book". Elle utilise quatre partitions avec une seule réplique synchrone. La clause des métadonnées de zone affiche la définition d'une seule règle de zone et l'association des règles de zone à des fragments. Les fragments primaires et synchrones sont associés à la règle de zone "stripeZone". La règle de zone contient les trois zones et utilise le positionnement exclusif. D'après cette règle, si le fragment primaire de la partition 0 est placé dans ALPHA, le fragment réplique de la partition 0 sera placée dans BETA ou dans GAMMA. De la même manière, les fragments primaires des autres partitions sont placés dans d'autres zones que les fragments réplique.

Réplique asynchrone dans une zone différente de celle du fragment primaire et du fragment réplique synchrone

Dans cet exemple, une connexion avec un temps d'attente élevé existe entre deux bâtiments. Vous souhaitez une haute disponibilité sans perte de données pour tous les scénarios. Toutefois, l'incidence de la réplication synchrone sur les performances entre les bâtiments nécessite un compromis. Vous souhaitez un fragment primaire avec une réplique synchrone dans un bâtiment et une réplique asynchrone dans l'autre bâtiment. Généralement, les défaillances qui se produisent sont des arrêts de JVM ou des blocages de l'ordinateur plutôt que des problèmes à grande échelle. Cette topologie permet d'éviter la perte de données en cas de défaillance normale. La perte d'un bâtiment est suffisamment rare pour qu'une perte de données soit acceptable dans ce cas-là. Vous pouvez créer deux zones, une pour chaque bâtiment. Fichier XML de déploiement :

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
  xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

 <objectgridDeployment objectgridName="library">
  <mapSet name="ms1" numberOfPartitions="13" minSyncReplicas="1"
   maxSyncReplicas="1" maxAsyncReplicas="1">
   <map ref="book" />
   <zoneMetadata>
    <shardMapping shard="P" zoneRuleRef="primarySync"/>
    <shardMapping shard="S" zoneRuleRef="primarySync"/>
    <shardMapping shard="A" zoneRuleRef="aysnc"/>
    <zoneRule name ="primarySync" exclusivePlacement="false" >
      <zone name="BldA" />
      <zone name="BldB" />
    </zoneRule>
    <zoneRule name="aysnc" exclusivePlacement="true">
      <zone name="BldA" />
      <zone name="BldB" />
    </zoneRule>
   </zoneMetadata>
  </mapSet>
 </objectgridDeployment>
</deploymentPolicy>
```

Le fragment primaire et le fragment réplique synchrone partagent une règle de zone primaySync avec un paramètre d'indicateur exclusif défini sur "false". Après le positionnement dans une zone du fragment primaire ou de son fragment réplique synchrone, l'autre est placé dans la même zone. La réplique asynchrone utilise une deuxième règle de zone avec les mêmes zones que la règle de zone primarySync mais elle utilise l'attribut **exclusivePlacement** défini sur "true". L'attribut indique qu'un fragment ne peut être placé dans une zone contenant un fragment issu d'une même partition. Par conséquent, le fragment réplique asynchrone n'est pas placé dans la même zone que le fragment primaire ou les fragments réplique synchrones.

Placement de tous les fragments primaires dans une zone et de tous les fragments réplique dans une autre

Dans ce cas, tous les fragments primaires se trouvent dans une zone spécifique et tous les fragments réplique dans une autre zone. Nous obtenons un fragment primaire et une réplique asynchrone unique. Tous les fragments réplique seront placées dans la zone A et les fragments primaires dans la zone B.

```
<?xml version="1.0" encoding="UTF-8"?>

<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation=
  "http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

 <objectgridDeployment objectgridName="library">
  <mapSet name="ms1" numberOfPartitions="13" minSyncReplicas="0"
   maxSyncReplicas="0" maxAsyncReplicas="1">
   <map ref="book" />
   <zoneMetadata>
    <shardMapping shard="P" zoneRuleRef="primaryRule"/>
    <shardMapping shard="A" zoneRuleRef="replicaRule"/>
    <zoneRule name ="primaryRule">
     <zone name="A" />
    </zoneRule>
    <zoneRule name="replicaRule">
     <zone name="B" />
      </zoneRule>
     </zoneMetadata>
    </mapSet>
  </objectgridDeployment>
</deploymentPolicy>
```

Cet exemple contient deux règles, l'une pour les fragments primaires (P), l'autre pour le fragment réplique (A).

Zones correspondant à des réseaux étendus (WAN)

Vous pouvez souhaiter déployer une instance unique de eXtreme Scale dans plusieurs bâtiments ou centres de données où les interconnexions réseau sont plus lentes. La lenteur accrue des connexions réseau entraîne la réduction de la bande passante et l'augmentation des temps d'attente pour les connexions. Dans ce mode, des partitions réseau sont plus susceptibles de se produire en raison de la congestion du réseau et d'autres facteurs. eXtreme Scale aborde cet environnement difficile de deux manières.

Signal de présence limité entre les zones

Les machines virtuelles Java assemblées en groupes centraux assurent le signal de présence entre elles. Lorsque le service de catalogue organise les machines virtuelles Java en groupes, ces derniers ne s'étendent pas aux zones. Dans ce groupe, un leader transmet les informations d'appartenance au service de catalogue. Ce dernier vérifie les défaillances signalées avant d'entreprendre une action. Pour ce faire, il tente de se connecter aux machines virtuelles Java suspectes. Si le catalogue trouve une fausse détection de défaillance, il n'entreprend pas d'action car la partition de groupe central fonctionnera à nouveau correctement après un court délai.

Le service de catalogue assurera régulièrement le signal de présence des leaders du groupe central à un rythme lent afin de gérer l'isolement du groupe central.

Routage par zone préférée

Avec le routage par zone préférée, vous pouvez définir la manière dont WebSphere eXtreme Scale envoie des transactions vers des zones.

Vous contrôlez l'insertion des fragments dans une grille de données. Voir «Configuration de zones pour le positionnement de fragments réplique», à la page 238 pour plus d'informations sur quelques scénarios de base et la configuration de votre stratégie de déploiement correspondante.

Le routage par zone préférée permet aux clients WebSphere eXtreme Scale de spécifier une préférence pour une zone particulière ou un ensemble de zones. En conséquence, les transactions client sont acheminées vers les zones préférées avant de tenter de les acheminer vers une autre zone.

Exigences concernant le routage par zone préférée

Avant d'utiliser le routage par zone préférée, vérifiez que l'application répond aux exigences de votre scénario.

Le placement de partition par conteneur est nécessaire pour pouvoir utiliser le routage par zone préférée. Cette stratégie de positionnement est adaptée aux applications stockant des données de session dans ObjectGrid. La stratégie de placement de partition par défaut de WebSphere eXtreme Scale est la `partition fixe`. Les clés sont hachées au moment de la validation de la partition, afin de déterminer quelle partition héberge la paire clé-valeur de la mappe par le biais du positionnement par partition fixe.

Le placement par conteneur affecte les données à une partition aléatoire lorsque la transaction est validée via l'objet SessionHandle. Vous devez pouvoir reconstruire l'objet SessionHandle pour récupérer vos données à partir de la grille de données.

Vous pouvez utiliser des zones pour contrôler plus précisément l'insertion des fragments primaires et de réplique dans votre domaine. L'utilisation de plusieurs zones dans le déploiement offre des avantages lorsque les données se trouvent dans plusieurs emplacements physiques. La séparation géographique des fragments primaires et des fragments de réplique est une façon de s'assurer que la perte irrémédiable d'un centre de données n'affecte pas la disponibilité des données.

Lorsque les données sont réparties dans plusieurs zones, il est probable que les clients soient également répartis dans la topologie. Le routage des clients vers leur zone locale ou leur centre de données local offre à l'évidence un avantage en terme de performance en réduisant la latence réseau. Routez les clients vers des zones locales ou des centres de données locaux lorsque cela est possible.

Configuration de votre topologie pour le routage par zone préférée

Réfléchissez au scénario suivant. Vous disposez de deux centres de données : Chicago et Londres. Pour réduire le temps de réponse des clients, vous souhaitez que les clients lisent et écrivent les données dans leur centre de données local.

Les fragments primaires doivent être placés dans chaque centre de données de sorte que les transactions puissent être écrites localement à partir de chaque emplacement. Les clients doivent avoir connaissance des zones pour pouvoir accéder à la zone locale.

Le placement par conteneur localise les nouveaux fragments primaires dans chaque conteneur qui est démarré. Des répliques sont placées en fonction des règles de zone et de placement définies par la stratégie de déploiement. Par défaut, une réplique est placée dans une zone différente de celle de son fragment primaire. Examinez la règle de déploiement suivante pour ce scénario.

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
 <objectgridDeployment objectgridName="universe">
```

```
  <mapSet name="mapSet1" placementStrategy="PER_CONTAINER"
   numberOfPartitions="3" maxAsyncReplicas="1">
   <map ref="planet" />
  </mapSet>
 </objectgridDeployment>
</deploymentPolicy>
```

Chaque conteneur qui démarre avec la stratégie de déploiement reçoit trois nouveaux fragments. Chaque fragment primaire a un fragment de réplique asynchrone. Démarrez chaque conteneur avec le nom de zone approprié. Utilisez le paramètre **-zone** si vous démarrez le conteneur avec le script **startOgServer**.

Pour un serveur de conteneur à Chicago :

- UNIX Linux

```
startOgServer.sh s1 -objectGridFile ../xml/universeGrid.xml
-deploymentPolicyFile ../xml/universeDp.xml
-catalogServiceEndPoints MyServer1.company.com:2809
-zone Chicago
```

- Windows

```
startOgServer.bat s1 -objectGridFile ../xml/universeGrid.xml
-deploymentPolicyFile ../xml/universeDp.xml
-catalogServiceEndPoints MyServer1.company.com:2809
-zone Chicago
```

Si vos conteneurs s'exécutent sur WebSphere Application Server, vous devez créer un groupe de noeuds et le nommer avec le préfixe `ReplicationZone`. Les serveurs qui s'exécutent sur les noeuds de ces groupes de noeuds sont placés dans la zone appropriée. Par exemple, les serveurs s'exécutant sur un noeud de Chicago peuvent appartenir au un groupe de noeuds nommé `ReplicationZoneChicago`.

Pour plus d'informations, voir «Configuration de zones pour le positionnement de fragments réplique», à la page 238.

Les fragments primaires de la zone Chicago ont des répliques dans la zone London. Les fragments primaires de la zone London ont des répliques dans la zone Chicago.

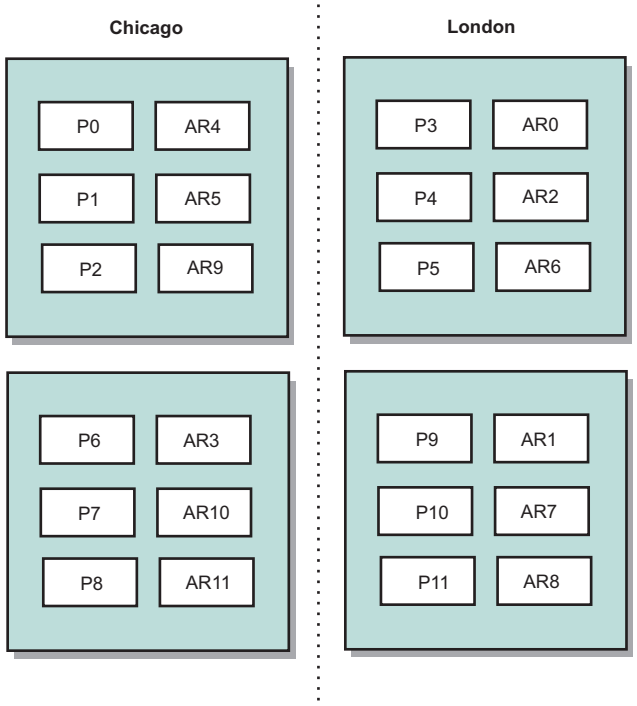


Figure 29. Segments principaux et répliques dans les zones

Définissez les zones préférées pour les clients. Fournissez un fichier de propriétés client à votre machine virtuelle Java (JVM) client. Créez le fichier `objectGridClient.properties` et veillez à le placer dans le chemin d'accès aux classes.

Incluez la propriété **`preferZones`** dans le fichier. Définissez la valeur de propriété sur la zone appropriée. Les clients dans Chicago doivent avoir la valeur suivante dans le fichier `objectGridClient.properties` :

```
preferZones=Chicago
```

Le fichier de propriétés de clients de la zone London doit contenir la valeur suivante :

```
preferZones=London
```

Cette propriété donne l'instruction à chaque client d'acheminer les transactions vers la zone locale dans la mesure du possible. La topologie réplique de manière asynchrone les données insérées dans un fragment primaire dans la zone locale dans la zone externe.

Utilisation de l'interface SessionHandle pour le routage vers la zone locale

La stratégie de placement par conteneur n'utilise pas un algorithme basé sur le hachage pour déterminer l'emplacement de vos paires clé-valeur dans la grille de données. Vous devez utiliser des objet SessionHandle pour que les transactions soient acheminées vers l'emplacement correct lorsque vous utilisez cette stratégie de placement. Lorsqu'une transaction est validée, un objet SessionHandle est lié à la session si aucun objet n'a été défini. L'objet SessionHandle peut également être

lié à la session en appelant la méthode Session.getSessionHandle avant de valider la transaction. Le fragment de code suivant montre un objet SessionHandle lié avant de valider la transaction.

```
Session ogSession = objectGrid.getSession();

// liaison du SessionHandle
SessionHandle sessionHandle = ogSession.getSessionHandle();

ogSession.begin();
ObjectMap map = ogSession.getMap("planet");
map.insert("planet1", "mercury");

// tran est acheminé vers la répartition spécifiée par SessionHandle
ogSession.commit();
```

Supposez que le code précédent a été exécuté sur un client dans le centre de données de Chicago. L'attribut **preferZones** a la valeur `Chicago` pour ce client. Par conséquent, le déploiement routera les transactions vers l'une des partitions principales dans la zone, la partition 0, 1, 2, 6, 7 ou 8.

L'objet SessionHandle fournit un chemin de retour vers la partition qui stocke ces données validées. L'objet SessionHandle doit être réutilisé ou reconstruit et défini dans la session pour revenir à la partition contenant les données validées.

```
ogSession.setSessionHandle(sessionHandle);
ogSession.begin();

// la valeur renvoyée sera "mercury"
String value = map.get("planet1");
ogSession.commit();
```

Le code de cette transaction réutilise l'objet SessionHandle qui a été créé au cours de la transaction d'insertion. La transaction get est routée vers la partition contenant les données insérées. Sans l'objet SessionHandle, la transaction ne peut pas extraire les données insérées.

Conséquences des échecs de conteneur et de zone sur le routage basé sur zones

En règle générale, un client avec la propriété **preferZones** définie route toutes les transactions vers la zone ou les zones spécifiées. Cependant, la perte d'un conteneur entraîne la promotion d'un fragment de réplique comme fragment primaire. Un client qui routait ses transactions vers les partitions de la zone locale doit extraire les données précédemment insérées à partir de la zone distante.

Imaginez le scénario suivant. Un conteneur de la zone de Chicago est perdu. Il contenait précédemment des fragments primaires pour les partitions 0, 1 et 2. Les nouveaux fragments primaires de ces partitions sont ensuite placés dans la zone London, car cette zone contenait les répliques de ces partitions.

Un client de Chicago qui utilise un objet SessionHandle qui pointe vers l'une des partitions basculées reroute maintenant leurs transactions vers London. Les clients de Chicago qui utilisent les nouveaux objets SessionHandle routent leurs transactions vers les fragments primaires de Chicago.

De même, si l'ensemble de la zone Chicago est perdue, toutes les fragments de réplique de la zone London sont promues comme fragments primaires. Dans ce scénario, tous les clients de Chicago routent leurs transactions vers London.

Définition des zones des serveurs de conteneur

Les zones sont des collections de serveurs de conteneur. Un serveur de conteneur peut appartenir uniquement à une seule zone. Un serveur de conteneur est affecté à une zone lorsqu'il démarre.

Pourquoi et quand exécuter cette tâche

Vous devez planifier vos zones avant de démarrer vos serveurs de conteneur, car les serveurs de conteneur définissent leur appartenance à une zone au démarrage. Si vous souhaitez modifier l'appartenance d'un serveur de conteneur à la zone dont il est membre, vous devez redémarrer le serveur avec les informations de la nouvelle zone.

Procédure

- **Définissez les zones des serveurs de conteneur autonomes.**
  1. Utilisez le paramètre **-zone** du script **startOgServer** pour spécifier la zone de les conteneurs dans le serveur démarré. Pour plus d'informations sur le démarrage des serveurs, voir «Script **startOgServer**», à la page 401.
  2. Vous pouvez également affecter des noms à des zones lorsque vous démarrez les serveur de conteneur à l'aide d'un programme avec l'API de serveur intégrée. Pour plus d'informations, voir «Utilisation de l'API de serveur embarqué pour démarrer et arrêter les serveurs», à la page 410.
- **Définissez des zones pour les serveurs de conteneur exécutés dans WebSphere Application Server.**

  Vous pouvez utiliser des groupes de noeuds pour placer les serveurs de conteneur dans des zones spécifiques. Utilisez la syntaxe suivante pour nommer votre groupe de noeuds pour lui affecter une zone : `ReplicationZone<identifier>`. Lorsque vous définissez des zones dans la stratégie de déploiement, vous devez nommer les zones exactement comme vous avez nommé les groupes de noeuds. Le nom du groupe de noeuds et le nom de zone dans le fichier descripteur XML de stratégie de déploiement doivent être identiques.

  Important : WebSphere Application Server n'interdit pas de placer les noeuds dans plusieurs groupes de noeuds. Etant donné que les serveurs de conteneur ne peuvent se trouver que dans une seule zone, vérifiez que vos noeuds se trouvent exactement dans un groupe de noeuds ReplicationZone.

  Par exemple, divisez quatre noeuds en deux zones, A et B.

  1. Configurez quatre nœuds, noeud 1, noeud 2, noeud 3 et noeud 4, chaque nœud possédant deux serveurs.
  2. Créez le groupe de noeuds ReplicationZoneA et le groupe de noeuds ReplicationZoneB.
  3. Ajoutez le noeud 1 et le noeud 2 à ReplicationZoneA et le noeuds 3 et 4 à ReplicationZoneB.
  4. Définissez ReplicationZoneA et ReplicationZoneB dans le fichier descripteur XML de la stratégie de déploiement. Voir «Exemple : zones dans un environnement WebSphere Application Server», à la page 251 pour un exemple.
  5. Lorsque les serveurs des noeuds 1 et 2 démarrent, ils rejoignent ReplicationZoneA, ou la zone A, dans la configuration WebSphere eXtreme Scale. Les serveurs sur les noeuds 3 et 4 rejoignent ReplicationZoneB, ou la zone B, dans la configuration WebSphere eXtreme Scale.

Exemple : Définitions de zone dans le fichier XML de descripteur de stratégie de déploiement

Vous pouvez définir des zones et des règles de zone avec le fichier XML du descripteur de stratégie de déploiement.

Exemple : fragments primaires et de réplique dans des zones différentes

Cet exemple place les fragments primaires dans une zone donnée et les fragments de réplique dans une zone différente avec une seule réplique asynchrone. Tous les fragments primaires commencent dans la zone DC1. Les fragments de réplique commencent dans la zone DC2.

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy
 ../deploymentPolicy.xsd" xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
 <objectgridDeployment objectgridName="library">
 <mapSet name="ms1" numberOfPartitions="13" minSyncReplicas="0"
  maxSyncReplicas="0" maxAsyncReplicas="1">
  <map ref="book" />
  <zoneMetadata>
    <shardMapping shard="P" zoneRuleRef="primaryRule"/>
    <shardMapping shard="A" zoneRuleRef="replicaRule"/>
    <zoneRule name ="primaryRule">
     <zone name="DC1" />
    </zoneRule>
    <zoneRule name="replicaRule">
    </zoneRule>
  </zoneMetadata>
  </mapSet>
  </objectgridDeployment>
 </deploymentPolicy>
```

Une réplique asynchrone est définie dans l'élément ms1 mapSet. Par conséquent, deux fragments existent pour chaque partition : un fragment primaire et un fragment de réplique asynchrone. Dans l'élément zoneMetadata, un élément shardMapping est défini pour chaque fragment: P pour le fragment primaire et DC1 pour le fragment de réplique asynchrone. L'attribut primaryRule spécifie la zone définie des fragments primaires, la zone DC1 simplement, et cette règle doit être utilisée pour le placement des fragments primaires. Les répliques asynchrones sont placées dans la zone DC2.

Toutefois, si la zone DC2 est perdue, les fragments de réplique deviennent indisponibles. La perte ou la défaillance d'un serveur de conteneur dans la zone DC1 peut générer une perte de données, même si une réplique a été spécifiée.

Pour faire face à cette éventualité, vous pouvez ajouter une zone ou une réplique, comme décrit dans les sections suivantes.

Exemple : ajout d'une zone, segmentation des fragments

Le code suivant configure une nouvelle zone :

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy
 ../deploymentPolicy.xsd" xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
  <objectgridDeployment objectgridName="library">
  <mapSet name="ms1" numberOfPartitions="13" minSyncReplicas="0"
   maxSyncReplicas="0" maxAsyncReplicas="1">
     <map ref="book" />
```

```
     <zoneMetadata>
      <shardMapping shard="P" zoneRuleRef="stripeRule"/>
      <shardMapping shard="A" zoneRuleRef="stripeRule"/>
      <zoneRule name="stripeRule" exclusivePlacement="true">
       <zone name="A" />
       <zone name="B" />
      <zone name="C" />
      </zoneRule>
    </zoneMetadata>
   </mapSet>
 </objectgridDeployment>
</deploymentPolicy>
```

Trois zones ont été définies dans ce code : A, B et C. Une règle de zone partagée, stripeRule, est définie à la place de règles de zone de fragments primaires et de fragments de réplique distinctes. Cette règle inclut toutes les zones, avec l'attribut exclusivePlacement affecté de la valeur `true`. La stratégie de placement eXtreme Scale garantit que les fragments primaires et de réplique se trouvent dans des zones distinctes. Avec cette segmentation du placement, les fragments primaires et de réplique sont répartis dans les deux zones conformément à la stratégie. L'ajout d'une troisième zone C permet de ne pas générer une perte de données en cas de perte d'une zone et de conserver les fragments primaire et de réplique de chaque partition. Un problème de zone entraîne la perte du fragment primaire, du fragment de réplique ou d'aucun fragment. Tout fragment perdu est remplacé à partir du fragment survivant dans une zone survivante et il est placé dans l'autres zone survivante.

Exemple : ajout d'une réplique et définition de plusieurs centres de données

Le scénario classique à deux centres de données utilise des réseaux haut débit à faible latence dans chaque centre de données, mais la latence entre les deux centres de données est élevée. Des répliques synchrones sont utilisées dans chaque centre de données où la faible latence réduit l'impact de la réplication sur les temps de réponse. La réplication asynchrone est utilisée entre les centres de données, de sorte que le réseau dont la latence est élevée n'a aucune incidence sur le temps de réponse.

```
<?xml version="1.0" encoding="UTF-8"?>
 <deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy
 ../deploymentPolicy.xsd" xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
 <objectgridDeployment objectgridName="library">
   <mapSet name="ms1" numberOfPartitions="13" minSyncReplicas="1"
   maxSyncReplicas="1" maxAsyncReplicas="1">
    <map ref="book" />
    <zoneMetadata>
     <shardMapping shard="P" zoneRuleRef="primarySync"/>
     <shardMapping shard="S" zoneRuleRef="primarySync"/>
     <shardMapping shard="A" zoneRuleRef="async"/>
    <zoneRule name ="primarySync" exclusivePlacement="false" >
      <zone name="DC1" />
      <zone name="DC2" />
     </zoneRule>
     <zoneRule name="async" exclusivePlacement="true">
     <zone name="DC1" />
      <zone name="DC2" />
     </zoneRule>
   </zoneMetadata>
  </mapSet>
  </objectgridDeployment>
 </deploymentPolicy>
```

Les fragments primaires et les fragments de réplique synchrones partagent la règle primarySync avec l'attribut exclusivePlacement affecté de la valeur `false`. L'attribut exclusivePlacement affecté de la valeur false crée une configuration avec les fragments primaires et de réplique synchrone de chaque partition placée dans la même zone. Le fragment de réplique asynchrone utilise une deuxième règle de zone avec principalement les mêmes zones que la règle de zone primarySyn. Toutefois, le fragment de réplique asynchrone utilise l'attribut exclusivePlacement affecté de la valeur `true`. L'attribut exclusivePlacement, lorsqu'il a la valeur true, signifie qu'un fragment ne peut pas être placé dans une zone contenant un fragment issu d'une même partition. En conséquence, le fragment de réplique asynchrone n'est pas placé dans la même zone que le fragment primaire ou de réplique synchrone. Il existe trois fragments par partition dans ce mapSet : un fragment primaire, un fragment de réplique synchrone et un fragment de réplique asynchrone, de sorte qu'il existe trois éléments shardMapping, un pour chaque fragment.

Si une zone est perdue, tous les fragments de réplique asynchrones sont perdus et non régénérés, car ils n'ont pas de zone distinctes. Si les fragments primaire et de réplique sont perdus, le fragment de réplique asynchrone restante devient le fragment primaire et un nouveau fragment de réplique synchrone est créé dans la zone. Les fragments primaires et de réplique sont segmentés dans chaque zone.

Avec le placement exclusif, chaque fragment a sa propre zone ; vous devez avoir suffisamment de zones pour tous les fragments à placer dans leurs propres zones. Si une règle a une zone, un seul fragment peut être placé dans la zone. Avec deux zones, vous pouvez disposer de deux segments dans la zone.

Exemple : zones dans un environnement WebSphere Application Server

Le code suivant configure une nouvelle zone :

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy
 ../deploymentPolicy.xsd" xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
  <objectgridDeployment objectgridName="library">
  <mapSet name="ms1" numberOfPartitions="13" minSyncReplicas="0"
   maxSyncReplicas="0" maxAsyncReplicas="1">
     <map ref="book" />
     <zoneMetadata>
      <shardMapping shard="P" zoneRuleRef="stripeRule"/>
      <shardMapping shard="A" zoneRuleRef="stripeRule"/>
      <zoneRule name="stripeRule" exclusivePlacement="true">
      <zone name="ReplicationZoneA" />
       <zone name="ReplicationZoneB" />
      <zone name="ReplicationZoneC" />
      </zoneRule>
    </zoneMetadata>
   </mapSet>
 </objectgridDeployment>
</deploymentPolicy>
```

Pour cet exemple, trois groupes de noeuds sont définis dans l'environnement WebSphere Application Server : ReplicationZoneA, ReplicationZoneB, and ReplicationZoneC. Le nom du groupe de noeuds et le nom de zone dans le fichier XML du descripteur de la stratégie de déploiement doit être identique et contenir le texte `ReplicationZone<identifier>`. Ce fichier définit une configuration similaire à la segmentation des fragments (par exemple, mais indique le nom requis pour une configuration WebSphere Application Server.

Affichage des informations de zone avec l'utilitaire xscmd

Vous pouvez utiliser le modèle de l'exemple d'utilitaire **xscmd** pour afficher les informations relatives à votre déploiement de zone en cours, y compris les données de placement de fragment.

Avant de commencer

- Déployez une grille de données distribuée avec plusieurs centres de données. Pour plus d'informations, voir «Routage par zone préférée», à la page 243.

Pourquoi et quand exécuter cette tâche

Vous pouvez déterminer les informations relatives à votre configuration liées aux paramètres de zone à l'aide de l'utilitaire **xscmd** fourni avec le produit.

Procédure

Utilisez l'utilitaire **xscmd** pour déterminer des informations sur les fragments de données. Exécutez la commande suivante :

```
xscmd -c showPlacement -z zone_name
```

Exemple

Vous pouvez également exécuter un scénario simple à l'aide de l'exemple d'initiation *racine_install_wxs*/ObjectGrid/gettingstarted. Pour plus d'informations, voir «Tutoriel : Démarrer avec WebSphere eXtreme Scale», à la page 1.

1. Démarrerzun serveur de catalogue :

   ```
   runcat.bat
   ```

2. Déterminez le nombre de répliques, de règles de zone, de conteneurs et les autres paramètres dont vous avez besoin, avec la commande suivante, par exemple : `startOgServer.bat serverA0 -objectgridFile xml\objectgrid.xml -deploymentPolicyFile xml\deployment.xml -zone zoneA`
3. Vous pouvez arrêter les processus de conteneur pour simuler une défaillance dans la grille de données : `stopOgServer.bat serverA0,serverA1,serverB0 -catalogServiceEndPoints localhost:2809`.

   Si le serveur qui contient le dernier fragment d'une partition est arrêté, eXtreme Scale affecte un nouveau fragment primaire. Vous pouvez rechercher les pertes de données :

   - Le script **runclient** insère et lit l'élément dans votre grille de données.
   - La commande **xscmd -c showMapSizes** indique le nombre d'éléments dans la grille de données.
4. Affichez les serveurs de conteneur actifs avec la commande suivante :

   ```
   xscmd -c showPlacement -z zone_name
   ```

Configuration de serveurs de catalogues et de serveurs de conteneurs

WebSphere eXtreme Scale comporte deux types de serveurs : les serveurs de catalogue et les serveurs conteneurs. Les serveurs de catalogues contrôlent le positionnement des fragments et détectent et surveillent les serveurs conteneurs. Ensemble, plusieurs serveurs de catalogues constituent le service de catalogue. Un serveur de conteneur est une machine virtuelle Java (JVM) qui stocke les données d'application de la grille de données.

Pourquoi et quand exécuter cette tâche

Les serveurs de catalogue et de conteneur peuvent démarrer dans des processus WebSphere Application Server en tant que processus autonomes Javas SE, ou en intégrant les serveurs dans des applications Java SE. La manière dont vous configurez les serveurs de catalogue et de conteneur dépend de votre topologie.

Serveurs de catalogue

- **Serveurs de catalogue autonomes** :

  Configurez les serveurs de catalogue autonomes avec un fichier de propriétés de serveur. Contrôlez le cycle de vie d'un serveur de catalogue avec les scripts **startOgServer** et **stopOgServer** ou à l'aide de l'API de serveur embarqué.
- **Serveurs de catalogue qui démarrent dans WebSphere Application Server:**

  Configurez les serveurs de catalogue qui s'exécutent dans WebSphere Application Server avec la console d'administration WebSphere Application Server, les tâches d'administration et les fichiers des propriétés de serveur. Le cycle de vie du serveur est contrôlé par le cycle de vie du processus dans WebSphere Application Server. Lorsque les processus démarrent ou s'arrêtent dans WebSphere Application Server, les serveurs de catalogue qui s'exécutent sur ces processus démarrent ou s'arrêtent également.

Serveurs de conteneur

- **Serveurs de catalogue autonomes** :

  Configurez les serveurs de conteneur autonomes avec un fichier de propriétés de serveur et un fichier XML de stratégie de déploiement. Contrôlez le cycle de vie d'un serveur de conteneur avec les scripts **startOgServer** et **stopOgServer** ou à l'aide de l'API de serveur embarqué.
- **Serveurs de conteneur qui démarrent dans WebSphere Application Server :**

  Configurez les serveurs de conteneur dans WebSphere Application Server avec un fichier de propriétés de serveur et un fichier XML de stratégie de déploiement intégré au module d'application Java EE. Le cycle de vie des serveurs de conteneurs est contrôlé par l'application. Les serveurs de conteneur démarrent et s'arrêtent avec l'application.

Utilisez les rubriques suivantes pour configurer les serveurs de catalogue et de conteneur :

Meilleure pratique : Mise en cluster du service de catalogue avec les domaines de services de catalogue

Lorsque vous utilisez le service de catalogue, un minimum de deux serveurs de catalogue sont requis pour éviter un point de défaillance unique. Selon le nombre de noeuds dans votre environnement, vous pouvez créer des configurations différentes pour qu'au moins deux serveurs de catalogue soient toujours en cours d'exécution.

Nombre de serveurs de catalogue

La meilleure pratique pour éviter un point de défaillance unique pour votre domaine de services de catalogue consiste à démarrer un minimum de trois serveurs de catalogue sur trois noeuds différents.

Si vous utilisez seulement deux noeuds, configurez deux serveurs de catalogue sur chacun des deux noeuds pour un total de quatre processus serveur de catalogue.

La création de cette configuration garantit que lorsqu'un seul des noeuds est démarré, les deux serveurs de catalogue nécessaires sont actifs. Vous devez démarrer au moins deux serveurs de catalogue en même temps. Lorsque les serveurs de catalogue démarrent, ils recherchent d'autres serveurs de catalogue dans la configuration et ne démarrent pas tant qu'un autre serveur de catalogue au moins n'est pas trouvé.

Exemple : Démarrage de quatre serveurs de catalogue sur deux noeuds dans un environnement autonome

Le script suivant démarre les serveurs de catalogue cs0 et cs1 sur le noeud hôte1 et démarre les serveurs de catalogue cs2 et cs3 sur le noeud host2.

```
./startOgServer.sh|bat cs0 -listenerPort 2809 -catalogServiceEndPoints
cs0:host1:6601:6602,cs1:host1:6603:6604,cs2:host2:6601:6602,cs3:host2:6603:6604
-quorum true -jvmArgs -Xmx256m

./startOgServer.sh|bat cs1 -listenerPort 2810 -catalogServiceEndPoints
cs0:host1:6601:6602,cs1:host1:6603:6604,cs2:host2:6601:6602,cs3:host2:6603:6604
-quorum true -jvmArgs -Xmx256m

./startOgServer.sh|bat cs2 -listenerPort 2809 -catalogServiceEndPoints
cs0:host1:6601:6602,cs1:host1:6603:6604,cs2:host2:6601:6602,cs3:host2:6603:6604
-quorum true -jvmArgs -Xmx256m

./startOgServer.sh|bat cs3 -listenerPort 2810 -catalogServiceEndPoints
cs0:host1:6601:6602,cs1:host1:6603:6604,cs2:host2:6601:6602,cs3:host2:6603:6604
-quorum true -jvmArgs -Xmx256m
```

A faire : Vous devez utiliser l'option **-listenerPort** , car les serveurs de catalogue qui s'exécutent sur un noeud requièrent chacun un numéro de port unique.

Exemple : Démarrage de plusieurs serveurs de catalogue dans un environnement WebSphere Application Server

Les serveurs de catalogue démarrent automatiquement dans un environnement WebSphere Application Server. Vous pouvez définir plusieurs serveurs de catalogues afin de commencer en créant un domaine de services de catalogue. Une fois que vous avez indiqué plusieurs noeuds finaux dans le domaine de services de catalogue, redémarrez les serveurs d'applications inclus afin que les serveurs de catalogue démarrent en parallèle.

- **WebSphere Application Server Network Deployment** : vous pouvez choisir plusieurs serveurs d'applications existants de la cellule pour les placer dans votre domaine de services de catalogue.
- **Base WebSphere Application Server** : vous pouvez démarrer le service de catalogue sur plusieurs noeuds autonomes. En définissant plusieurs profils dans la même image d'installation à l'aide de l'outil de gestion des profils, vous pouvez créer un ensemble de noeuds autonomes ayant chacun des ports uniques. Dans chaque serveur d'applications, définissez le domaine de services de catalogue. Vous pouvez spécifier n'importe quels autres serveurs d'applications en ajoutant des serveurs distants à la configuration. Après avoir créé cette configuration sur tous les serveurs autonomes, vous pouvez démarrer l'ensemble de serveurs d'applications de base en parallèle en exécutant le script **startServer** ou en utilisant un service Windows pour démarrer les serveurs.

Optimisation de la valeur de l'intervalle des pulsations pour la détection des basculements

Le paramètre d'intervalle du signal de présence permet de configurer le laps de temps séparant deux vérifications par le système des serveurs en panne.

Pourquoi et quand exécuter cette tâche

La configuration des basculements varie en fonction du type d'environnement que vous utilisez. Si vous utilisez un environnement autonome, vous pouvez configurer les basculements à l'aide de la ligne de commande. Si vous utilisez un environnement WebSphere Application Server Network Deployment, vous devez les configurer à partir de la console d'administration de WebSphere Application Server Network Deployment.

Procédure

- Configurez les basculements pour les environnements autonomes.

  Vous pouvez configurer les intervalles des pulsations sur la ligne de commande à l'aide du paramètre **-heartbeat** dans le fichier de script **startOgServer**. Affectez à ce paramètre l'une des valeurs suivantes :

  Tableau 13. Intervalles de signal de présence

Valeur	Action	Description
0	Standard (par défaut)	Les basculements sont généralement détectés dans les 30 secondes.
-1	Elevé	Les basculements sont généralement détectés dans les 5 secondes.
1	Souple	Les basculements sont généralement détectés dans les 180 secondes.

  Un intervalle élevé entre les signaux de présence peut être utile si les processus et le réseau sont stables. Si le réseau ou les processus ne sont pas configurés de manière optimale, il peut manquer des signaux de présence, ce qui peut fausser la détection des incidents.

- Configurez les basculements pour les environnements WebSphere Application Server.

  Vous pouvez configurer WebSphere Application Server Network Deployment Version 6.0.2 ou ultérieure pour permettre des basculements très rapides de WebSphere eXtreme Scale. La durée par défaut de pour les incidents matériels est d'environ 200 secondes. Un incident matériel est un ordinateur physique, une panne du serveur, déconnexion de câble réseau ou une erreur du système d'exploitation. Les incidents dus aux pannes de processus ou à des échecs logiciels sont généralement basculés en moins d'une seconde. La détection des incidents logiciels est effectuée lorsque les sockets réseau du processus inactif sont fermés automatiquement par le système d'exploitation du serveur qui héberge le processus.

  Configuration des signaux de présence du groupe central

  Si WebSphere eXtreme Scale est exécuté dans un processus WebSphere Application Server, il hérite des caractéristiques de reprise en ligne des paramètres du groupe central du serveur d'applications. Les sections suivantes décrivent comment configurer les paramètres des signaux de présence du groupe central pour différentes versions de WebSphere Application Server Network Deployment :

– **Mise à jour des paramètres des groupes centraux de WebSphere Application Server Network Deployment Version 6.x et 7.x :**

  Spécifiez l'intervalle des signaux de présence en secondes sur les versions 6.0 à 6.1.0.12 de WebSphere Application Server ou en millisecondes à partir de la version 6.1.0.13. Vous devez également spécifier le nombre de signaux de présence manqués. Cette valeur indique le nombre maximal de signaux de présence manquants avant qu'une machine virtuelle Java (JVM) ne soit considérée comme défectueuse. Le délai de détection des incidents matériels est approximativement égal au produit de l'intervalle des signaux de présence par le nombre de signaux de présence manqués.

  Ces propriétés sont spécifiées à l'aide des propriétés personnalisées sur le groupe central à l'aide de la console d'administration de WebSphere. Pour des informations de configuration détaillées, voir la rubrique Propriétés personnalisées de groupe central. Ces propriétés doivent être spécifiées pour tous les groupes centraux utilisés par l'application :

  - L'intervalle des pulsations est spécifié à l'aide de la propriété personnalisée IBM_CS_FD_PERIOD_SEC pour les secondes ou de la propriété personnalisée IBM_CS_FD_PERIOD_MILLIS pour les millisecondes (nécessite la version 6.1.0.13 ou une version ultérieure).
  - Le nombre de signaux de présence manqués est spécifié à l'aide de la propriété personnalisée IBM_CS_FD_CONSECUTIVE_MISSED.

  La valeur par défaut de la propriété IBM_CS_FD_PERIOD_SEC est de 20 et celle de la propriété IBM_CS_FD_CONSECUTIVE_MISSED, de 10. Si la propriété IBM_CS_FD_PERIOD_MILLIS est spécifiée, elle remplace les propriétés personnalisées IBM_CS_FD_PERIOD_SEC définies. Les valeurs de ces propriétés correspondent à des entiers.

  Utilisez les paramètres suivants pour spécifier un délai de détection des incidents de 1500 ms pour les serveurs WebSphere Application Server Network Deployment Version 6.x :

  - Spécifiez `IBM_CS_FD_PERIOD_MILLIS = 750` (WebSphere Application Server Network Deployment V6.1.0.13 et versions ultérieures)
  - Spécifiez `IBM_CS_FD_CONSECUTIVE_MISSED = 2`

– **Mise à jour des paramètres des groupes centraux de WebSphere Application Server Network Deployment Version 7.0**

  WebSphere Application Server Network Deployment Version 7.0 fournit deux paramètres de groupe central qui peuvent être ajustés pour augmenter ou réduire le délai de détection des incidents :

  - **Période de transmission du signal de présence.** La valeur par défaut est de 30000 millisecondes.
  - **Période d'expiration du signal de présence.** La valeur par défaut est de 180000 millisecondes.

  Pour plus de détails sur la manière de modifier ces paramètres, voir la rubrique relative à la WebSphere Application Server Network Deployment reconnaissance et de détection des incidents dans le centre de documentation.

  Utilisez les paramètres suivants pour spécifier un délai de détection des incidents de 1500 ms pour les serveurs WebSphere Application Server Network Deployment Version 7 :

  - Spécifiez une période de transmission du signal de présence de 750 millisecondes.
  - Spécifiez une période d'expiration du signal de présence de 1500 millisecondes.

Que faire ensuite

Lorsque vous modifiez ces paramètres pour réduire les délais de basculement, certains points d'optimisation du système sont à prendre en compte. Tout d'abord, Java n'est pas un environnement en temps réel. Des unités d'exécution peuvent être retardées si la JVM connaît des délais de récupération de place importants. Les unités d'exécution risquent également d'être retardées si la charge de la machine qui héberge la JVM est considérable (à cause de la JVM elle-même ou d'autres processus exécutés sur cette machine). Si les unités d'exécution sont retardées, les signaux de présence risquent de ne pas être envoyés à temps. Au pire, ils risquent d'être retardés du délai requis pour la reprise en ligne. Si des unités d'exécution sont retardées, des incidents sont détectés à tort. Le système doit être optimisé et dimensionné de sorte à éviter la détection de faux incidents en production. Il est recommandé pour cela de tester la charge de manière adéquate.

Remarque : La version actuelle d'eXtreme Scale prend en charge WebSphere Real Time.

Configuration de WebSphere eXtreme Scale avec WebSphere Application Server

Vous pouvez exécuter les processus des services de catalogue et des serveurs conteneurs dans WebSphere Application Server. La procédure de configuration de ces serveurs est différente d'une configuration autonome. Je service de catalogue peut être automatiquement démarré dans des serveurs ou des gestionnaires de déploiement WebSphere Application Server. Le processus de conteneur démarre lorsqu'une application eXtreme Scale est déployée et démarrée dans l'environnement WebSphere Application Server.

Pourquoi et quand exécuter cette tâche

Avertissement : Ne placez pas les serveurs de conteneur avec les serveurs de catalogue dans un environnement de production. Incluez le service de catalogue dans plusieurs processus d'agents de noeud ou sur un serveur d'applications qui n'héberge pas d'application eXtreme Scale.

Configuration du service de catalogue dans WebSphere Application Server

Les processus de service de catalogue peuvent s'exécuter dans WebSphere Application Server. Le cycle de vie du serveur dans WebSphere Application Server détermine quand le service de catalogue démarre et s'arrête.

Procédure

1. Choisissez un ou plusieurs processus WebSphere Application Server pour l'extension avec le profil WebSphere eXtreme Scale. Pour plus d'informations, voir «Création et augmentation de profils pour WebSphere eXtreme Scale», à la page 184. Si vous voulez que le service de catalogue démarre automatiquement dans WebSphere Application Server Network Deployment sur le gestionnaire de déploiement, installez WebSphere eXtreme Scale sur le noeud du gestionnaire de déploiement et étendez le profil du gestionnaire de déploiement.
2. Configurez les fichiers de propriétés du serveur pour les processus WebSphere Application Server et ajoutez-les au chemin d'accès aux classes du noeud. Pour plus d'informations, voir Fichier de propriétés du serveur.

3. Configurez un domaine de services de catalogue. Le domaine de services de catalogue est un groupe de serveurs de catalogue dans votre environnement. Pour plus d'informations, voir «Création de domaines de services de catalogue dans WebSphere Application Server».
4. Démarrez les processus WebSphere Application Server qui hébergent les serveurs de catalogue. Pour plus d'informations, voir «Démarrage et arrêt des serveurs dans un environnement WebSphere Application Server», à la page 409.

Création de domaines de services de catalogue dans WebSphere Application Server :

Un domaine de services de catalogue définit un groupe de serveurs de catalogue qui gèrent le positionnement des fragments et qui surveillent l'état des serveurs conteneurs de la grille de données.

Avant de commencer

- Installer WebSphere eXtreme Scale sur WebSphere Application Server. Pour plus d'informations, voir «Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client avec WebSphere Application Server», à la page 163.

Pourquoi et quand exécuter cette tâche

En créant un domaine de services de catalogue, vous définissez une collection de serveurs de catalogue à haute disponibilité.

Ces serveurs de catalogue peuvent s'exécuter dans WebSphere Application Server au sein d'une simple cellule ou d'un groupe central. Le domaine de services de catalogue peut également définir un groupe distant de serveurs qui s'exécutent dans différents processus Java SE ou dans d'autres cellules WebSphere Application Server.

Pour les serveurs de catalogue exécutés sur les serveurs d'applications existants de la cellule : Lorsque vous définissez un domaine de services de catalogue qui place les serveurs de catalogue sur les serveurs d'applications dans la cellule, les mécanismes des groupes centraux de WebSphere Application Server sont utilisés. Le service de catalogue démarre automatiquement sur les serveurs d'applications de la cellule. Il en résulte que les membres d'un même domaine de services de catalogue ne peuvent donc pas s'étendre au-delà des limites d'un groupe central et qu'un domaine ne peut donc pas s'étendre sur plusieurs cellules. Toutefois, les serveurs de conteneurs WebSphere eXtreme Scale et les clients peuvent s'étendre à plusieurs cellules en se connectant à un serveur de catalogue dans les limites de cellule (domaine de services de catalogue autonome ou intégré dans une autre cellule, par exemple).

Pour les serveurs de catalogues distants : vous pouvez connecter les conteneurs et les clients WebSphere eXtreme Scale à un domaine de service de catalogue exécuté dans une autres cellule WebSphere Application Server ou comme processus autonome. Comme les serveurs de catalogue configuré à distance ne démarrent pas automatiquement dans la cellule, vous devez démarrer manuellement les serveurs de catalogue configurés à distance. Lorsque vous configurez un domaine de services de catalogue distant, le nom de domaine doit correspondre au nom de domaine que vous avez défini lorsque vous démarrez les serveurs de catalogue distants. Le nom de domaine de services de catalogue par défaut des serveurs de catalogue autonome est `DefaultDomain`. Définissez un nom de domaine de services de catalogue avec la commande **`startOgServer`** et le paramètre **`-domain`**, un fichier de propriétés de serveur ou avec l'API de serveur embarqué. Vous devez démarrer

chaque processus de serveur de catalogue distant dans le domaine distant avec le même nom de domaine. Voir «Démarrage d'un service de catalogue autonome», à la page 395 pour plus d'informations sur le démarrage des serveurs de catalogue.

Avertissement : En environnement de production, ne faites pas cohabiter les services de catalogue avec des serveurs conteneurs WebSphere eXtreme Scale. Incluez le service de catalogue dans plusieurs processus d'agent de noeud ou dans sur un serveur d'applications qui n'héberge pas d'application WebSphere eXtreme Scale.

Procédure

1. Créez le domaine de services de catalogue.
   a. Dans la console d'administration de WebSphere Application Server, cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **Domaines de services de catalogue** > **Nouveau**.
   b. Définissez un nom, une valeur par défaut et des justificatifs d'identification pour l'authentification JMX de votre domaine. Si vous configurez des noeuds finaux distants pour le domaine de services de catalogue, le nom de ce dernier doit correspondre au nom du domaine de services de catalogue que vous définissez lorsque vous démarrez les serveurs de catalogue.
   c. Ajoutez des points de contact de serveurs de catalogue. Vous pouvez sélectionner des serveurs d'applications existants ou ajouter des serveurs distants qui exécutent un service de catalogue.
2. Testez la connexion aux serveurs de catalogue dans le domaine de services de catalogue. Pour les serveurs d'applications existants, les serveurs de catalogue démarrent lorsque le serveur d'applications associé est démarré. Pour les serveurs d'applications distants, vous devez démarrer les serveurs manuellement en utilisant la commande **startOgServer** ou l'API de serveur embarqué.
   a. Dans la console d'administration de WebSphere Application Server, cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **Domaines de services de catalogue**.
   b. Sélectionnez le domaine que vous voulez tester et cliquez sur **Tester la connexion**. Lorsque vous cliquez sur ce bouton, tous les points de contact des domaines de services de catalogue définis sont interrogés l'un après l'autre (s'il existe des points de contact) et la procédure retourne un message indiquant que la connexion au domaine a réussi.

Tâches d'administration des domaines de services de catalogue :

Les langages de script Jacl ou Jython permettent de gérer les domaines de services de catalogue présents dans votre configuration WebSphere Application Server.

Conditions requises

WebSphere eXtreme Scale Client doit être installé dans votre environnement WebSphere Application Server.

Afficher la liste de toutes les tâches d'administration

Pour obtenir la liste de toutes les tâches d'administration associées aux domaines de services de catalogue, exécutez la commande suivante avec wsadmin :

```
wsadmin>$AdminTask help XSDomainManagement
```

Commandes

Les tâches d'administration de domaines de services de catalogue comprennent les commandes suivantes :

- «createXSDomain»
- «deleteXSDomain», à la page 263
- «getDefaultXSDomain», à la page 263
- «listXSDomains», à la page 264
- «modifyXSDomain», à la page 264
- «testXSDomainConnection», à la page 269
- «testXSServerConnection», à la page 269

createXSDomain

La commande **createXSDomain** enregistre un nouveau domaine de services de catalogue.

Tableau 14. Arguments de la commande createXSDomain

Argument	Description
-name (requis)	Spécifie le nom du domaine de services de catalogue à créer.
-default	Spécifie si le domaine de services de catalogue est le domaine par défaut de la cellule. La valeur par défaut est `true`. (booléen : a soit la valeur `true`, soit la valeur `false`).
-properties	Spécifie les propriétés personnalisées du domaine de service de catalogue.

Tableau 15. Arguments de la procédure defineDomainServers

Argument	Description
name_of_endpoint	Spécifie le nom du point de contact du service de catalogue. • **Pour les serveurs d'applications existants** : le nom du noeud final doit avoir le format *cell_name**node_name**server_name* • **Pour les serveurs distantes** : définit le nom d'hôte du serveur distant. Vous pouvez utiliser le même nom pour plusieurs noeuds finaux, mais les valeurs de port client doivent être uniques pour chaque noeud final.
custom_properties	Spécifie les propriétés personnalisées du point de contact du domaine de services de catalogue. Si vous ne disposez pas de propriétés personnalisées, utilisez des guillemets doubles (`""`) pour cet argument.

Tableau 15. Arguments de la procédure defineDomainServers (suite)

Argument	Description
endpoint_ports	Spécifie les numéros de port du point de contact du domaine de services de catalogue. Les ports doivent être définis dans l'ordre suivant : `<client_port>,<listener_port>` **Port client** Indique le port utilisé pour la communication entre les serveurs de catalogue et le domaine de services de catalogue. Cette valeur est nécessaire pour les serveurs de catalogue qui s'exécutent uniquement dans des processus WebSphere Application Server et elle peut correspondre à n'importe quel port inutilisé autre part. **Port d'écoute** Indique le port utilisé pour établir des communications avec les clients. Cette valeur est obligatoire pour les noeuds finaux distants et elle doit correspondre à la valeur utilisée au démarrage du service de catalogue. Le port d'écoute est utilisé par les clients et les conteneurs pour communiquer avec le service de catalogue. **Pour les noeuds finaux distants WebSphere eXtreme Scale** : définit le port d'écoute ORB (Object Request Broker) qui permet aux conteneurs et aux clients de communiquer avec le service de catalogue via l'ORB. Pour les noeuds finaux WebSphere Application Server, la valeur de port d'écoute est facultative, car elle est héritée de la configuration de port BOOTSTRAP_ADDRESS.

Tableau 16. Arguments de la procédure configureClientSecurity

Argument	Description
-securityEnabled	Spécifie que la sécurité du client est activée pour le serveur de catalogue. Le fichier des propriétés du serveur qui est associé au serveur de catalogue sélectionné doit avoir un paramètre **securityEnabled** correspondant dans le fichier des propriétés du serveur. Si ces paramètres ne correspondent pas, une exception est générée. (booléen : a soit la valeur `true`, soit la valeur `false`).

Tableau 16. Arguments de la procédure configureClientSecurity (suite)

Argument	Description
-credentialAuthentication (facultatif)	Indique si l'authentification des données d'identification est imposée ou prise en charge. **Jamais** Aucune authentification de certificat client n'est imposée. **Requis** L'authentification des données d'identification est toujours imposée. Si le serveur ne prend pas en charge l'authentification des données d'identification, le client ne peut pas se connecter au serveur. **Prise en charge** (Par défaut) L'authentification des données d'identification est imposée seulement si à la fois le client et le serveur prennent en charge l'authentification des données d'identification.
-authenticationRetryCount (facultatif)	Spécifie le nombre de tentatives d'authentification si les données d'identification sont arrivées à expiration. Si vous ne voulez pas réessayer l'authentification, définissez la valeur à `0`. La valeur par défaut est `0`.
-credentialGeneratorClass	Indique la classe d'implémentation de com.ibm.websphere.objectgrid.security.plugins.builtins.WSTokenCredentialGenerator pour que le client récupère les jetons de sécurité de l'unité d'exécution.
-credentialGeneratorProps	Spécifie les propriétés de la classe d'implémentation CredentialGenerator. Les propriétés sont envoyées à l'objet avec la méthode setProperties(String). La valeur des propriétés du générateur de données d'identification est utilisée seulement si une valeur est spécifiée pour la zone **Classe du générateur de données d'identification**.

Valeur retournée :

Exemples de mode de traitement par lots

Le mode de traitement par lots impose de formater correctement l'entrée de commande. Utilisez le mode interactif pour que les valeurs que vous entrez soient correctement traitées. Lorsque vous utilisez le mode de traitement par lots, vous devez définir les arguments d'étape **-defineDomainServers** en utilisant un tableau de propriétés spécifiques. Ce tableau a le format *name_of_endpoint custom_properties endpoint_ports*. La valeur *endpoint_ports* est la liste des ports qui doivent être définis dans l'ordre suivant : *<client_port>,<listener_port>*.

- Créez un domaine de services de catalogue de noeuds finaux distants en utilisant Jacl :

```
$AdminTask createXSDomain {-name TestDomain -default true -defineDomainServers
{{xhost1.ibm.com "" ,2809}} -configureClientSecurity {-securityEnabled false
-credentialAuthentication Required -authenticationRetryCount 0 -credentialGeneratorClass
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator
-credentialGeneratorProps "manager manager1"}}}
```

- Créez un domaine de services de catalogue de noeuds finaux distants en utilisant la chaîne Jython :

```
AdminTask.createXSDomain('[-name TestDomain -default true
-defineDomainServers [[xhost1.ibm.com "" ,2809]
[xhost2.ibm.com "" ,2809]] -configureClientSecurity [-securityEnabled false
-credentialAuthentication Required -authenticationRetryCount 0 -credentialGeneratorClass
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator
-credentialGeneratorProps "manager manager1"] ]')
```

- réez un domaine de services de catalogue de noeuds finaux de serveur d'applications existants en utilisant Jacl :

```
$AdminTask createXSDomain {-name TestDomain -default true -defineDomainServers
 {{cellName/nodeName/serverName "" 1109}}}
```

Exemples de mode interactif

- Jacl :

```
$AdminTask createXSDomain {-interactive}
```

- Chaîne Jython :

```
AdminTask.createXSDomain ('[-interactive]')
```

deleteXSDomain

La commande **deleteXSDomain** supprime un domaine de services de catalogue.

Paramètres requis :

-name
Spécifie le nom du domaine de services de catalogue à supprimer.

Valeur retournée :

Exemples de mode de traitement par lots

- Jacl :

```
$AdminTask deleteXSDomain {-name TestDomain }
```

- Chaîne Jython :

```
AdminTask.deleteXSDomain('[-name TestDomain ]')
```

Exemples de mode interactif

- Jacl :

```
$AdminTask deleteXSDomain {-interactive}
```

- Chaîne Jython :

```
AdminTask.deleteXSDomain ('[-interactive]')
```

getDefaultXSDomain

La commande **getDefaultXSDomain** retourne le domaine de services de catalogue par défaut de la cellule.

Paramètres requis : aucun.

Valeur de retour : nom du domaine de services de catalogue.

Exemples de mode de traitement par lots

- Jacl :

```
$AdminTask getDefaultXSDomain
```

- Chaîne Jython :

```
AdminTask.getDefaultXSDomain
```

Exemples de mode interactif

- Jacl :

```
$AdminTask getDefaultXSDomain {-interactive}
```

- Chaîne Jython :

```
AdminTask.getDefaultXSDomain ('[-interactive]')
```

listXSDomains

La commande **listXSDomains** retourne la liste des domaine de services de catalogue existants.

Paramètres requis : aucun.

Valeur de retour : liste de tous les domaine de services de catalogue dans la cellule.

Exemples de mode de traitement par lots

- Jacl :

```
$AdminTask listXSDomains
```

- Chaîne Jython :

```
AdminTask.listXSDomains
```

Exemples de mode interactif

- Jacl :

```
$AdminTask listXSDomains {-interactive}
```

- Chaîne Jython :

```
AdminTask.listXSDomains ('[-interactive]')
```

modifyXSDomain

La commande **modifyXSDomain** modifie un domaine de services de catalogue existant.

Le mode de traitement par lots impose de formater correctement l'entrée de commande. Utilisez le mode interactif pour que les valeurs que vous entrez soient correctement traitées. Lorsque vous utilisez le mode de traitement par lots, vous devez définir les arguments d'étape **-modifyEndpoints**, **-addEndpoints** et **-removeEndpoints** en utilisant un tableau de propriétés spécifiques. Ce tableau a le format *name_of_endpoint host_name custom_properties endpoint_ports*. La valeur *endpoint_ports* est la liste des ports qui doivent être définis dans l'ordre suivant : <*client_port*>,<*listener_port*>.

Tableau 17. Arguments de la commande modifyXSDomain

Argument	Description
-name (requis)	Spécifie le nom du domaine de services de catalogue que vous souhaitez éditer.
-default	Avec la valeur `true`, spécifie que le domaine de services de catalogue est le domaine par défaut de la cellule (booléen).
-properties	Spécifie les propriétés personnalisées du domaine de services de catalogue.

Tableau 18. Arguments de la procédure modifyEndpoints

Argument	Description
name_of_endpoint	Spécifie le nom du point de contact du service de catalogue. • **Pour les serveurs d'applications existants** : le nom du noeud final doit avoir le format *cell_name**node_name**server_name* • **Pour les serveurs distantes** : définit le nom d'hôte du serveur distant. Vous pouvez utiliser le même nom pour plusieurs noeuds finaux, mais les valeurs de port client doivent être uniques pour chaque noeud final.
endpoint_ports	Spécifie les numéros de port du point de contact du domaine de services de catalogue. Les noeuds finaux doivent être définis dans l'ordre suivant : *<client_port>*,*<listener_port>* **Port client** Indique le port utilisé pour la communication entre les serveurs de catalogue et le domaine de services de catalogue. Cette valeur est nécessaire pour les serveurs de catalogue qui s'exécutent uniquement dans des processus WebSphere Application Server et elle peut correspondre à n'importe quel port inutilisé autre part. **Port d'écoute** Indique le port utilisé pour établir des communications avec les clients. Cette valeur est obligatoire pour les noeuds finaux distants et elle doit correspondre à la valeur utilisée au démarrage du service de catalogue. Le port d'écoute est utilisé par les clients et les conteneurs pour communiquer avec le service de catalogue. **Pour les noeuds finaux distants WebSphere eXtreme Scale** : définit le port d'écoute ORB (Object Request Broker) qui permet aux conteneurs et aux clients de communiquer avec le service de catalogue via l'ORB. Pour les noeuds finaux WebSphere Application Server, la valeur de port d'écoute est facultative, car elle est héritée de la configuration de port BOOTSTRAP_ADDRESS.

Tableau 19. Arguments de la procédure addEndpoints

Argument	Description
name_of_endpoint	Spécifie le nom du point de contact du service de catalogue. • **Pour les serveurs d'applications existants** : le nom du noeud final doit avoir le format *cell_name**node_name**server_name* • **Pour les serveurs distantes** : définit le nom d'hôte du serveur distant. Vous pouvez utiliser le même nom pour plusieurs noeuds finaux, mais les valeurs de port client doivent être uniques pour chaque noeud final.
custom_properties	Spécifie les propriétés personnalisées du point de contact du domaine de services de catalogue. Si vous ne disposez pas de propriétés personnalisées, utilisez des guillemets doubles ("") pour cet argument.

Tableau 19. Arguments de la procédure addEndpoints (suite)

Argument	Description
endpoint_ports	Spécifie les numéros de port du point de contact du domaine de services de catalogue. Les noeuds finaux doivent être définis dans l'ordre suivant : `<client_port>,<listener_port>` **Port client** Indique le port utilisé pour la communication entre les serveurs de catalogue et le domaine de service de catalogue. Cette valeur est nécessaire pour les serveurs de catalogue qui s'exécutent uniquement dans des processus WebSphere Application Server et elle peut correspondre à n'importe quel port inutilisé autre part. **Port d'écoute** Indique le port utilisé pour établir des communications avec les clients. Cette valeur est obligatoire pour les noeuds finaux distants et elle doit correspondre à la valeur utilisée au démarrage du service de catalogue. Le port d'écoute est utilisé par les clients et les conteneurs pour communiquer avec le service de catalogue. **Pour les noeuds finaux distants WebSphere eXtreme Scale** : définit le port d'écoute ORB (Object Request Broker) qui permet aux conteneurs et aux clients de communiquer avec le service de catalogue via l'ORB. Pour les noeuds finaux WebSphere Application Server, la valeur de port d'écoute est facultative, car elle est héritée de la configuration de port BOOTSTRAP_ADDRESS.

Tableau 20. Arguments de la procédure removeEndpoints

Argument	Description
name_of_endpoint	Spécifie le nom du point de contact de domaine de services de catalogue à supprimer.

Tableau 21. Arguments de la procédure configureClientSecurity

Argument	Description
-securityEnabled	Spécifie que la sécurité du client est activée pour le serveur de catalogue. Le fichier des propriétés du serveur qui est associé au serveur de catalogue sélectionné doit avoir un paramètre **securityEnabled** correspondant dans le fichier des propriétés du serveur. Si ces paramètres ne correspondent pas, une exception est générée. (booléen : a soit la valeur `true`, soit la valeur `false`).
-credentialAuthentication (facultatif)	Indique si l'authentification des données d'identification est imposée ou prise en charge. **Jamais** Aucune authentification de certificat client n'est imposée. **Requis** L'authentification des données d'identification est toujours imposée. Si le serveur ne prend pas en charge l'authentification des données d'identification, le client ne peut pas se connecter au serveur. **Prise en charge** (Par défaut) L'authentification des données d'identification est imposée seulement si à la fois le client et le serveur prennent en charge l'authentification des données d'identification.
-authenticationRetryCount (facultatif)	Spécifie le nombre de tentatives d'authentification si les données d'identification sont arrivées à expiration. Si vous ne voulez pas réessayer l'authentification, définissez la valeur à `0`. La valeur par défaut est `0`.
-credentialGeneratorClass	Indique la classe d'implémentation de com.ibm.websphere.objectgrid.security.plugins.builtins. WSTokenCredentialGenerator pour que le client récupère les jetons de sécurité de l'unité d'exécution.
-credentialGeneratorProps	Spécifie les propriétés de la classe d'implémentation CredentialGenerator. Les propriétés sont envoyées à l'objet avec la méthode setProperties(String). La valeur des propriétés du générateur de données d'identification est utilisée seulement si une valeur est spécifiée pour la zone **Classe du générateur de données d'identification**.

Valeur retournée :

Exemples de mode de traitement par lots

- Jacl :

```
$AdminTask modifyXSDomain {-name TestDomain -default true -modifyEndpoints
{{xhost1.ibm.com "" ,2809}} -addEndpoints {{xhost2.ibm.com "" ,2809}}}
-removeEndpoints {{xhost3.ibm.com}}}
```

- Chaîne Jython :

```
AdminTask.modifyXSDomain('[-name TestDomain
-default false -modifyEndpoints [[xhost1.ibm.com "" ,2809]]
-addEndpoints [[xhost3.ibm.com "" ,2809]]
-removeEndpoints [[xhost2.ibm.com]]]')
```

- Utilisation de la sécurité du client avec la commande modify :

```
$AdminTask modifyXSDomain {-name monDomaine -default false
-configureClientSecurity {-securityEnabled true -
Supported -authenticationRetryCount 1 -credentialGeneratorClass
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator
 -credentialGeneratorProps "manager manager1"}}
```

Exemples de mode interactif

- Jacl :

  ```
  $AdminTask modifyXSDomain {-interactive}
  ```

- Chaîne Jython :

  ```
  AdminTask.modifyXSDomain ('[-interactive]')
  ```

testXSDomainConnection

La commande **testXSDomainConnection** teste la connexion à un domaine de services de catalogue.

Paramètres requis :

-name
Spécifie le nom du domaine de services de catalogue vers lequel tester la connexion.

Paramètres facultatifs

-timeout
Spécifie le délai d'attente maximal en secondes de la connexion.

Valeur de retour : si la connexion peut être établie, retourne `true`. Dans le cas contraire, des informations d'erreur de connexion sont retournées.

Exemples de mode de traitement par lots

- Jacl :

  ```
  $Admintask testXSDomainConnection
  ```

- Chaîne Jython :

  ```
  AdminTask.testXSDomainConnection
  ```

Exemples de mode interactif

- Jacl :

  ```
  $AdminTask testXSDomainConnection {-interactive}
  ```

- Chaîne Jython :

  ```
  AdminTask.testXSDomainConnection ('[-interactive]')
  ```

testXSServerConnection

La commande **testXSServerConnection** teste la connexion à un serveur de catalogue. Cette commande fonctionne pour les serveurs autonomes et les serveurs qui font partie d'un domaine de services de catalogue.

Paramètres requis :

hôte
Spécifie l'hôte sur lequel réside le serveur de catalogue.

listenerPort
Spécifie le port d'écoute du serveur de catalogue.

Paramètres facultatifs

timeout
Spécifie en secondes pendant combien de temps au maximum attendre la connexion au serveur de catalogue.

domain
Spécifie le nom du domaine de service de catalogue. Si vous définissez une valeur pour ce paramètre, les propriétés de la sécurité du client pour le domaine de services de catalogue spécifié sont utilisées pour tester la connexion. Sinon, une recherche est effectuée pour trouver le domaine de services de catalogue pour l'hôte et le port d'écoute spécifié. Si un domaine de services de catalogue est trouvé, les propriétés de la sécurité du client qui sont définies pour le domaine de services de catalogue sont utilisées pour tester le serveur. Sinon, aucune propriété de la sécurité du client n'est utilisée lors du test.

Valeur retournée :

Exemples de mode de traitement par lots

- Jacl :
```
$Admintask testXSServerConnection {-host xhost1.ibm.com -listenerPort 2809}
```
- Chaîne Jython :
```
AdminTask.testXSServerConnection('[-host xshost3.ibm.com -listenerPort 2809]')
```

Exemples de mode interactif

- Jacl :
```
$AdminTask testXSServerConnection {-interactive}
```
- Chaîne Jython :
```
AdminTask.testXSServerConnection ('[-interactive]')
```

Collection de domaine de services de catalogue :

Cette page permet de gérer les domaine de services de catalogue. Les domaine de services de catalogue définissent un groupe de serveurs de catalogues qui gèrent le positionnement de fragments et surveille l'état des serveurs de conteneurs dans la grille de données.

Pour afficher cette page de la console d'administration, cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **domaine de services de catalogue**. Pour créer un domaine de services de catalogue, cliquez sur **Nouveau**. Pour supprimer un domaine de services de catalogue, sélectionnez le domaine à supprimer et cliquez sur **Supprimer**.

Tester la connexion :

Lorsque vous cliquez sur le bouton de **test de la connexion**, tous les noeuds finaux du domaine de services de catalogue défini sont interrogés un par un. Si l'un d'entre eux est disponible, il retourne un message qui indique que la connexion au domaine de services de catalogue a abouti. Vous pouvez utiliser ce bouton pour tester la configuration des informations de connexion et de sécurité.

Définir la valeur par défaut :

Indique le domaine de services de catalogue utilisé comme valeur par défaut. Sélectionnez un domaine de services de catalogue comme valeur par défaut et

cliquez sur **Définir la valeur par défaut**. Un seul domaine de serveur de catalogue peut être sélectionné comme valeur par défaut.

Nom :

Indique le nom du domaine de services de catalogue.

Valeur par défaut :

Indique le domaine de services de catalogue de la liste qui est la valeur par défaut. Le domaine de services de catalogue par défaut est indiqué par l'icône :  .

Paramètres du domaine de services de catalogue :

Cette page permet de gérer les paramètres d'un domaine de services de catalogue spécifique. Les domaine de services de catalogue définissent un groupe de serveurs de catalogues qui gèrent le positionnement de fragments et surveille l'état des serveurs de conteneurs dans la grille de données. Vous pouvez définir un domaine de services de catalogue figurant dans la même cellule que le gestionnaire de déploiement. Vous pouvez également définir des domaine de services de catalogue distants si votre configuration WebSphere eXtreme Scale se trouve dans une cellule différente ou que votre grille de données est composée de processus Java SE.

Pour afficher cette page de la console d'administration, cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **Domaines de services de catalogue** > ***catalog_service_domain_name***.

Tester la connexion :

Lorsque vous cliquez sur le bouton de **test de la connexion**, tous les noeuds finaux du domaine de services de catalogue défini sont interrogés un par un. Si l'un d'entre eux est disponible, il retourne un message qui indique que la connexion au domaine de services de catalogue a abouti. Vous pouvez utiliser ce bouton pour tester la configuration des informations de connexion et de sécurité.

Name :

Indique le nom du domaine de services de catalogue.

Activer ce domaine de services de catalogue en tant que valeur par défaut sauf si un autre domaine de service de catalogue est explicitement indiqué :

Si vous cochez cette case, le domaine de services de catalogue sélectionné devient le domaine de services de catalogue par défaut de la cellule. Chaque profil de serveur dans la cellule qui est étendu avec le profil WebSphere eXtreme Scale appartient au domaine du service de catalogue sélectionné.

Pour WebSphere eXtreme Scale, tous les conteneurs eXtreme Scale embarqués dans les modules d'application Java EE se connectent au domaine par défaut. Les clients peuvent se connecter au domaine par défaut en utilisant l'API ServerFactory.getServerProperties().getCatalogServiceBootstrap() pour extraire les noeuds finaux de service de catalogue à utiliser lors de l'appel de l'API ObjectGridManager.connect().

Si vous changez le domaine par défaut pour pointer vers un groupe de serveurs de catalogue différents, tous les conteneurs et clients font référence au nouveau domaine après leur redémarrage.

Serveurs de catalogues :

Indique une liste de serveurs de catalogues appartenant à ce domaine de services de catalogue.

Cliquez sur **Nouveau** pour ajouter un serveur de catalogues à la liste. Ce serveur de catalogues doit déjà être défini dans la configuration eXtreme Scale. Vous pouvez également modifier ou supprimer un serveur dans la liste en sélectionnant le noeud final, puis en cliquant sur **Editer** ou **Supprimer**. Définissez les propriétés suivantes pour chaque noeud final du serveur de catalogues :

Noeud final du serveur de catalogues
Indique le nom du serveur d'applications ou du serveur distant existant sur lequel le service de catalogues s'exécute. Un domaine de services de catalogue ne doit pas contenir une combinaison de serveurs d'applications et de serveurs distants existants.

- **Serveur d'applications existant** : définir le chemin d'un serveur d'applications, d'un agent de noeud ou de gestionnaire de déploiement dans la cellule. Un service de catalogue démarre automatiquement dans le serveur sélectionné. Sélectionnez un serveur dans la liste des serveurs d'applications existants. Tous les serveurs d'applications que vous définissez dans le domaine de services de catalogue doivent se trouver dans le même groupe central.
- **Serveur distant** : définit le nom d'hôte du serveur de catalogue distant.

  Pour les noeuds finaux distants WebSphere eXtreme Scale : définit le nom d'hôte du processus serveur de catalogue distant. Vous devez démarrer les serveurs distants avec le script **`startOgServer`** ou l'API de serveur embarqué API.

Port client
Spécifie le port utilisé pour la communication entre les serveurs de catalogues dans le domaine de services de catalogue. Cette valeur est nécessaire pour les serveurs de catalogue qui s'exécutent dans des processus WebSphere Application Server. Vous pouvez définir n'importe quel port comme valeur s'il n'est pas utilisé par un autre processus.

Port d'écoute
Indique le port utilisé pour établir des communications avec les clients. Cette valeur est obligatoire pour les noeuds finaux distants et elle doit correspondre à la valeur utilisée au démarrage du service de catalogue. Le port d'écoute est utilisé par les clients et les conteneurs pour communiquer avec le service de catalogue.

Pour les noeuds finaux distants WebSphere eXtreme Scale : définit le port d'écoute ORB (Object Request Broker) qui permet aux conteneurs et aux clients de communiquer avec le service de catalogue via l'ORB. Pour les noeuds finaux WebSphere Application Server, la valeur de port d'écoute est héritée de la configuration du port BOOTSTRAP_ADDRESS.

Statut

Tableau 22. Etat de noeud final de serveur de catalogues

Icône	Définition
	Inconnu
	Démarré
	Arrêté

Propriétés de sécurité du client :

Utilisez cette page pour définir la sécurité client d'un domaine de services de catalogue. Ces paramètres s'appliquent à tous les serveurs dans votre domaine de services de catalogue. Ces propriétés peuvent être remplacées en définissant un fichier `splicer.properties` avec la propriété personnalisée com.ibm.websphere.xs.sessionFilterProps ou en raccordant le fichier EAR d'application.

Pour afficher cette page de la console d'administration, cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **domaine de services de catalogue** > ***catalog_service_domain_name*** > **Propriétés de sécurité du client**.

Activer la sécurité du client :

Indique que la sécurité du client est activée pour le serveur de catalogue. Le fichier des propriétés du serveur qui est associé au serveur de catalogue sélectionné doit avoir un paramètre **`securityEnabled`** correspondant dans le fichier des propriétés du serveur. Si ces paramètres ne correspondent pas, une exception est générée.

Authentification des données d'identification :

Indique si l'authentification des données d'identification est appliquée ou prise en charge.

Jamais
: Aucune authentification n'est appliquée.

Requis
: L'authentification des données d'identification est toujours appliquée. Si le serveur ne prend pas en charge l'authentification des données d'identification, le client ne peut pas se connecter au serveur.

Prise en charge
: L'authentification des données d'identification est appliquée uniquement si le client et le serveur la prennent en charge.

Nombre de tentatives d'authentification :

Indique le nom de nouvelles tentatives d'authentification si les données d'identification ont expiré.

Si vous ne voulez pas tente d'authentifier de nouveau les données, définissez la valeur `0`.

Classe de générateur de données d'identification :

Indique la classe d'implémentation com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator pour que le client puisse extraire les données d'identification depuis l'objet CredentialGenerator.

Vous pouvez choisir depuis deux classes de générateurs d'identification de données prédéfinis ou vous pouvez définir un générateur personnalisé. Si vous choisissez un générateur personnalisé, vous devez indiquer le nom de la classe du générateur de données d'identification.

- com.ibm.websphere.objectgrid.security.plugins. UserPasswordCredentialGenerator
- com.ibm.websphere.objectgrid.security.plugins. UserPasswordCredentialGenerator
- Générateur de données d'identification personnalisé

Type de sujet :

Indique si vous utilisez l'appelant J2EE ou le type de sujet J2EE runAs. Vous devez définir cette valeur lorsque vous choisissez le générateur de données d'identification WSTokenCredentialGenerator.

- **runAs** : le sujet contient le principal de l'exécution J2EE comme identité et l'exécution de J2EE en tant que données d'identification.
- **caller** : le sujet contient le principal de l'appelant J2EE et ses données d'identification.

ID utilisateur :

Définissez un ID utilisateur lorsque vous utilisez l'implémentation de générateur de donnée d'identification UserPasswordCredentialGenerator.

Mot de passe :

Définissez un mot de passe lorsque vous utilisez l'implémentation de générateur de donnée d'identification UserPasswordCredentialGenerator.

Propriétés du générateur de données d'identification :

Définissez les propriétés d'une classe d'implémentation CredentialGenerator personnalisée. Les propriétés sont définies dans l'objet avec la méthode setProperties(String). La valeur des propriétés du générateur de données d'identification est utilisée seulement si une valeur est spécifiée pour la zone **Classe du générateur de données d'identification**.

Propriétés personnalisées du domaine de services de catalogue :

Vous pouvez continuer à modifier la configuration du domaine de services de catalogue en définissant les propriétés personnalisées.

Pour afficher cette page de la console d'administration, cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **domaine de services de catalogue** > **Propriétés personnalisées**. Pour créer une propriété personnalisée, cliquez sur **Nouveau**.

Nom :

Indique le nom de la propriété personnalisée du domaine de services de catalogue.

Valeur :

Indique une valeur pour la propriété personnalisée du domaine de services de catalogue.

Configuration des serveurs de conteneurs dans WebSphere Application Server

Configurez les serveurs de conteneur dans WebSphere Application Server avec un fichier de propriétés de serveur et un fichier XML de stratégie de déploiement intégré au module d'application Java EE. Les serveurs de conteneur s'arrêtent et démarrent lorsque l'application est arrêtée et démarrée.

Avant de commencer

Configurez un domaine de services de catalogue. Pour plus d'informations, voir «Création de domaines de services de catalogue dans WebSphere Application Server», à la page 258.

Pourquoi et quand exécuter cette tâche

Pour créer des serveurs de conteneur dans WebSphere Application Server, vous devez imbriquer les fichiers XML de configuration WebSphere eXtreme Scale pour créer les serveurs de conteneur dans le module d'application.

Procédure

1. Identifiez les serveurs d'applications sur lequel vous souhaitez déployer l'application Java EE qui contient les définitions de serveur de conteneur WebSphere eXtreme Scale. Vérifiez que les profils de serveur d'applications cible ont été étendus avec le profil WebSphere eXtreme Scale. Dans un environnement de production, ne faites pas cohabiter les serveurs que vous utilisez pour les serveurs de conteneur et les serveurs de catalogue. Pour plus d'informations, voir «Création et augmentation de profils pour WebSphere eXtreme Scale», à la page 184.
2. Configurez un fichier de propriétés du serveur et ajoutez-le au chemin d'accès aux classes de chaque noeud de serveur d'applications cible. Pour plus d'informations, voir Fichier de propriétés du serveur.
3. Ajoutez le fichier XML du descripteur d'ObjectGrid et le fichier XML de stratégie de déploiement au module d'application. Pour plus d'informations, voir «Configuration des applications WebSphere Application Server pour démarrer automatiquement les serveurs de conteneur».

Configuration des applications WebSphere Application Server pour démarrer automatiquement les serveurs de conteneur :

Les serveurs de conteneur dans un environnement WebSphere Application Server démarrent automatiquement quand un module contenant les fichiers XML eXtreme Scale démarre.

Avant de commencer

WebSphere Application Server et WebSphere eXtreme Scale doivent être installés, et vous devez être capable d'accéder à la console d'administration de WebSphere Application Server.

Pourquoi et quand exécuter cette tâche

Les applications Java Platform, Enterprise Edition ont des règles de chargeur de classe complexes qui compliquent grandement le chargement des classes lors de l'utilisation d'une grille de données partagée dans un serveur Java EE. Une application Java EE correspond généralement à un seul fichier EAR (Enterprise Archive). Le fichier EAR contient un ou plusieurs modules EJB (Enterprise JavaBeans) ou modules WAR (Web archive) déployés.

WebSphere eXtreme Scale surveille le démarrage de chaque module et recherche des fichiers XML eXtreme Scale. Si le service de catalogue détecte qu'un module démarre avec les fichiers XML, le serveur d'applications est enregistré en tant que conteneur de serveur machine virtuelle Java (JVM). En enregistrant les serveurs de conteneur dans le service de catalogue, une même application peut être déployée dans des grilles de données différents, mais utilisée comme une grille de données unique par le service de catalogue. Le service de catalogue n'est pas concerné par les cellules, les grilles, ou les grilles dynamiques. Une grille de données unique peut couvrir plusieurs cellules, si nécessaire.

Procédure

1. Modularisez le fichier EAR pour disposer de modules incluant les fichiers XML eXtreme Scale dans le dossier `META-INF`. WebSphere eXtreme Scale détecte la présence des fichiers `objectGrid.xml` et `objectGridDeployment.xml` dans le dossier `META-INF` des modules EJB et WEB lorsqu'ils démarrent. Si un seul fichier `objectGrid.xml` est détecté, la machine JVM est supposée être un client. Sinon, la machine virtuelle Java est supposée faire office de grille de données définie dans le fichier `objectGridDeployment.xml`.

   Vous devez utiliser les noms corrects pour ces fichiers XML. Les noms de fichier sont sensibles à la casse. Si les fichiers sont absents, le conteneur ne démarre pas. Vous pouvez vérifier si le fichier `systemout.log` contient des messages indiquant que des fragments sont placés. Un module EJB ou d'archive Web utilisant eXtreme Scale doit avoir des fichiers XML eXtreme Scale dans son répertoire `META-INF`.

   Les fichiers XML eXtreme Scale incluent :

   - Un fichier XML de descripteur d'ObjectGrid nommé `objectGrid.xml`. Pour plus d'informations, voir Fichier XML du descripteur d'ObjectGrid.
   - Un fichier XML de descripteur de déploiement nommé `objectGridDeployment.xml`. Pour plus d'informations, voir Fichier XML du descripteur de la règle de déploiement.
   - (Facultatif) Un fichier XML de descripteur de métadonnées d'entité, si des entités sont utilisées. Le nom du fichier `entity.xml` doit correspondre au nom spécifié dans le fichier `objectGrid.xml`. Pour plus d'informations, voir Fichier XML du descripteur de métadonnées d'entité.

   L'environnement d'exécution détecte ces fichiers, puis contacte le service de catalogue pour l'informer qu'un autre conteneur est disponible pour héberger les fragments pour ce eXtreme Scale.

   Conseil : Si votre application comporte des entités et que vous prévoyez d'utiliser un serveur un conteneur, affectez la valeur **`minSyncReplicas`**) `0` dans le fichier XML du descripteur de déploiement. Sinon, vous risquez de voir l'un des messages suivants dans le fichier `SystemOut.log` car le positionnement ne pourra se produire tant qu'un autre serveur n'a pas démarré pour satisfaire à la règle minSyncReplica :

```
CWPRJ1005E: Erreur lors de la résolution de l'association d'entités.
Entité=nom_entité,
association=nom_association.

CWOBJ3013E: Le référentiel EntityMetadata n'est pas disponible.
Le seuil du délai d'attente a été atteint lors de la
tentative d'inscription de l'entité : nom_entité.
```

2. Déployez et démarrez votre application.

   Le conteneur démarre automatiquement quand le module est démarré. Le service de catalogue commence à placer les serveurs principaux et secondaires de partition (fragments) dès que possible. Ce placement a lieu immédiatement, à moins que vous ne définissiez l'environnement pour le retarder. Pour plus d'informations, voir «Contrôle du placement», à la page 427.

Que faire ensuite

Les applications dans la même cellule que les conteneurs, peuvent se connecter à ces grilles de données à l'aide d'une méthode ObjectGridManager.connect(null, null), puis appeler la méthode getObjectGrid(ccc, "object grid name"). La méthode connect ou getObjectGrid peut être bloquée jusqu'à ce que les conteneurs aient placés les fragments, mais ce blocage représente un problème uniquement quand la grille de données démarre.

Chargeurs de classe

Tout plug-in ou objet stocké dans un eXtreme Scale est chargé sur un certain chargeur de classe. Deux modules EJB dans un même fichier d'archive d'entreprise peuvent inclure ces objets. Ces objets sont les identiques, mais ils sont chargés avec différents chargeurs de classe. Si l'application A stocke un objet Personne dans une mappe qui est locale pour le serveur, l'application B reçoit une exception `ClassCastException` si elle essaie de lire cet objet. Cette exception se produit car l'application B a chargé l'objet Personne sur un chargeur de classe différent.

Une manière de résoudre ce problème consiste à faire en sorte qu'un module racine contienne les plug-in et les objets nécessaires qui sont stockés dans le eXtreme Scale. Chaque module utilisant eXtreme Scale doit référencer ce module pour ses classes. Une autre solution consiste à placer ces objets partagés dans un fichier JAR d'utilitaire qui se trouve dans un chargeur de classe commun partagé par les modules et les applications. Les objets peuvent également être placés dans des classes WebSphere ou le répertoire `lib/ext`, mais cet placement complique le déploiement.

Les modules EJB dans un fichier d'archive d'entreprise partagent généralement le même ClassLoader et ne sont pas affectés par ce problème. Chaque module de fichier d'archive Web possède son propre ClassLoader et est affecté par ce problème.

Connexion à un client de grille de données uniquement

Si la propriété **`catalog.services.cluster`** est définie dans les propriétés personnalisées d'une cellule, d'un noeud ou d'un serveur, un module dans le fichier EAR peut appeler la méthode ObjectGridManager.connect (ServerFactory.getServerProperties().getCatalogServiceBootstrap(), null, null) pour obtenir un ClientClusterContext. Le module peut également appeler la méthode ObjectGridManager.getObjectGrid(ccc, "grid name") pour obtenir une référence à la grille de données. Si des objets d'application sont stockés dans des mappes, vérifiez que ces objets sont présents dans un chargeur de classe commun.

Les clients Java ou les clients en dehors de la cellule peuvent se connecter au port IIOP d'amorçage du service de catalogue. Dans WebSphere Application Server, le gestionnaire de déploiement héberge le service de catalogue par défaut. Le client peut alors obtenir un ClientClusterContext et la grille de données nommée.

Gestionnaire d'entités

Avec le gestionnaire d'entités, les blocs de données sont stockés dans les mappes et non pas les objets d'application, ce qui réduit les problèmes de chargeur de classe. Les plug-in, en revanche, peuvent présenter un problème. Notez également qu'un fichier XML de descripteur ObjectGrid de remplacement client est toujours nécessaire lorsqu'une grille de données a des entités définies : ObjectGridManager.connect("host:port[,host:port], null, objectGridOverride) or ObjectGridManager.connect(null, objectGridOverride).

Configuration de IBM eXtremeMemory et de IBM eXtremeIO

En configurant eXtremeMemory, vous pouvez stocker des objets dans la mémoire native plutôt que dans le segment de mémoire Java. La configuration eXtremeMemory active eXtremeIO, un nouveau mécanisme de transport. En retirant des objets du segment de mémoire Java, vous pouvez éviter les pauses de récupération d'espace, ce qui permet de bénéficier de performances plus constantes et de temps de réponse plus prévisibles.

Avant de commencer

- Linux eXtremeIO et eXtremeMemory sont pris en charge sur les systèmes Linux x86 64 bits qui utilisent un kit SDK 64 bits uniquement.
- Vous devez utiliser des ensembles de mappes qui ont toutes les mappes configurées avec le mode de copie `COPY_TO_BYTES` ou `COPY_TO_BYTES_RAW`. Si aucune mappe du groupe de mappes n'est utilisé par l'un de ces modes de copie, les objets sont stockés dans le segment de mémoire Java et l'ORB (Object Request Broker) est utilisé.
- Vous ne pouvez pas utiliser eXtremeIO et eXtremeMemory dans les scénarios de configuration suivants :
  - Lorsque vous utilisez des serveurs de conteneur qui s'exécutent dans un environnement WebSphere Application Server.
  - Lorsque vous utilisez des plug-ins d'expulseur personnalisés.
  - Lorsque vous utilisez des index composites.
  - Lorsque vous utilisez des chargeurs intégrés en écriture différée.
  - Lorsque vous utilisez l'interface ReplicationMapListener pour créer une implémentation d'un programme d'écoute d'événement pour les mappes côté client qui sont en mode de réplication.

Pourquoi et quand exécuter cette tâche

La machine JVM repose sur l'heuristique d'utilisation pour collecter, réduire et augmenter la mémoire des processus. Le récupérateur de place exécute ces opérations. Toutefois, l'exécution de la récupération de place a un coût qui augmente proportionnellement avec la taille du segment de mémoire Java et le nombre d'objets dans la grille de données. La machine JVM fournit différentes heuristiques pour différents cas d'utilisation et objectifs : récupération de place avec traitement optimal, temps de pause optimal, générationnelle, équilibrée et temps réel. Aucune heuristique n'est parfaite. Une seule heuristique ne peut pas convenir à toutes les configurations possibles.

WebSphere eXtreme Scale utilise la mise en cache des données avec des mappes réparties qui ont des entrées avec un cycle de vie bien connu. Ce cycle de vie inclut les opérations suivantes : GET, INSERT, DELETE et UPDATE. En utilisant ces cycles de vie de mappe bien connus, eXtremeMemory et eXtremeIO peuvent utiliser la mémoire plus efficacement que les heuristiques d'utilisation JVM.

Le diagramme suivant montre comment l'utilisation d'eXtremeMemory améliore la cohérence des temps de réponse dans l'environnement. Lorsque le temps de réponse relatif atteint les percentiles les plus élevés, les demandes qui utilisent eXtremeMemory ont des temps de réponse relativement plus lents. Le diagramme montre les percentiles 95-100.

.

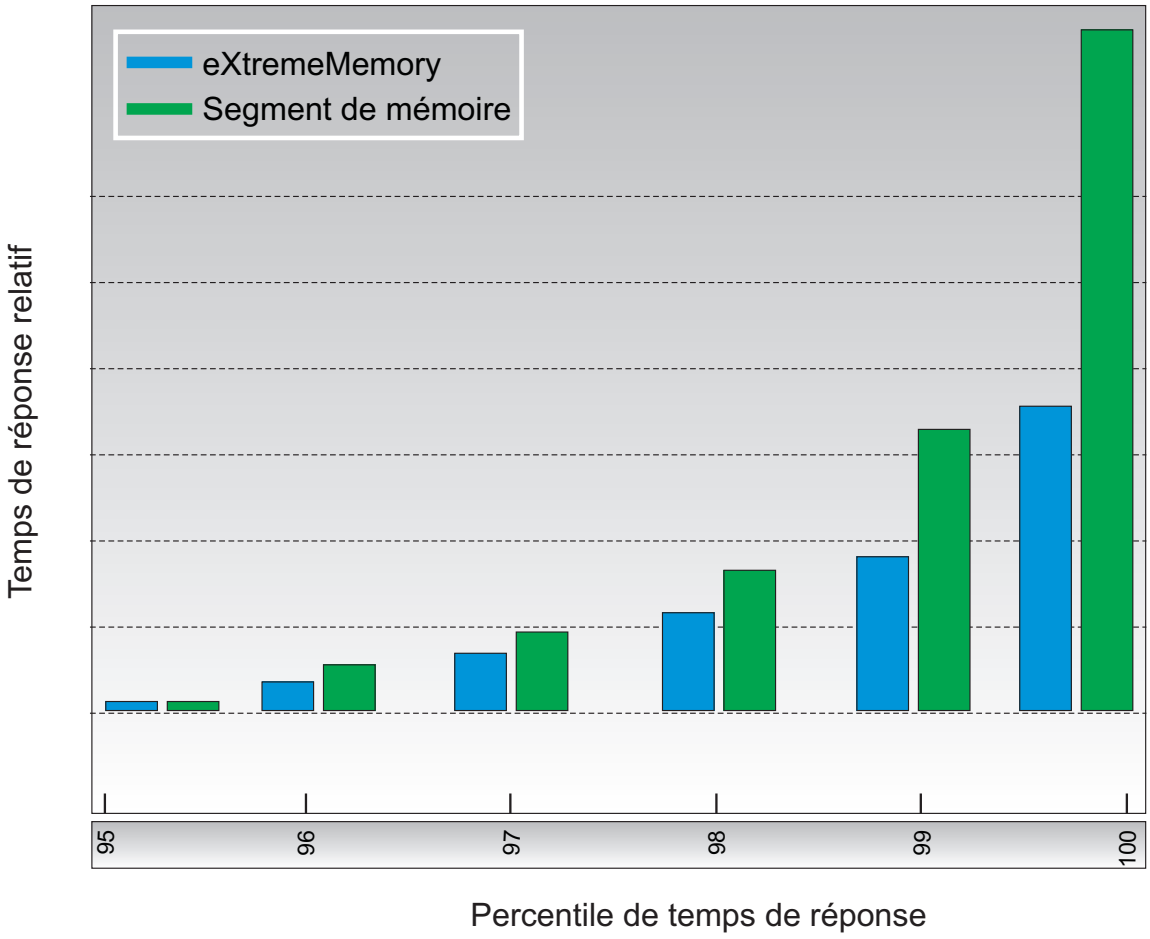


Figure 30. Comparaison des temps de réponse eXtremeMemory et de segment de mémoire

Lorsque vous utilisez eXtremeMemory, eXtremeIO est utilisé pour la communication entre les serveurs de conteneur. Les objets sont sérialisés en octets dans le serveur de conteneur. Pour activer eXtremeIO et eXtremeMemory, vous devez définir les propriétés de serveur requises sur tous les serveurs de conteneur dans les grilles de données et redémarrer les serveurs.

Procédure

1. Facultatif : Déterminez la valeur de la propriété **maxXMSize** à utiliser.
   a. Dans votre configuration existante, déterminez la taille par entrée. Exécutez la commande **xscmd -c showMapSizes** pour déterminer cette taille.
   b. Calculez la valeur **maxXMSize**. Pour obtenir la taille totale maximale des entrées (*maximum_total_size*), multipliez *size_per_entry* * *maximum_number_of_entries*. N'utilisez pas plus de 60 % de **maxXMSize** pour tenir compte du traitement des métadonnées. Multipliez *maximum_total_size** 1.65 pour la valeur **maxXMSize**.
2. Mettez à jour le fichier de propriétés de chaque serveur de conteneur dans la configuration afin d'activer le nouveau transport. Les propriétés de serveur suivantes actives le nouveau transport :

   Propriétés requises

   7.1.1+ **enableXM**
   Lorsque la valeur est `true`, active IBM eXtremeMemory sur le serveur et configure le serveur pour utiliser IBM eXtremeIO pour la réplication synchrone et asynchrone. Les entrées de cache sont stockées dans la mémoire native et non pas dans le segment de mémoire Java. tous les serveurs de conteneurs dans les grilles de données doivent utiliser la même valeur pour la propriété **enableXM**.

   Valeur par défaut : `false`

   Propriétés suggérées

   7.1.1+ **maxXMSize**
   Définit la quantité maximale de mémoire, en mégaoctets, utilisée par le serveur pour le stockage eXtremeMemory.

   Valeur par défaut : 25 % de la mémoire totale du système

   Propriétés facultatives

   7.1.1+ **maxXIONetworkThreads**
   Définit le nombre maximum d'unités d'exécution à allouer dans le pool d'unités d'exécution du réseau de transport eXtremeIO.

   Valeur par défaut :`50`

   7.1.1+ **minXIONetworkThreads**
   Définit le nombre minimum d'unités d'exécution à allouer dans le pool d'unités d'exécution du réseau de transport eXtremeIO.

   Valeur par défaut :`50`

   7.1.1+ **maxXIOWorkerThreads**
   Définit le nombre maximum d'unités d'exécution à allouer dans le pool d'unités d'exécution de traitement des demandes de transport.

   Valeur par défaut :`128`

   7.1.1+ **minXIOWorkerThreads**
   Définit le nombre minimum d'unités d'exécution à allouer dans le pool d'unités d'exécution de traitement des demandes de transport.

   Valeur par défaut :`128`

   7.1.1+ **xioChannel.xioContainerTCPNonSecure.Port**
   Indique le numéro de port d'écoute non sécurisé de eXtremeIO sur le

serveur. Si vous ne définissez pas de valeur, un port éphémère est utilisé. Cette propriété est utilisée uniquement lorsque la propriété **transportType** a la valeur `TCP/IP`.

7.1.1+ **xioChannel.xioContainerTCPSecure.Port**
Indique le numéro de port SSL de eXtremeIO sur le serveur. Cette propriété est utilisée uniquement lorsque la propriété **transportType** a la valeur `SSL-Supported` ou `SSL-Required`.

3. Redémarrez les serveurs de conteneur pour utiliser le nouveau mécanisme de transport. Pour plus d'informations, voir «Démarrage et arrêt des serveurs sécurisés», à la page 395 et «Démarrage et arrêt des serveurs dans un environnement WebSphere Application Server», à la page 409.

Configuration de plusieurs topologies de centres de données

Avec la réplication asynchrone multimaître, vous liez un ensemble de domaines de services de catalogue. Les domaines de services de catalogue connectés sont ensuite synchronisés en utilisant la réplication via les liaisons. Vous pouvez définir les liaisons à l'aide de fichiers de propriétés, lors de l'exécution avec des programmes JMX (Java Management Extensions) ou avec les utilitaires de ligne de commande. Le groupe de liaisons actuel d'un domaine est stocké dans le service de catalogue. Vous pouvez ajouter et supprimer des liens sans redémarrer le domaine de services de catalogue qui héberge la grille de données.

Avant de commencer

- Voir «Planification de plusieurs topologies de centre de données», à la page 36 pour plus d'informations sur les topologies de réplication multimaîtres et les considérations de conception. Vous pouvez configurer des liaisons entre les domaines de services de catalogue avec le fichier de propriétés du serveur pour former la topologie lors du démarrage du serveur. Vous pouvez configurer des liaisons lors de l'exécution.
- Si vous utilisez des chargeurs dans la topologie de réplication multimaître, vous devez planifier la manière dont vous allez gérer des données exactes entre les centres de données. Les méthodes que vous pouvez utiliser varient en fonction de la topologie que vous utilisez. Pour plus d'informations, voir «Remarques sur les chargeurs dans une topologie multimaître», à la page 41.

Procédure

- Définissez des liaisons dans le fichier de propriétés du serveur de catalogue de chaque domaine de services de catalogue de la topologie à des fin d'amorçage.

  Voir Fichier de propriétés du serveur pour plus d'informations sur la définition de ce fichier pour le serveur de catalogue.

  Important : Les noms de propriété sont sensibles à la casse.

  Nom de domaine local :
  Indiquez le nom du domaine de services de catalogue du serveur de catalogue actuel, le domaine A, par exemple.

  ```
  domainName=A
  ```

  Liste facultative des noms de domaines externes :

  Indiquez les noms des domaines de services de catalogue auxquels vous souhaitez vous lier dans la topologie de réplication multimaître ; le domaine B, par exemple :

  ```
  foreignDomains=B
  ```

Liste facultative des noms de domaines externes :

Spécifie les informations de connexion des serveurs de catalogue des domaines externes ; le domaine B, par exemple :

```
B.endPoints=hostB1:2809, hostB2:2809
```

Si un domaine externe comporte plusieurs serveurs de catalogue, spécifiez-les tous.

- Utilisez l'utilitaire **xscmd** ou la programmation JMX pour ajouter ou supprimer des liaison lors de l'exécution.

  Les liens d'un domaine sont conservés dans le service de catalogue dans la mémoire répliquée. Cet ensemble de liens peut être modifié à tout moment par l'administrateur sans nécessiter pour autant un redémarrage de ce domaine ou des autres domaines. L'utilitaire **xscmd** inclut plusieurs options pour l'utilisation des liaisons.

  L'utilitaire **xscmd** se connecte à un service de catalogue et donc un domaine de services de catalogue unique. Par conséquent, l'utilitaire **xscmd** peut être utilisé pour créer et supprimer des liaisons entre le domaine auquel il se connecte et n'importe quel autre domaine.

  Utilisez la ligne de commande pour créer une liaison, par exemple :

  ```
  xscmd –c establishLink -cep host:2809 -fd dname -fe fdHostA:2809,fdHostB:2809
  ```

  Cette commande établit une nouvelle liaison entre le domaine local et le domaine externe nommé `dname`. Le service de catalogue `dname` est exécuté à l'adresse `fdHostA:2809` et à l'adresse `fdHostB:2809`. Le domaine de services de catalogue local a un hôte d'écoute de service de catalogue et le port `host:2809`. Indiquez tous les noeuds finaux du service de catalogue à partir du domaine externe de sorte que la connectivité à la tolérance aux pannes pour le domaine soit possible. N'utilisez pas une seule paire `host:port` pour le service de catalogue du domaine de services de catalogue externe.

  Vous pouvez utiliser n'importe quelle machine virtuelle Java avec **xscmd** et utilisez l'option **-cep**. Si le serveur de catalogue est hébergé sur un gestionnaire de déploiement WebSphere Application Server, le port est habituellement le 9809.

  Les ports spécifiés pour le domaine externe ne sont pas des ports JMX. Ce sont les ports que l'on utilise d'ordinaire pour les clients eXtreme Scale.

  Une fois que la commande d'ajout de nouveau lien a été émise, le service de catalogue donne instruction à tous les conteneurs qu'il gère de commencer à se répliquer vers le domaine externe. Un lien n'est pas nécessaire des deux côtés. Il suffit d'en créer un sur l'une des deux extrémités.

  La ligne de commande permet également de supprimer un lien, par exemple :

  ```
  xscmd –c dismissLink -cep host:2809 -fd dname
  ```

  Cette commande se connecte au service de catalogue d'un domaine et lui donne instruction d'arrêter la réplication vers un domaine spécifique. Une liaison doit être supprimée sur un côté uniquement.

Liaison entre deux domaines de services de catalogue

Supposons que vous souhaitiez définir une configuration à deux domaines comportant les domaines de services de catalogue A et B.

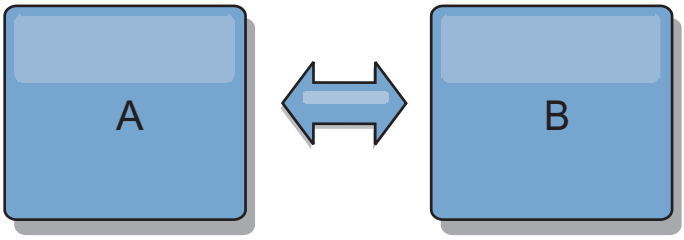


Figure 31. Liaison entre les domaines de services de catalogue

Voici le fichier de propriétés du serveur de catalogue dans le domaine A :

```
domainName=A
foreignDomains=B
B.endPoints=hostB1:2809, hostB2:2809
```

Voici le fichier de propriétés du serveur de catalogue dans le domaine B. Notez les similitudes entres les deux fichiers de propriétés.

```
domainName=B
foreignDomains=A
A.endPoints=hostA1:2809,hostA2:2809
```

Une fois les deux domaines démarrés, toutes les grilles de données ayant les caractéristiques suivantes sont répliquées entre ces domaines.

- Dispose d'un service de catalogue privé avec un nom de domaine unique
- A le même nom de grille de données que les autres grilles du domaine
- A le même nombre de partitions que les autres grilles de données dans le domaine
- Est une grille de données FIXED_PARTITION (les grilles de données PER_CONTAINER ne peuvent pas être répliquées)
- A le même nombre de partitions (sans forcément pour autant avoir le même nombre et le même type de fragments réplique)
- A les mêmes types de données répliqués que les autres grilles du domaine
- A les mêmes nom de groupe de mappes, noms de mappe et modèles de mappes dynamiques que les autres grilles dans le domaine

La règle de réplication d'un domaine de service de catalogue est ignorée.

L'exemple qui précède montre comment configurer chaque domaine pour qu'il ait un lien vers l'autre domaine, mais, en fait, il suffit de définir un lien dans une seule direction. C'est particulièrement utile lorsqu'on a affaire à des topologies en étoile, la configuration s'en trouve considérablement simplifiée. Le fichier de propriétés du concentrateur ne nécessite pas d'être modifié au fur et à mesure que des noeuds sont ajoutés à la topologie et il suffit que le fichier de chacun de ces noeuds comprenne des informations relatives au concentrateur. De la même manière, dans une topologie en anneau, il suffit que chacun des domaines ait un lien avec le domaine qui le précède et avec celui qui le suit dans l'anneau.

Exemple : topologie en étoile

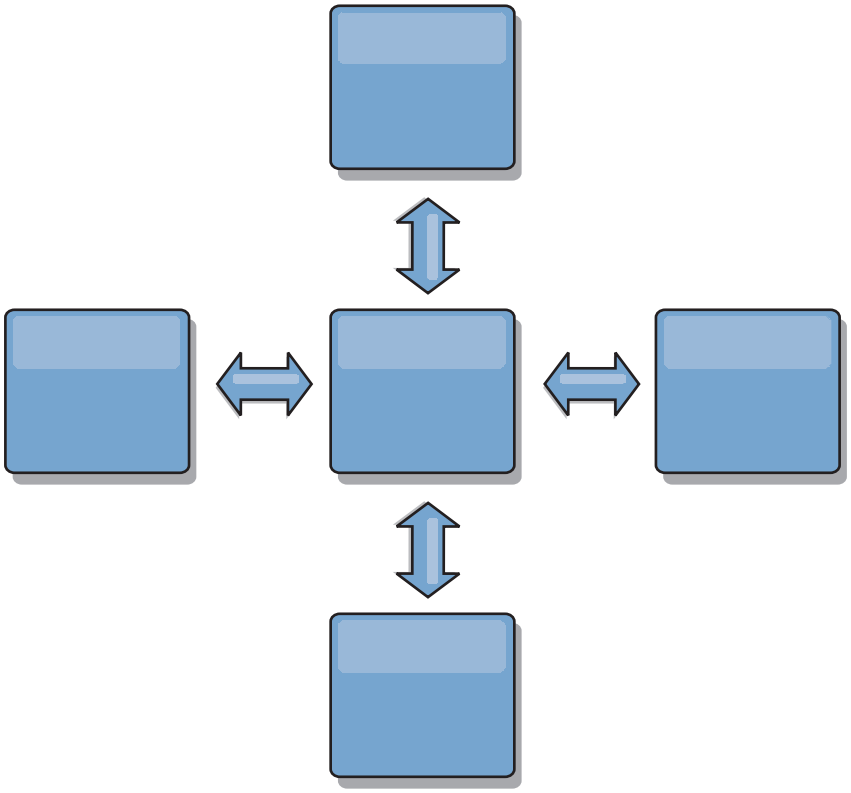


Figure 32. Topologie en étoile

La topologie avec un concentrateur et quatre domaines A, B, C et D possède des fichiers de propriétés de serveur, comme dans les exemples suivants.

```
domainName=Hub
```

La Branche A a les propriétés de serveur suivantes :

```
domainName=A
foreignDomains=Hub
Hub.endPoints=hostH1:2809, hostH2:2809
```

La branche B a les propriétés de serveur suivantes :

```
domainName=B
foreignDomains=Hub
Hub.endPoints=hostH1:2809, hostH2:2809
```

La branche C a les propriétés de serveur suivantes :

```
domainName=C
foreignDomains=Hub
Hub.endPoints=hostH1:2809, hostH2:2809
```

La branche D a les propriétés de serveur suivantes :

```
domainName=D
foreignDomains=Hub
Hub.endPoints=hostH1:2809, hostH2:2809
```

Que faire ensuite

Vous pouvez fournir un arbitre de collisions personnalisé pour résoudre les conflits entre les domaine de services de catalogue. Pour plus d'informations, voir Développement d'arbitres personnalisés pour la réplication multi-maître.

Configuration des ports

WebSphere eXtreme Scale est un cache réparti qui nécessite l'ouverture de ports pour communiquer avec l'ORB (Object Request Broker) et la pile TCP (Transmission Control Protocol) parmi les machine virtuelle Java (JVM) et les autres serveurs.

Configuration de ports en mode autonome

Vous pouvez configurer les ports nécessaires pour les serveurs et les clients dans un déploiement eXtreme Scale à l'aide de paramètres de ligne de commande, de fichiers de propriétés ou à l'aide d'un programme. La plupart des exemples dans les sections suivantes décrivent les paramètres de ligne de commande dans le script **startOgServer**. Des options de configuration équivalentes peuvent être également définies dans des fichiers de propriétés, à l'aide de l'API de serveur embarqué ou l'API client.

Procédure

1. Démarrez les noeuds finaux de service de catalogue.

   WebSphere eXtreme Scale utilise IIOP pour la communication entre les machines virtuelles Java. Les machines virtuelles Java de service de catalogue sont les seuls processus qui requièrent la configuration explicite de ports pour les services IIOP et de ports des services de groupe. Les autres processus d'allouer dynamiquement les ports.

   Le port du client et le port homologue sont utilisés pour la communication entre les services de catalogue dans un domaine de services de catalogue. Pour indiquer le port client et le port homologue, utilisez l'option de ligne de commande suivante :

   -catalogServiceEndPoints <serverName:hostName:clientPort:peerPort>
   Dans le conteneur, fait référence à l'hôte et au port ORB (Object Request Broker) dans le service de catalogue. Chaque attribut est défini comme suit :

   serverName
   Spécifie un nom permettant d'identifier le processus que vous lancez.

   hostName
   Spécifie le nom d'hôte de l'ordinateur sur lequel le serveur est lancé.

   clientPort
   Spécifie le port utilisé pour la communication de service de catalogue homologue.

   peerPort
   Cette value est identique à haManagerPort. Spécifie le port utilisé pour la communication de service de catalogue homologue.

   L'exemple suivant démarre le serveur de catalogue, cs1, qui se trouve dans le même domaine de services de catalogue que les serveurs cs2 et cs3 :

   ```
   startOgServer.bat|sh cs1 -catalogServiceEndPoints
   cs1:MyServer1.company.com:6601:6602,
   cs2:MyServer2.company.com:6601:6602,
   cs3:MyServer3.company.com:6601:6602
   ```

   Les noeud finaux de service de catalogue peuvent être également définis à l'aide de la propriété de serveur de catalogue catalogClusterEndPoints. Le port

d'écoute ORB (Object Request Broker) est utilisé pour la communication entre les services de catalogue dans un domaine de services de catalogue et pour la communication entre les services de catalogue et les serveurs de conteneur et les clients. Pour indiquer le port d'écoute et l'hôte d'écoute, utilisez l'option de ligne de commande suivante :

-listenerHost <nom d'hôte>

Indique le nom d'hôte auquel l'ORB (Object Request Broker) se connecte pour communiquer avec IIOP (Internet Inter-ORB Protocol). La valeur doit être un nom qualifié complet de domaine ou une adresse IP. Si la configuration implique plusieurs cartes réseau, configurez l'hôte du programme d'écoute et le port d'écoute pour que l'ORB (Object Request Broker) dans la machine JVM connaisse l'adresse IP à laquelle se connecter. Si vous ne définissez pas l'adresse IP à utiliser, des symptômes (délais de connexion, défaillances inhabituelles d'API et clients qui semblent se bloquer) apparaissent.**Valeur par défaut :** `localhost`

-listenerPort <port>

Indique le numéro de port auquel se connecte l'ORB (Object Request Broker). Ce paramètre configure les conteneurs et les clients pour communiquer avec le service de catalogue via l'ORB. Dans WebSphere Application Server, le port d'écoute est hérité par la configuration de port BOOTSTRAP_ADDRESS. Cette propriété s'applique au serveur de conteneur et au service de catalogue.**Valeur par défaut :** `2809`

Le port d'écoute et l'hôte d'écoute peuvent être également définis à l'aide des propriétés listenerHost et listenerPort.

Le port de service JMX est utilisé pour la communication entre les clients JMX. Pour indiquer le port de service JMX, utilisez l'option de ligne de commande suivante :

-JMXServicePort <port>

Spécifie le numéro du port sur lequel le serveur MBean écoute les communications avec Java Management Extensions (JMX). Vous devez utiliser un numéro de port différent pour chaque machine virtuelle Java dans votre configuration. Si vous voulez utiliser JMX/RMI, définissez explicitement l'option **-JMXServicePort** et le numéro de port, même si vous souhaitez utiliser la valeur de port par défaut. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue.

Valeur par défaut : `1099`

Le port de service JMX peut être également défini à l'aide de la propriété de serveur JMXServicePort. Lorsque la sécurité est activée, un port SSL (Secure Socket Layer) est aussi nécessaire. Pour indiquer le port SSL, utilisez l'option de ligne de commande suivante :

```
-jvmArgs -Dcom.ibm.CSI.SSLPort=<sslPort>
```

```
./startOgServer.sh cs1 -listenerHost hostA -listenerPort 2809
-catalogServiceEndPoints cs1:hostA:6601:6611,cs2:hostB:6601:6611
```

Figure 33. Exemple d'utilisation de ligne de commande. Démarrez le premier serveur de catalogues sur hostA. Exemple de commande :

Démarrez le second serveur de catalogues sur hostB. Exemple de commande :

```
./startOgServer.sh cs2 -listenerHost hostB -listenerPort 2809
-catalogServiceEndPoints cs1:hostA:6601:6611,cs2:hostB:6601:6611
```

2. Démarrez les noeuds finaux de serveur de conteneur.

La commande suivante démarre une machine virtuelle Java de conteneur à utiliser avec l'exemple de service de catalogue :

```
./startOgServer.sh c0 -catalogServiceEndPoints hostA:2809,hostB:2809
```

Le serveur de conteneur machines virtuelles Java utilise deux ports. Le port du gestionnaire haute disponibilité est utilisé pour les communications internes entre les serveurs de conteneur homologues et serveurs de catalogue. Le port d'écoute est utilisé pour les communications IIOP entre les serveurs de conteneur homologues, les serveurs de catalogue et les clients. L'hôte d'écoute est utilisé pour associer l'ORB à un adaptateur réseau spécifique. Si vous ne spécifiez rien, les deux ports sont dynamiquement sélectionnés. Toutefois, si vous souhaitez configurer les ports de manière explicite, comme dans un environnement de pare-feu, vous pouvez utiliser une option de ligne de commande pour spécifier le port ORB. Pour indiquer le port d'écoute et l'hôte d'écoute, utilisez les options de ligne de commande suivantes :

```
-listenerHost
<nom_hôte>
-listenerPort <port>
```

Le port d'écoute et l'hôte d'écoute peuvent être également définis à l'aide des propriétés listenerHost et listenerPort.

Pour indiquer le port du gestionnaire haute disponibilité, utilisez l'option de ligne de commande suivante :

-haManagerPort <port>
Synonyme avec port homologue. Indique le numéro de port utilisé par le gestionnaire de haute disponibilité. Si cette propriété n'est pas définie, le service de catalogue génère automatiquement un port disponible. Cette propriété s'applique à la fois au serveur conteneur et au service de catalogue. (Requis pour les environnements WebSphere Application Server uniquement.)

Le port du gestionnaire HA peut être également défini en utilisant la propriété de serveur HAManagerPort.

Lorsque la sécurité est activée, un port SSL (Secure Socket Layer) est aussi nécessaire. Pour indiquer le port SSL, utilisez l'option de ligne de commande suivante :

```
-jvmArgs -Dcom.ibm.CSI.SSLPort=<sslPort>
```

3. Démarrez les noeuds finaux client.

Les clients ont uniquement besoin de connaître les noeuds finaux d'écoute de service de catalogue. Les clients extraient les points de contact des machines virtuelles Java du serveur de conteneur, qui sont les machines virtuelles Java qui conservent les données, automatiquement du service de catalogue. Pour vous connecter au service de catalogue dans l'exemple précédente, le client doit envoyer la liste suivante de paires `host:port` à l'API connexion :

```
hostA:2809,hostB:2809
```

Le client peut également recevoir des rappels des serveurs de conteneur lors de l'utilisation de l'API DataGrid. Ces rappels communiquent en utilisant IIOP avec le port d'écoute ORB. Pour indiquer le port et l'adaptateur réseau pour recevoir des rappels, définissez les propriétés **listenerHost** et **listenerPort** dans le fichier de propriétés du client.

Lorsque la sécurité est activée, un port SSL (Secure Socket Layer) est aussi nécessaire. Pour indiquer le port SSL, utilisez la propriété système suivante lors du démarrage du processus client :

```
-Dcom.ibm.CSI.SSLPort=<sslPort>
```

Configuration de ports dans un environnement WebSphere Application Server

Les services de catalogue WebSphere eXtreme Scale, les serveurs de conteneur et les clients, lorsqu'ils s'exécutent dans des processus WebSphere Application Server, utilisent des ports et des services déjà définis pour le processus.

Pourquoi et quand exécuter cette tâche

Les sections suivantes décrivent les informations relatives à l'utilisation de ports dans le déploiement.

1. Noeuds finaux de service de catalogue

   Les services de catalogue WebSphere eXtreme Scale s'exécutent dans n'importe quel processus WebSphere Application Server et sont configurés en utilisant la console d'administration ou des tâches d'administration. Tous les ports sont hérités du processus, sauf pour le port client, qui est explicitement configuré. Pour plus d'informations sur les ports utilisés par le service de catalogue, voir «Planification des ports réseau», à la page 64. Pour plus d'informations sur la configuration d'un domaine de services de catalogue, voir Service de catalogue à haute disponibilité.

2. Noeuds finaux du serveur de conteneur

   Les serveurs de conteneur WebSphere eXtreme Scale sont hébergés dans des modules Java EE. Les serveurs de conteneur utilisent les ports définis pour le processus du serveur d'applications. Pour plus d'informations sur les ports utilisés par le service de conteneur, voir «Planification des ports réseau», à la page 64. Pour plus d'informations sur le démarrage d'un conteneur dans un module Java EE, tel qu'un module Enterprise JavaBeans (EJB) ou un module Web, voir «Configuration des applications WebSphere Application Server pour démarrer automatiquement les serveurs de conteneur», à la page 275.

3. Noeuds finaux du client

   Les clients WebSphere eXtreme Scale sont hébergés dans des modules Web Java EE ou EJB.

   Les clients se connectent à l'aide d'un programme au domaine de services de catalogue en utilisant l'API ObjectGridManager.connect(). Lorsque vous vous connectez à un domaine de services de catalogue hébergé dans la même cellule, la connexion client trouve automatiquement le domaine de services de catalogue par défaut en utilisant l'appel d'API suivant sur ObjectGridManager:

   ```
   connect(securityProps, overRideObjectGridXML)
   ```

   Si le domaine de services de catalogue par défaut est hébergé à distance (en dehors de la cellule), les noeuds finaux de service de catalogue doivent être définis en utilisant la méthode suivante dans l'API ObjectGridManager :

   ```
   connect(catalogServerAddresses, securityProps, overRideObjectGridXml)
   ```

   Si le domaine de services de catalogue par défaut est défini dans la cellule, l'API CatalogServerProperties peut être utilisée pour extraire les adresses de serveur de catalogue. La tâche d'administration XSDomainManagement peut également être utilisée pour extraire un domaine de services de catalogue configuré.

Serveurs avec plusieurs cartes réseau

Vous pouvez exécuter les processus eXtreme Scale sur un serveur doté de plusieurs cartes réseau.

Si un serveur ou un client est en cours d'exécution sur un serveur contenant plusieurs cartes réseau, vous devez spécifier le port réseau et le nom d'hôte dans votre configuration eXtreme Scale à lier à une carte spécifiée. Si cette configuration n'est pas spécifiée, l'environnement d'exécution eXtreme Scale en choisit une automatiquement, ce qui peut entraîner des échecs de connexion ou un ralentissement des performances.

Pour les serveurs de catalogue ou de conteneur, vous devez définir l'hôte d'écoute et le port d'écoute de l'une des manières suivantes :

- propriétés du serveur
- paramètre de ligne de commande dans le script startOgServer.sh | bat.

Pour les clients, vous ne pouvez pas utiliser la ligne de commande et vous devez utiliser les propriétés du client.

Configuration des transports

Les transports permettent l'échange d'objets et de données entre différents processus serveur dans votre configuration.

Pourquoi et quand exécuter cette tâche

Le mécanisme de transport principal est l'ORB (Object Request Broker). Ce mécanisme stocke les entrées de cache dans le segment de mémoire Java.

7.1.1+ Vous devez utiliser l'ORB comme mécanisme de transport dans les scénarios de configuration suivants :

- Lorsque vous utilisez un système autre que Linux x86 64 bits.
- Lorsque vous utilisez des serveurs de conteneur qui s'exécutent dans un environnement WebSphere Application Server.
- Lorsque vous utilisez des plug-ins d'expulsion ou des index composites.

7.1.1+ Si vous utilisez eXtremeMemory, un nouveau transport appelé eXtremeIO est utilisé. Avec eXtremeMemory, les entrées de cache sont stockées dans la mémoire native. La mémoire native ne passe pas par la récupération de place, ce qui permet de rendre les performances constantes et de prévoir les temps de réponse. Les objets sont sérialisés en octets dans le serveur de conteneur. Pour plus d'informations, voir «Configuration de IBM eXtremeMemory et de IBM eXtremeIO», à la page 278.

Configuration d'ORB

L'ORB (Object Request Broker) est utilisé par WebSphere eXtreme Scale pour communiquer sur une pile TCP. Utilisez le fichier `orb.properties` pour transmettre les propriétés utilisées par l'ORB pour modifier le comportement du transport de la grille de données. Aucune action n'est requise pour utiliser l'ORB fourni par WebSphere eXtreme Scale ou WebSphere Application Server pour vos serveurs WebSphere eXtreme Scale.

Configuration de la fonction ORB (Object Request Broker) dans un environnement WebSphere Application Server

Vous pouvez utiliser WebSphere eXtreme Scale avec des applications qui utilisent ORB (Object Request Broker) directement dans des environnements WebSphere Application Server or WebSphere Application Server Network Deployment.

Procédure

1. Nommez vos serveurs d'applications de façon appropriée.

   Dans un environnement WebSphere Application Server, les serveurs ne peuvent pas porter le même nom lorsqu'ils utilisent ORB pour communiquer entre eux. Vous pouvez contourner cette restriction en spécifiant la propriété système **-Dcom.ibm.websphere.orb.uniqueServerName=true** pour les processus de même nom. Par exemple, lorsque des serveurs avec le nom `server1` sur chaque noeud sont utilisés comme domaine de services de catalogue ou plusieurs agents de noeud sont utilisés pour former un domaine de services de catalogue.

2. Optimisez les propriétés ORB dans la configuration WebSphere Application Server.

   Voir «Propriétés ORB», à la page 492 pour plus d'informations sur les propriétés que vous pouvez optimiser. En fonction de la propriété, vous pouvez modifier un paramètre dans la console d'administration ou dans le fichier *racine_was*`properties/orb.properties`.

3. Si vous utilisez plusieurs cartes d'interface réseau, vous devez définir la valeur ORB_LISTENER_ADDRESS dans le panneau des ports dans la console d'administration de WebSphere Application Server. Répétez cette étape pour chaque serveur d'applications dans la configuration.

   a. Pour un serveur d'applications, cliquez sur **Serveurs** > **Serveurs d'applications** > ***server_name***. Dans Communications, cliquez sur **Ports**. Le panneau Ports du serveur spécifié s'affiche.

   b. Cliquez sur **Détails** et modifiez la valeur ORB_LISTENER_ADDRESS.

   c. Entrez l'adresse IP dans la zone **Hôte**. Cette valeur doit être une adresse privée pour un environnement à plusieurs interfaces réseau.

      Remarque : Les noms d'hôte DNS ne sont pas pris en charge pour la valeur ORB_LISTENER_ADDRESS.

   d. Entrez le numéro de port dans la zone **Port**. Le numéro de port définit le port pour lequel le service est configuré pour accepter les demandes des client. La valeur de port est utilisée avec le nom d'hôte.

Que faire ensuite

7.1.1+ Vous pouvez utiliser l'outil **wxsLogAnalyzer** pour vérifier les paramètres de l'ORB dans votre environnement. Pour plus d'informations, voir «Analyse des journaux et des données de trace», à la page 540.

Configuration d'ORB (Object Request Broker) avec des processus autonomes WebSphere eXtreme Scale

Vous pouvez utiliser WebSphere eXtreme Scale avec des applications qui utilisent Object Request Broker (ORB) directement dans des environnements ne contenant pas WebSphere Application Server ou WebSphere Application Server Network Deployment.

Avant de commencer

Si vous utilisez l'ORB dans le même processus que eXtreme Scale lorsque vous exécutez des applications, ou d'autres composants et infrastructures, non inclus avec eXtreme Scale, il se peut que vous deviez effectuer des tâches supplémentaires pour vous assurer que eXtreme Scale fonctionne correctement dans votre environnement.

Pourquoi et quand exécuter cette tâche

Ajoutez la propriété **ObjectGridInitializer** au fichier `orb.properties` pour initialiser l'utilisation de l'ORB dans votre environnement. Utilisez l'ORB pour activer la communication entre les processus eXtreme Scale et les autres processus de votre environnement.

Procédure

1. L'installation autonome n'inclut pas de fichier `orb.properties`. Vous devez placer le fichier `orb.properties` dans le répertoire `java/jre/lib`. Pour les descriptions des propriétés et des paramètres, voir «Propriétés ORB», à la page 492.
2. Dans le fichier `orb.properties`, tapez la ligne suivante et sauvegardez vos modifications :

```
org.omg.PortableInterceptor.ORBInitializerClass.com.ibm.ws.objectgrid.corba.ObjectGridInitializer
```

Résultats

eXtreme Scale initialise correctement l'ORB et coexiste avec d'autres applications pour lesquelles l'ORB est activé.

Pour utiliser une version personnalisée de l'ORB avec eXtreme Scale, voir «Configuration d'un ORB personnalisé».

Que faire ensuite

7.1.1+ Vous pouvez utiliser l'outil **xsLogAnalyzer** pour vérifier les paramètres de l'ORB dans votre environnement. Pour plus d'informations, voir «Analyse des journaux et des données de trace», à la page 540.

Configuration d'un ORB personnalisé

WebSphere eXtreme Scale utilise l'ORB (Object Request Broker) pour activer les communications entre les processus. Aucune action n'est requise pour utiliser pour vos serveurs WebSphere eXtreme Scale l'ORB fourni par WebSphere eXtreme Scale ou par WebSphere Application Server. L'utilisation des mêmes ORB pour vos clients WebSphere eXtreme Scale ne vous demandera guère plus. Si vous devez utiliser à la place d'un ORB personnalisé, l'ORB fourni avec IBM SDK est un bon choix, même si vous le configurer. Il est également possible d'utiliser d'autres ORB fournis par d'autres constructeurs, là aussi moyennant quelque configuration.

Avant de commencer

Déterminez si vous utilisez l'ORB fourni avec WebSphere eXtreme Scale ou WebSphere Application Server, l'ORB fourni avec IBM SDK, ou un ORB de fournisseur externe.

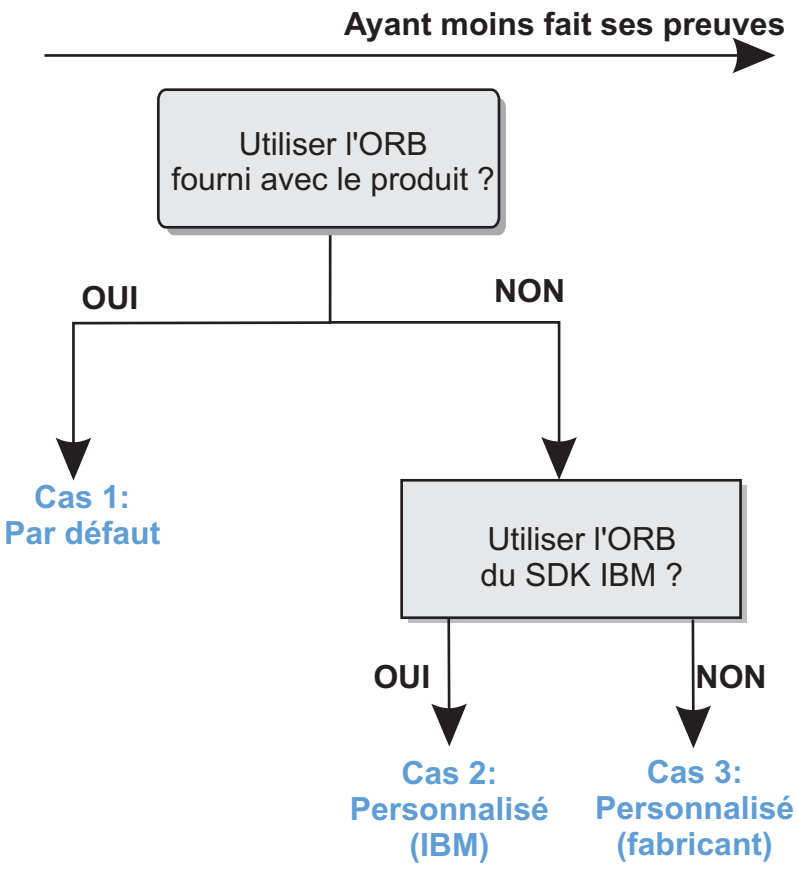


Figure 34. Choix de l'ORB

Vous n'êtes pas obligé de faire les mêmes choix pour les processus serveur WebSphere eXtreme Scale et les processus client WebSphere eXtreme Scale. eXtreme Scale prend en charge les kits de développeur de la plupart des fournisseurs, mais il est recommandé d'utiliser pour vos processus, tant serveur que client, l'ORB fourni avec eXtreme Scale. eXtreme Scale ne prend pas en charge l'ORB fourni avec le Java Development Kit (JDK) de Sun Microsystems.

Pourquoi et quand exécuter cette tâche

Avant d'utiliser l'ORB que vous avez choisi, familiarisez-vous avec la configuration requise.

Cas n° 1 : ORB par défaut

- Pour vos processus serveur WebSphere eXtreme Scale, aucune configuration n'est requise pour utiliser l'ORB fourni avec WebSphere eXtreme Scale ou avec WebSphere Application Server.
- Pour vos processus client WebSphere eXtreme Scale, un minimum de configuration du chemin d'accès aux classes est requis pour pouvoir utiliser l'ORB fourni avec WebSphere eXtreme Scale ou avec WebSphere Application Server.

Cas n° 2 : ORB personnalisé (IBM)

Pour configurer vos processus client WebSphere eXtreme Scale afin qu'ils utilisent l'ORB fourni avec le IBM SDK, voyez les instructions plus bas. Vous pouvez utiliser l'ORB IBM, que vous utilisiez le IBM SDK ou un autre kit de développement. Vous pouvez utiliser IBM SDK Version 5 ou suivante.

Cas 3: ORB personnalisé (fourni par un fournisseur externe)

L'utilisation d'un ORB de fournisseur pour les processus client WebSphere eXtreme Scale est l'option qui a subi le moins de test. Avant de contacter le support technique IBM, vous devez vous assurer que les problèmes rencontrés en utilisant des ORB d'éditeurs de logiciels indépendants sont bien reproductibles avec l'ORB IBM ORB et un JRE compatible.

L'ORB fourni avec le Java Development Kit (JDK) de Sun Microsystems n'est pas pris en charge.

Procédure

- Configurez vos processus client pour qu'ils utilisent l'un des ORB par défaut (**Cas 1**). Utilisez l'argument JVM suivant :

```
-jvmArgs –Djava.endorsed.dirs=default_ORB_directory${pathSeparator}JRE_HOME/lib/endorsed
```

  Le répertoire ORB par défaut est : *wxs_home*`/lib/endorsed`. Il peut être aussi nécessaire de mettre à jour les propriétés suivantes dans le fichier `orb.properties` :

  ```
  org.omg.CORBA.ORBClass=com.ibm.CORBA.iiop.ORB
  org.omg.CORBA.ORBSingletonClass=com.ibm.rmi.corba.ORBSingleton
  ```

- Configurez les processus client ou serveur pour qu'ils utilisent le IBM SDK version 5 **(cas n° 2)**.

  1. Copiez les fichiers de l'ORB d'archive Java (JAR) dans un répertoire vide, ou le répertoire *custom_ORB_directory*.
     - `ibmorb.jar`
     - `ibmorbapi.jar`
  2. Dans les scripts qui lancent la commande Java, spécifiez le répertoire *`custom_ORB_directory`* comme répertoire validé.

     Conseil : Si vos commandes Java spécifient déjà un répertoire validé, une autre option consiste à placer le répertoire *`custom_ORB_directory`* sous le répertoire validé existant. En plaçant le répertoire *`custom_ORB_directory`* sous le répertoire validé existant, vous n'avez pas à mettre à jour les scripts. Si vous décidez de mettre à jour les scripts, veillez à ajouter le répertoire *`custom_ORB_directory`* comme préfixe à l'argument existant `–Djava.endorsed.dirs=` au lieu de remplacer complètement l'argument.
     - Modifiez les scripts pour un environnement eXtreme Scale autonome.

       Modifiez le chemin pour la variable *OBJECTGRID_ENDORSED_DIRS* dans le fichier `setupCmdLine.bat|sh` pour indiquer le répertoire *custom_ORB_directory*. Sauvegardez vos modifications.
     - Modifiez les scripts lorsque'eXtreme Scale est imbriqué dans un environnement WebSphere Application Server.

       Ajoutez la propriété système et les paramètres suivants au script `startOgServer` :

       ```
       -jvmArgs –Djava.endorsed.dirs=répertoire_ORB_personnalisé
       ```
     - Modifiez les scripts personnalisés qui vous servent à démarrer un processus d'application client ou un processus serveur.

       ```
       -Djava.endorsed.dirs=répertoire_ORB_personnalisé
       ```

Configuration des clients

Vous pouvez configurer WebSphere eXtreme Scale pour qu'il soit exécuté dans un environnement autonome ou configurer eXtreme Scale pour l'exécuter dans un environnement avec WebSphere Application Server. Pour qu'un déploiement WebSphere eXtreme Scale sélectionne les modifications de configuration dans la grille de serveurs, vous devez redémarrer les processus pour que ces modifications entrent en vigueur au lieu d'être appliquées de manière dynamique. Toutefois, côté client, vous ne pouvez pas modifier les paramètres de configuration d'une instance de client existante, mais vous pouvez créer une instance de client avec les paramètres nécessaires en utilisant un fichier XML ou à l'aide d'un programme. Lorsque vous créez un client, vous pouvez remplacer les paramètres par défaut provenant de la configuration de serveur actuelle.

Vous pouvez configurer un client eXtreme Scale des différentes manières suivantes, dont chacune peut être effectuée par programmation ou à l'aide d'un fichier XML de substitution par le client :

- configuration XML
- configuration par programmation
- configuration Spring Framework
- désactivation du cache local

Vous pouvez remplacer les plug-in suivants sur un client :

- **plug-in ObjectGrid**
  - plug-in TransactionCallback
  - plug-in ObjectGridEventListener
- **plug-in BackingMap**
  - plug-in Evictor
  - plug-in MapEventListener
  - attribut numberOfBuckets
  - attribut ttlEvictorType
  - attribut timeToLive

Configuration des clients avec la configuration XML

Vous pouvez utiliser le fichier XML de configuration pour modifier les paramètres du client.

Pourquoi et quand exécuter cette tâche

Pour modifier les paramètres d'un client WebSphere eXtreme Scale , vous devez créer un fichier XML ObjectGrid ayant une structure semblable à celle du fichier utilisé pour le serveur de conteneur.

Vous pouvez remplacer les paramètres suivants sur le client :

1. Créez une instance ObjectGrid propre au client.
2. Copiez le fichier XML ObjectGrid utilisé pour ouvrir le serveur.
3. Editez le nouveau fichier pour personnaliser le client :
   - Pour définir ou mettre à jour les attributs du client, indiquez une nouvelle valeur ou modifiez la valeur existante.
   - Pour supprimer un plug-in du client, utilisez la chaîne vide comme valeur pour l'attribut className.

- Pour modifier un plug-in existant, indiquez une nouvelle valeur pour l'attribut className.
- Vous pouvez aussi ajouter les plug-in pris en charge pour les remplacements par les clients : TRANSACTION_CALLBACK, OBJECTGRID_EVENT_LISTENER, EVICTOR et MAP_EVENT_LISTENER.

4. Créez un client à l'aide du nouveau fichier XML de remplacement par les clients.

Procédure

1. Créez un fichier XML de configuration ObjectGrid pour le client qui ait une structure semblable à celle du fichier pour le serveur de conteneur.

   Supposons que le fichier XML a été associé à un fichier XML de stratégie de déploiement et que ces fichier ont été utilisés pour démarrer un serveur de conteneur.

   companyGridServerSide.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">

    <objectGrids>
        <objectGrid name="CompanyGrid">
            <bean id="TransactionCallback"
                classname="com.company.MyTxCallback" />
            <bean id="ObjectGridEventListener"
                classname="com.company.MyOgEventListener" />
            <backingMap name="Customer"
                pluginCollectionRef="customerPlugins" />
            <backingMap name="Item" />
            <backingMap name="OrderLine" numberOfBuckets="1049"
                timeToLive="1600" ttlEvictorType="LAST_ACCESS_TIME" />
            <backingMap name="Order" lockStrategy="PESSIMISTIC"
                pluginCollectionRef="orderPlugins" />
        </objectGrid>
    </objectGrids>

    <backingMapPluginCollections>
        <backingMapPluginCollection id="customerPlugins">
            <bean id="Evictor"
                classname="com.ibm.websphere.objectGrid.plugins.builtins.LRUEvictor" />
            <bean id="MapEventListener"
                classname="com.company.MyMapEventListener" />
        </backingMapPluginCollection>
        <backingMapPluginCollection id="orderPlugins">
            <bean id="MapIndexPlugin"
                classname="com.company.MyMapIndexPlugin" />
        </backingMapPluginCollection>
    </backingMapPluginCollections>
</objectGridConfig>
```

   Sur un serveur de conteneur, l'instance ObjectGrid nommée CompanyGrid se comporte conformément à ce qui est défini dans le fichier `companyGridServerSide.xml`. Par défaut, les paramètres du client CompanyGrid sont identiques à ceux de l'instance CompanyGrid qui s'exécute sur le serveur.

   Le fichier XML ObjectGrid suivant peut être utilisé pour définir certains attributs et plug-in du client CompanyGrid.

   `companyGridClientSide.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">

    <objectGrids>
        <objectGrid name="CompanyGrid">
            <bean id="TransactionCallback"
                classname="com.company.MyClientTxCallback" />
            <bean id="ObjectGridEventListener" classname="" />
            <backingMap name="Customer" numberOfBuckets="1429"
                pluginCollectionRef="customerPlugins" />
```

```
            <backingMap name="Item" />
            <backingMap name="OrderLine" numberOfBuckets="701"
                timeToLive="800" ttlEvictorType="LAST_ACCESS_TIME" />
            <backingMap name="Order" lockStrategy="PESSIMISTIC"
                pluginCollectionRef="orderPlugins" />
        </objectGrid>
    </objectGrids>

    <backingMapPluginCollections>
        <backingMapPluginCollection id="customerPlugins">
            <bean id="Evictor"
                className="com.ibm.websphere.objectGrid.plugins.builtins.LRUEvictor" />
            <bean id="MapEventListener" className="" />
        </backingMapPluginCollection>
        <backingMapPluginCollection id="orderPlugins">
            <bean id="MapIndexPlugin"
                className="com.company.MyMapIndexPlugin" />
        </backingMapPluginCollection>
    </backingMapPluginCollections>
</objectGridConfig>
```

Résumé des remplacements définis :

- Le TransactionCallback du client est com.company.MyClientTxCallback et non le paramètre com.company.MyTxCallback du côté serveur.
- Le client n'est associé à aucun plug-in ObjectGridEventListener car la valeur className est la chaîne vide.
- Le client associe numberOfBuckets à la valeur 1429 pour le backingMap Customer, conserve le plug-in Evictor et supprime le plug-in MapEventListener.
- Les attributs numberOfBuckets et timeToLive du backingMap OrderLine ont été modifiés.
- Bien qu'un attribut lockStrategy différent ait été indiqué, les conséquences sont nulles car cet attribut n'est pas pris en charge pour un remplacement par le client.

2. Créez le client en utilisant le fichier XML.

   Pour créer le client CompanyGrid à l'aide du fichier `companyGridClientSide.xml`, transmettez le fichier XML ObjectGrid sous la forme d'une URL à l'une des méthodes de connexion dans l'interface ObjectGridManager :

```
ObjectGridManager ogManager =
 ObjectGridManagerFactory.ObjectGridManager();
ClientClusterContext clientClusterContext =
 ogManager.connect("MyServer1.company.com:2809", null, new URL(
                "file:xml/companyGridClientSide.xml"));
```

Activation du mécanisme d'invalidation de client

Dans un environnement WebSphere eXtreme Scale réparti, le côté client dispose par défaut d'un cache local lorsqu'il utilise la stratégie de verrouillage optimiste ou lorsque le verrouillage est désactivé. Ce cache contient ses propres données locales. Si un client eXtreme Scale valide une mise à jour, celle-ci est envoyée au cache local du client et au serveur. Toutefois, les autres clients eXtreme Scale ne reçoivent pas les informations relatives à cette mise à jour et leurs données risquent de devenir obsolètes.

Cache local

Les applications doivent être informées de l'éventuelle présence de données obsolètes dans le client eXtreme Scale. Vous pouvez utiliser la classe intégrée ObjectGridEventListener

com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener JMS (Java Message Service) pour activer le mécanisme d'invalidation de client dans un environnement eXtreme Scale distribué.

Le mécanisme d'invalidation de client permet de résoudre les problèmes liés à la présence de données obsolètes dans le cache local du client en environnement eXtreme Scale réparti. Ce mécanisme vérifie que la mémoire est synchronisée avec les serveurs ou les autres clients. Toutefois, même lorsque ce mécanisme existe, la mise à jour de la mémoire n'est pas immédiate. Lorsque l'environnement d'exécution de eXtreme Scale publie des mises à jour, un délai est généré.

Il existe deux modèles pour le mécanisme d'invalidation de client dans un environnement eXtreme Scale réparti :

- Modèle client-serveur : tous les processus serveur ont un rôle de diffuseur de publications qui publie toutes les modifications transactionnelles vers la destination JMS désignée. Tous les processus client ont un rôle de récepteur : ils reçoivent toutes les modifications transactionnelles à partir de la destination JMS désignée.
- Modèle client ayant les deux rôles : aucune interaction n'existe entre les processus serveur et la destination JMS. Tous les processus client ont le rôle de diffuseur de publications JMS et de récepteur. Les modifications transactionnelles effectuées sur le client sont publiées vers la destination JMS et tous les clients les reçoivent.

Pour plus d'information, consultez «Programme d'écoute d'événement JMS», à la page 232.

Modèle client-serveur

Dans un modèle client-serveur, les serveurs ont le rôle de diffuseur de publications JMS et le client a le rôle de récepteur JMS.

Exemple XML de modèle client-serveur

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="AgentObjectGrid">
      <bean id="ObjectGridEventListener"
        className="com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener">
        <property name="invalidationModel" type="java.lang.String" value="CLIENT_SERVER_MODEL" description="" />
        <property name="invalidationStrategy" type="java.lang.String" value="PUSH" description="" />
        <property name="mapsToPublish" type="java.lang.String" value="agent;profile;pessimisticMap" description="" />
        <property name="jms_topicConnectionFactoryJndiName" type="java.lang.String" value="defaultTCF" description="" />
        <property name="jms_topicJndiName" type="java.lang.String" value="defaultTopic" description="" />
        <property name="jms_topicName" type="java.lang.String" value="defaultTopic" description="" />
        <property name="jms_userid" type="java.lang.String" value="" description="" />
        <property name="jms_password" type="java.lang.String" value="" description="" />
        <property name="jndi_properties" type="java.lang.String"
          value="java.naming.factory.initial=org.apache.activemq.jndi.ActiveMQInitialContextFactory;
     java.naming.provider.url=
    tcp://localhost:61616;connectionFactoryNames=defaultTCF;topic.defaultTopic=defaultTopic"
          description="jndi properties" />
      </bean>

      <backingMap name="agent" readOnly="false" pluginCollectionRef="agent" preloadMode="false"
        lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
        timeToLive="28800" />
      <backingMap name="profile" readOnly="false" pluginCollectionRef="profile" preloadMode="false"
        lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
        timeToLive="2700" />
      <backingMap name="pessimisticMap" readOnly="false" pluginCollectionRef="pessimisticMap" preloadMode="false"
        lockStrategy="PESSIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
        timeToLive="2700" />
      <backingMap name="excludedMap1" readOnly="false" pluginCollectionRef="excludedMap1" preloadMode="false"
        lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
        timeToLive="2700" />
      <backingMap name="excludedMap2" readOnly="false" pluginCollectionRef="excludedMap2" preloadMode="false"
        lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
```

```
        timeToLive="2700" />
    </objectGrid>
  </objectGrids>

  <backingMapPluginCollections>
    <backingMapPluginCollection id="agent">
      <bean id="ObjectTransformer" className="com.ibm.ws.objectgrid.test.scenario.AgentObjectTransformer" />
    </backingMapPluginCollection>
    <backingMapPluginCollection id="profile">
      <bean id="ObjectTransformer" className="com.ibm.ws.objectgrid.test.scenario.ProfileObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
        <property name="maxSize" type="int" value="2000" description="set max size for LRU evictor" />
        <property name="sleepTime" type="int" value="15" description="evictor thread sleep time" />
        <property name="numberOfLRUQueues" type="int" value="50" description="set number of LRU queues" />
      </bean>
    </backingMapPluginCollection>

    <backingMapPluginCollection id="pessimisticMap" />
    <backingMapPluginCollection id="excludedMap1" />
    <backingMapPluginCollection id="excludedMap2" />
  </backingMapPluginCollections>

</objectGridConfig>
```

Modèle client ayant les deux rôles

Dans ce modèle, chaque client a le rôle de diffuseur de publications JMS et de récepteur. Le client publie chaque modification transactionnelle validée vers une destination JMS désignée et reçoit toutes les modifications des autres clients. Aucune interaction n'a lieu entre le serveur et JMS.

Exemple XML de modèle où le client a les deux rôles

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="AgentObjectGrid">
      <bean id="ObjectGridEventListener"
        className="com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener">
        <property name="invalidationModel" type="java.lang.String" value="CLIENT_AS_DUAL_ROLES_MODEL" description="" />
        <property name="invalidationStrategy" type="java.lang.String" value="PUSH" description="" />
        <property name="mapsToPublish" type="java.lang.String" value="agent;profile;pessimisticMap" description="" />
        <property name="jms_topicConnectionFactoryJndiName" type="java.lang.String" value="defaultTCF" description="" />
        <property name="jms_topicJndiName" type="java.lang.String" value="defaultTopic" description="" />
        <property name="jms_topicName" type="java.lang.String" value="defaultTopic" description="" />
        <property name="jms_userid" type="java.lang.String" value="" description="" />
        <property name="jms_password" type="java.lang.String" value="" description="" />
        <property name="jndi_properties" type="java.lang.String"
          value="java.naming.factory.initial=org.apache.activemq.jndi.ActiveMQInitialContextFactory;java.naming.provider.url=
     tcp://localhost:61616;connectionFactoryNames=defaultTCF;topic.defaultTopic=defaultTopic"
          description="jndi properties" />
      </bean>

      <backingMap name="agent" readOnly="false" pluginCollectionRef="agent" preloadMode="false"
        lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
        timeToLive="28800" />
      <backingMap name="profile" readOnly="false" pluginCollectionRef="profile" preloadMode="false"
        lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
        timeToLive="2700" />
      <backingMap name="pessimisticMap" readOnly="false" pluginCollectionRef="pessimisticMap" preloadMode="false"
        lockStrategy="PESSIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
        timeToLive="2700" />
      <backingMap name="excludedMap1" readOnly="false" pluginCollectionRef="excludedMap1" preloadMode="false"
        lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
        timeToLive="2700" />
      <backingMap name="excludedMap2" readOnly="false" pluginCollectionRef="excludedMap2" preloadMode="false"
        lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
        timeToLive="2700" />
    </objectGrid>
  </objectGrids>

  <backingMapPluginCollections>
    <backingMapPluginCollection id="agent">
      <bean id="ObjectTransformer" className="com.ibm.ws.objectgrid.test.scenario.AgentObjectTransformer" />
    </backingMapPluginCollection>
    <backingMapPluginCollection id="profile">
      <bean id="ObjectTransformer" className="com.ibm.ws.objectgrid.test.scenario.ProfileObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
        <property name="maxSize" type="int" value="2000" description="set max size for LRU evictor" />
        <property name="sleepTime" type="int" value="15" description="evictor thread sleep time" />
        <property name="numberOfLRUQueues" type="int" value="50" description="set number of LRU queues" />
      </bean>
    </backingMapPluginCollection>

    <backingMapPluginCollection id="pessimisticMap" />
```

```
    <backingMapPluginCollection id="excludedMap1" />
    <backingMapPluginCollection id="excludedMap2" />
  </backingMapPluginCollections>

</objectGridConfig>
```

Définition des valeurs de délai d'attente de nouvelles tentatives de demande

Avec des mappes fiables, vous pouvez fournir une valeur de délai d'attente en millisecondes à WebSphere eXtreme Scale pour les demandes de transaction.

Pourquoi et quand exécuter cette tâche

Vous pouvez définir la valeur du délai d'attente dans le fichier des propriétés du client ou dans une session. La valeur de session remplace la valeur dans les propriétés du client. Si la valeur définie est supérieure à zéro, la demande est réessayée jusqu'à ce que le dépassement du délai d'attente soit atteint ou qu'une erreur permanente se produise. Une erreur permanente peut être une exception `DuplicateKeyException`. La valeur zéro indique le mode fail-fast et eXtreme Scale ne retente pas la transaction, quelle que soit son type.

Pendant l'exécution, la valeur de délai d'expiration des transactions est utilisée avec la valeur de délai d'attente de nouvelle tentative, ce qui garantit que le délai d'attente de relance ne dépasse pas le délai d'expiration des transactions.

Deux types de transactions existent : les transactions de validation automatique et les transactions qui utilisent des méthodes explicites begin et commit. Les exceptions valides de tentatives diffèrent de ces deux types de transactions :

- Pour les transactions appelées au sein d'une session, les transactions sont retentées pour les exceptions CORBA SystemException et eXtreme Scale TargetNotAvailable.
- Pour les transactions de validation automatique, les transactions sont retentées pour les exceptions de disponibilité CORBA eXtreme Scale. Ces exceptions incluent les exceptions ReplicationVotedToRollbackTransactionException, TargetNotAvailable et AvailabilityException.

Les erreurs d'application et les autres erreurs permanentes s'exécutent immédiatement et le client ne retente pas la transaction. Ces erreurs permanentes incluent les exceptions DuplicateKeyException et KeyNotFoundException. Utilisez le paramètre fail-fast pour exécuter toutes les exceptions sans retenter les transactions.

Exceptions pour lesquels le client retente la transaction :

- ReplicationVotedToRollbackTransactionException (uniquement en validation automatique)
- TargetNotAvailable
- org.omg.CORBA.SystemException
- AvailabilityException (uniquement en validation automatique)
- LockTimeoutException (uniquement en validation automatique)
- UnavailableServiceException (uniquement en validation automatique)

Exceptions permanentes pour lesquelles la transaction n'est pas retentée :

- DuplicateKeyException
- KeyNotFoundException

- LoaderException
- TransactionAffinityException
- LockDeadlockException
- OptimisticCollisionException

Procédure

- Définissez la valeur du délai d'attente dans un fichier de propriétés client.

  Pour définir la valeur de requestRetryTimeout dans un client, ajoutez ou modifiez la propriété dans le Fichier de propriétés du client. Les propriétés du client sont définies par défaut dans le fichier `objectGridClient.properties`. La propriété requestRetryTimeout est définie en millisecondes. Une valeur supérieure à zéro indique que la demande doit être réessayée en cas de survenue d'exceptions pour lesquelles le retry est possible. Une valeur de `0` indique que les échecs ne donnent lieu à aucune nouvelle tentative dans les exceptions. Pour utiliser le comportement par défaut, supprimez la propriété ou donnez-lui une valeur de `-1`. Exemple de valeur dans le fichier `objectGridClient.properties` :

  ```
  requestRetryTimeout = 30000
  ```

  La valeur de requestRetryTimeout est spécifiée en millisecondes. Dans l'exemple, si la valeur est utilisée dans une instance ObjectGrid, la valeur de requestRetryTimeout sera de 30 secondes.

- Définissez la valeur du délai d'attente à l'aide d'un programme.

  Pour définir par programmation les propriétés du client, commencez par créer un fichier de propriétés dans un `<emplacement>` approprié à votre application. Dans l'exemple suivant, le fichier des propriétés du client correspond au fragment de code `objectGridClient.properties` de la section précédente. Après vous être connecté à une instance ObjectGridManager, définissez les propriétés du client, comme indiqué. Ensuite, lorsque vous avez une instance ObjectGrid, l'instance a les propriétés client que vous avez définies dans le fichier. A chaque fois que vous serez amené à modifier ce fichier, vous devrez explicitement obtenir une nouvelle instance ObjectGrid.

  ```
  ObjectGridManager manager = ObjectGridManagerFactory.getObjectGridManager();
  String objectGridName = "testObjectGrid";
  URL clientXML = null;
  ClientClusterContext ccc = manager.connect("localhost:2809", null, clientXML);
  File file = new File("<location>/objectGridClient.properties");
  URL url = file.toURI().toURL();
  ccc.setClientProperties(objectGridName, url);
  ObjectGrid objectGrid = ogManager.getObjectGrid(ccc, objectGridName);
  ```

- Définissez le fichier de remplacement pendant une validation de session.

  Pour définir dans l'objet Session pendant combien de temps effectuer de nouvelles tentatives ou pour remplacer la propriété client requestRetryTimeout, appelez la méthode setRequestRetryTimeout(long) dans l'interface Session.

  ```
  Session sessionA = objectGrid.getSession();
  sessionA.setRequestRetryTimeout(30000);
  ObjectMap mapA = sessionA.getMap("payroll");
  String key = "key:" + j;
  mapA.insert(key, "valueA");
  ```

  Cette session utilise à présent la valeur 30 000 millisecondes (30 secondes) pour requestRetryTimeout, quelle que soit la valeur définie dans le fichier des propriétés du client. Pour plus d'informations sur l'interface de session, voir Utilisation des sessions pour accéder aux données de la grille.

Configuration de l'intégration du cache

WebSphere eXtreme Scale peut s'intégrer aux autres produits de mise en cache. Vous pouvez aussi utiliser le fournisseur de cache dynamique WebSphere eXtreme Scale pour connecter WebSphere eXtreme Scale au composant de cache dynamique WebSphere Application Server. Autre extension de WebSphere Application Server : le gestionnaire de sessions HTTP WebSphere eXtreme Scale, qui permet la mise en cache des sessions HTTP.

Configuration de gestionnaires de sessions HTTP

Le gestionnaire de sessions HTTP offre des fonctions de réplication de sessions pour une application associée. Le gestionnaire de sessions fonctionne avec le conteneur Web pour créer et gérer les cycles de vie des sessions HTTP qui sont associés à l'application.

Configuration du gestionnaire de sessions HTTP avec WebSphere Application Server

Alors que WebSphere Application Server offre une fonction de gestion de session, les performances se dégradent alors que le nombre de demandes augmente. WebSphere eXtreme Scale est livré avec une implémentation de la gestion des sessions qui fournit la réplication de sessions, la haute disponibilité, une meilleure évolutivité et des options de configuration plus robustes.

Avant de commencer

- WebSphere eXtreme Scale doit être installé dans votre cellule WebSphere Application Server ou WebSphere Application Server Network Deployment pour pouvoir utiliser le gestionnaire de sessions d'eXtreme Scale. Pour plus d'informations, voir «Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client avec WebSphere Application Server», à la page 163.
- La sécurité globale doit être activée dans la console d'administration de WebSphere Application Server si les serveurs de catalogues de votre domaine de services de catalogue ont SSL activé ou si vous voulez utiliser SSL pour un domaine de services de catalogue avec SSL pris en charge. Vous activer SSL pour un serveur de catalogue en affectant à l'attribut la valeur `SSL-Required` dans Fichier de propriétés du serveur. Pour plus d'informations sur la configuration de la sécurité globale, voir Paramètres de sécurité globale.

Pourquoi et quand exécuter cette tâche

Le gestionnaire de sessions HTTP de WebSphere eXtreme Scale prend en charge les serveurs imbriqués et éloignés pour la mise en cache.

- **Scénario de serveurs imbriqués**

  Dans ce scénario, les serveurs WebSphere eXtreme Scale sont regroupés dans les processus où les servlets sont exécutés. Le gestionnaire de sessions peut communiquer directement avec l'instance ObjectGrid locale, pour éviter les retards coûteux du réseau.

  Si vous utilisez WebSphere Application Server, placez dans les répertoires `META-INF` de vos fichiers d'archive Web (WAR) les fichiers *rép_base_wxs*`/session/samples/objectGrid.xml` et *rép_base_wxs*`/session/samples/objectGridDeployment.xml` fournis. eXtreme Scale détecte automatiquement ces fichiers au démarrage de l'application et démarre automatiquement les conteneurs eXtreme Scale dans le même processus que le gestionnaire de sessions.

Vous pouvez modifier le fichier `objectGridDeployment.xml` suivant que vous souhaitiez utiliser une réplication synchrone ou asynchrone et en fonction du nombre de fragments réplique à configurer.

- **Scénario de serveurs éloignés**

  Dans les scénarios de serveurs distants, les serveurs conteneurs s'exécutent dans des processus différents que les servlets. Le gestionnaire de sessions communique avec un serveur conteneur distant. Pour pouvoir utiliser un serveur distant connecté à un réseau, le gestionnaire de sessions doit être configuré avec les noms d'hôte et les numéros de port du domaine de services de catalogue. Le gestionnaire de sessions utilise ensuite une connexion client eXtreme Scale pour communiquer avec le serveur de catalogues et avec les serveurs conteneurs.

  Si les serveurs de conteneur sont démarrés dans des processus autonomes indépendants, démarrez les conteneurs eXtreme Scale avec les fichiers `objectGridStandAlone.xml` et `objectGridDeploymentStandAlone.xml` fournis dans le répertoire des exemples du gestionnaire de sessions.

Procédure

1. Raccordez votre application de sorte qu'elle puisse utiliser le gestionnaire de sessions. Pour utiliser le gestionnaire de sessions, vous devez ajouter les déclarations de filtre appropriées aux descripteurs de déploiement Web de l'application. En outre, les paramètres de configuration du gestionnaire de sessions sont transmis au gestionnaire de sessions sous la forme de paramètres d'initialisation du contexte de servlet dans les descripteurs de déploiement. Vous pouvez introduire ces informations dans votre application de différentes manières :

   - **Raccord automatique avec WebSphere Application Server**

     Vous pouvez configurer votre application pour qu'elle utilise un gestionnaire WebSphere eXtreme Scale de sessions HTTP lorsque vous installez votre application. Vous pouvez également modifier la configuration de l'application ou du serveur pour qu'ils utilisent le gestionnaire WebSphere eXtreme Scale de sessions HTTP. Pour plus d'informations, voir «Fractionnement automatique des applications pour la gestion de session HTTP dans WebSphere Application Server», à la page 305.

   - **Raccord automatique de l'application avec des propriétés personnalisées**

     Vous n'avez pas besoin de raccorder manuellement vos applications lorsqu'elles s'exécutent dans WebSphere Application Server ou dans WebSphere Application Server Network Deployment.

     Ajoutez une propriété personnalisée à une cellule ou à un serveur pour définir à cette étendue le fichier `splicer.properties` pour toutes les applications Web. Utilisez les étapes suivantes pour configurer la propriété personnalisée :

     a. Dans la console d'administration de WebSphere Application Server, accédez au chemin correct de l'endroit où vous voulez définir la propriété personnalisée pour indiquer l'emplacement du fichier `splicer.properties`.

        – Pour définir la propriété personnalisée pour toutes les applications ou pour une application spécifique, cliquez sur **Administration du système** > **Cellule** > **Propriétés personnalisées**.

        – Pour définir la propriété personnalisée à appliquer à toutes les applications sur un serveur d'applications spécifique, cliquez sur **Serveur d'applications** > ***<nom_serveur>*** > **Administration** > **Propriétés personnalisées**. Le nom de propriété est

com.ibm.websphere.xs.sessionFilterProps et sa valeur se trouve dans le fichier splicer.properties dont a besoin votre application. Exemple de chemin d'emplacement d'un fichier : /opt/splicer.properties.

b. Ajoutez la propriété personnalisée com.ibm.websphere.xs.sessionFilterProps. La valeur de cette propriété personnalisée indique l'emplacement du fichier splicer.properties à éditer. Le fichier existe dans le gestionnaire_déploiement. Si vous voulez indiquer le fichier splicer.properties pour une application spécifique à l'aide d'une propriété personnalisée au niveau de la cellule, entrez le nom de la propriété personnalisée comme <*application_name*>,com.ibm.websphere.xs.sessionFilterProps, où *application_name* est le nom de l'application pour laquelle vous voulez appliquer la propriété personnalisée.

Important : Vérifiez que le fichier mis à jour splicer.properties se trouve dans le même chemin sur tous les noeuds contenant un serveur d'applications hébergeant l'application ou les applications qui sont à raccorder pour la réplication de session.

Les portées de cellule, de serveur et d'application sont les seules portées disponibles lors d'une exécution dans un gestionnaire de déploiement. Si vous avez besoin d'une autre portée, raccordez manuellement vos applications Web.

A faire : Notez aussi que le raccordement automatique ne fonctionne que si tous les noeuds exécutant l'application contiennent le fichier splicer.properties dans le même chemin. Dans le cas d'environnements mixtes contenant des noeuds Windows et UNIX, cette manière de procéder n'est pas possible et vous devez raccorder manuellement l'application.

- **Raccorder l'application avec le script addObjectGridFilter**

  Utilisez un script de ligne de commande fourni avec eXtreme Scale pour raccorder une application avec des déclarations de filtre et une configuration sous forme de paramètres d'initialisation de contexte de servlet. Pour un déploiement WebSphere Application Server, ce script se trouve dans <was_home>/optionalLibraries/ObjectGrid/session/bin/addObjectGridFilter.bat/sh. Pour un déploiement autonome, le script se trouve dans WXS_HOME/ObjectGrid/session/bin/addObjectGridFilter.sh/bat. Le script **oddObjectGridFilter** utilise deux paramètres :

  – Application : chemin absolu du fichier archive d'entreprise à raccorder
  – Chemin absolu du fichier de propriétés du raccordeur qui contient des propriétés de configuration.

  Voici le format du script :

  Windows

  ```
  addObjectGridFilter.bat [ear_file] [splicer_properties_file]
  ```

  UNIX

  ```
  addObjectGridFilter.sh [ear_file] [splicer_properties_file]
  ```

  UNIX **Exemple en utilisant eXtreme Scale installé sur WebSphere Application Server sur UNIX :**

  a. cd *rép_base_wxs*/optionalLibraries/ObjectGrid/session/bin
  b. addObjectGridFilter.sh /tmp/mySessionTest.ear *racine_was*/optionalLibraries/ObjectGrid/session/samples/splicer.properties

UNIX **Exemple en utilisant eXtreme Scale installé dans un répertoire autonome sur UNIX :**

a. `cd racine_was/session/bin`

b. `addObjectGridFilter.sh /tmp/mySessionTest.ear racine_was/session/samples/splicer.properties`

Le filtre de servlet qui est raccordé conserve les valeurs de configuration par défaut. Vous pouvez remplacer ces valeurs par défaut par des options de configuration que vous spécifiez dans le fichier de propriétés, dans le second argument. Pour une liste des paramètres que vous pouvez utiliser, voir «Paramètres d'initialisation du contexte de servlet», à la page 321.

Vous pouvez modifier et utiliser l'exemple de fichier `splicer.properties` fourni avec l'installation d'eXtreme Scale. Vous pouvez également utiliser le script **addObjectGridServlets**, qui insère le gestionnaire de sessions en étendant chaque servlet. Mais le script recommandé est le script **addObjectGridFilter**.

- **Raccorder manuellement l'application avec le script de génération Ant**

  WebSphere eXtreme Scale est fourni avec un fichier `build.xml` qui peut être utilisé par Apache Ant, qui est inclus dans le dossier *racine_was*`/bin` d'une installation WebSphere Application Server. Vous pouvez modifier le fichier `build.xml` pour changer les propriétés de configuration du gestionnaire de sessions. Les propriétés de configuration sont identiques aux noms de propriété dans le fichier `splicer.properties`. Pour modifier le fichier `build.xml`, appelez le processus Ant en exécutant la commande suivante :

  – UNIX `ant.sh`, `ws_ant.sh`

  – Windows `ant.bat`, `ws_ant.bat`

  (UNIX) ou (Windows).

- **Mettre à jour manuellement le descripteur Web**

  Editez le fichier `web.xml` intégré à l'application Web pour incorporer la déclaration de filtre, son mappage de servlet et les paramètres d'initialisation du contexte de servlet. N'utilisez pas cette méthode car elle est source d'erreurs possibles.

Pour une liste des paramètres que vous pouvez utiliser, voir «Paramètres d'initialisation du contexte de servlet», à la page 321.

2. Déployez l'application. Déployez l' application à l'aide de votre procédure normale pour un serveur ou un cluster. Une fois que vous avez déployé l'application, vous pouvez la démarrer.
3. Accédez à l'application. Vous pouvez maintenant accéder à l'application, qui interagit avec le gestionnaire de sessions et WebSphere eXtreme Scale.

Que faire ensuite

Vous pouvez modifier la majorité des attributs de configuration du gestionnaire de sessions lorsque vous instrumentez votre application pour utiliser le gestionnaire de sessions. Ces attributs sont : réplication synchrone ou asynchrone, taille de la table de session en mémoire, etc. En dehors des attributs modifiables lors de l'instrumentation de l'application, les seuls autres attributs de configuration que vous pouvez modifier après le déploiement de l'application sont ceux liés à la topologie des clusters de serveurs WebSphere eXtreme Scale et à la manière dont leurs clients (gestionnaires de sessions) s'y connectent.

Comportement de scénarios distant : si la grille de données complète qui héberge les données de sessions d'application est inaccessible à partir du client du conteneur Web, le client utilise le conteneur Web de base dans WebSphere Application Server pour la gestion des sessions. La grille de données peut être inaccessible dans les scénarios suivants :

- Problème de réseau entre le conteneur Web et les serveurs de conteneur distants.
- Arrêt des processus serveur de conteneur distant

Le nombre de références de session conservées en mémoire, spécifié par le paramètre **sessionTableSize** , est toujours maintenu lorsque les sessions sont stockées dans le conteneur Web de base. Les sessions les moins utilisées sont invalidés à partir du cache de session du conteneur Web lorsque la valeur **sessionTableSize** est dépassée. Si la grille de données distante devient disponible, les sessions ayant été invalidées à partir du cache du conteneur Web peuvent extraire les données de la grille de données distante et charger les données dans une nouvelle session. Si l'ensemble de la grille de données distante n'est pas disponible et que la session est invalidée dans le cache de session, les données de session utilisateur sont perdues. Compte tenu de ce problème, n'arrêtez pas l'ensemble de la grille de données distante de production lorsque le système est chargé.

Fractionnement automatique des applications pour la gestion de session HTTP dans WebSphere Application Server :

Vous pouvez configurer votre application WebSphere Application Server pour qu'elle conserve les sessions vers une grille de données. Cette grille de données peut être un serveur conteneur intégré qui s'exécute au sein de WebSphere Application Server. Il peut s'agir également d'une grille de données distantes.

Avant de commencer

Pour pouvoir changer la configuration dans WebSphere Application Server, vous devez avoir :

- le nom de la grille de données de session que vous voulez utiliser. Voir «Configuration du gestionnaire de sessions HTTP avec WebSphere Application Server», à la page 301 pour savoir comment créer une grille de données de session
- si le service de catalogue que vous voulez utiliser pour gérer vos sessions se trouve hors de la cellule dans laquelle vous installez votre application de sessions, vous devez créer un domaine de services de catalogue. Pour plus d'informations, voir «Création de domaines de services de catalogue dans WebSphere Application Server», à la page 258.
- Si vous configurez un domaine de services de catalogue, il peut être nécessaire d'activer la sécurité du client sur le domaine de services de catalogue si les serveurs de conteneurs requièrent une authentification. Ces paramètres indiquent à l'environnement d'exécution l'implémentation à utiliser CredentialGenerator. Cette implémentation génère des données d'identification à envoyer à la grille de données distante. Pour plus d'informations sur la configuration de ces paramètres, voir «Configuration de la sécurité client dans un domaine de services de catalogue», à la page 528.
- activé la sécurité global dans la console d'administration WebSphere Application Server si vous voulez prendre en charge l'un des scénarios suivants :
  - Activer SSL (Secure Socket Layer) pour les serveurs de catalogue dans le domaine de services de catalogue.

– Utiliser SSL pour le domaine de services de catalogue avec SSL pris en charge.

Vous spécifiez l'utilisation de SSL pour un serveur de catalogue en affectant à l'attribut **transportType** la valeur SSL-Required dans Fichier de propriétés du serveur. Pour plus d'informations sur la configuration de la sécurité globale, voir Paramètres de sécurité globale.

- Si vous utilisez la version 7.1.0.3 ou une version suivante, vous pouvez rendre persistantes les sessions qui utilisent la réécriture d'URL ou les cookies comme suivi de session dans la grille de données. Pour les versions antérieures à la version 7.1.0.3, vous ne pouvez pas rendre persistantes les sessions utilisant la réécriture d'URL comme mécanisme de suivi de session. Pour activer la persistance des sessions qui utilisent la réécriture d'URL, affectez à la propriété **useURLEncoding** la valeur true dans le fichier splicer.properties.
- 7.1.1+ Lorsque vous raccordez automatiquement des applications pour la gestion de session HTTP dans WebSphere Application Server, tous les serveurs d'applications qui hébergent l'application Web ont la propriété personnalisée de conteneur Web **HttpSessionIdReuse** affectée de la valeur true. Cette propriété permet aux sessions qui ont basculé d'un serveur d'applications vers un autre ou qui ont été invalidées depuis le cache de session interne dans un scénario distant, de conserver leur ID de session dans les demandes. Si vous ne voulez pas conserver ce comportement, affectez à la propriété personnalisée de conteneur Web la valeur false sur tous les serveurs d'applications applicables avant de configurer la gestion des sessions des applications. Pour plus d'informations sur cette propriété personnalisée, voir «Traitement des problèmes d'intégration du cache», à la page 545.

Procédure

- **Pour configurer la gestion de session lors de l'installation de l'application, effectuez la procédure suivante :**
  1. Dans la console d'administration de WebSphere Application Server, cliquez sur **Applications** > **Nouvelle application** > **Nouvelle application d'entreprise**. Sélectionnez le chemin **Détaillé** pour la création de l'application, puis effectuez les premières étapes de l'assistant.
  2. A l'étape **Paramètres de gestion des sessions eXtreme Scale session** de l'assistant, configurez la grille de données que vous voulez utiliser. Vous avez le choix entre deux types de grille : **Grille de données distante eXtreme Scale** ou **Grille de données imbriquée eXtreme Scale**.
     – Pour l'option **Grille de données distante eXtreme Scale**, choisissez le domaine de services de catalogue qui gère la grille de données des sessions et choisissez une grille de données dans la liste des grilles de données de sessions actives.
     – Pour l'option **Grille de données imbriquée eXtreme Scale**, vous pouvez soit chosir la configuration ObjectGrid par défaut, soit spécifier l'emplacement des fichiers de configuration ObjectGrid.
  3. Terminez l'installation de l'application en effectuant la procédure de l'assistant.

  Vous pouvez également installer l'application à l'aide d'un script wsadmin. Dans l'exemple suivant, le paramètre**-SessionManagement** crée une configuration identique à celle que vous pouvez créer dans la console d'administration :

  Pour la configuration d'une grille de données eXtreme Scale distante :

```
AdminApp.install('C:/A.ear', '[  -nopreCompileJSPs -distributeApp
-nouseMetaDataFromBinary -nodeployejb -appname A -edition 8.0
-createMBeansForResources -noreloadEnabled -nodeployws -validateinstall
```

```
off -noprocessEmbeddedConfig -filepermission .*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755
-buildVersion Unknown -noallowDispatchRemoteInclude -noallowServiceRemoteInclude
-asyncRequestDispatchType DISABLED -nouseAutoLink -SessionManagement [[true
XSRemoteSessionManagement cs0:!:grid0]]
-MapWebModToVH [[MicroWebApp microwebapp.war,WEB-INF/web.xml default_host] [MicroSipApp
microsipapp.war,WEB-INF/web.xml default_host] [MicroDG1App microdg1app.war,WEB-INF/web.xml
default_host] [MicroDG2App microdg2app.war,WEB-INF/web.xml default_host] [MicroSip2App
microsip2app.war,WEB-INF/web.xml default_host]]]')
```

Pour une grille eXtreme Scale imbriquée, avec la configuration par défaut :

```
AdminApp.install('C:/A.ear', '[  -nopreCompileJSPs -distributeApp
-nouseMetaDataFromBinary -nodeployejb -appname A -edition 8.0
-createMBeansForResources -noreloadEnabled -nodeployws -validateinstall
off -noprocessEmbeddedConfig -filepermission .*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755
-buildVersion Unknown -noallowDispatchRemoteInclude -noallowServiceRemoteInclude
-asyncRequestDispatchType DISABLED -nouseAutoLink -SessionManagement [[true
XSRemoteSessionManagement  :!: :!:default]]  -MapWebModToVH [[MicroWebApp microwebapp.war,
WEB-INF/web.xml default_host] [MicroSipApp
microsipapp.war,WEB-INF/web.xml default_host] [MicroDG1App microdg1app.war,WEB-INF/web.xml
default_host] [MicroDG2App microdg2app.war,WEB-INF/web.xml default_host] [MicroSip2App
microsip2app.war,WEB-INF/web.xml default_host]]]')
```

Pour une grille eXtreme Scale imbriquée, avec une configuration personnalisée :

```
AdminApp.install('C:/A.ear', '[  -nopreCompileJSPs -distributeApp
-nouseMetaDataFromBinary -nodeployejb -appname A -edition 8.0
-createMBeansForResources -noreloadEnabled -nodeployws -validateinstall
off -noprocessEmbeddedConfig -filepermission .*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755
-buildVersion Unknown -noallowDispatchRemoteInclude -noallowServiceRemoteInclude
-asyncRequestDispatchType DISABLED -nouseAutoLink -SessionManagement [[true
XSRemoteSessionManagement  :!: :!:custom:!:c:\XS\objectgrid.xml:!:c:\XS\objectgriddeployment.xml]]
-MapWebModToVH [[MicroWebApp microwebapp.war,WEB-INF/web.xml default_host] [MicroSipApp
microsipapp.war,WEB-INF/web.xml default_host] [MicroDG1App microdg1app.war,WEB-INF/web.xml
default_host] [MicroDG2App microdg2app.war,WEB-INF/web.xml default_host] [MicroSip2App
microsip2app.war,WEB-INF/web.xml default_host]]]')
```

- **Pour configurer la gestion de session sur une application existante dans la console d'administration de WebSphere Application Server :**
  1. Dans la console d'administration WebSphere Application Server, cliquez sur **Applications** > **Types d'application** > **Applications d'entreprise WebSphere** > *application_name* > **Propriétés du module Web** > **Gestion des sessions** > **Paramètres de gestion des sessions eXtreme Scale**.
  2. Mettez à jour les zones pour activer la persistance de session dans la grille de données.

  Vous pouvez également mettre à jour l'application à l'aide d'un script wsadmin. Dans l'exemple suivant, le paramètre**-SessionManagement** crée une configuration identique à celle que vous pouvez créer dans la console d'administration :

  Pour la configuration de la grille de données eXtreme Scale distante :

  ```
  AdminApp.edit('DefaultApplication','[-SessionManagement[[true
  XSRemoteSessionManagement cs0:!:grid0]]]')
  ```

  Les caractères **:!:** envoyés sont utilisés comme délimiteurs. Les valeurs envoyées sont les suivantes :

  catalogServiceName:!:*gridName*

  Pour le scénario intégré eXtreme Scale avec la configuration par défaut :

  ```
  AdminApp.edit('DefaultApplication','[-SessionManagement[[true
  XSEmbeddedSessionManagement :!:!:!:default]]]')
  ```

Les caractères **:!:** envoyés sont utilisés comme délimiteurs. Les valeurs envoyées sont les suivantes :

catalogServiceName:!:*gridName*:!:default:!:
absolutePath_to_objectGridXmlfile:!:*absolutePath_to_DeploymentXmlfile*

Pour le scénario intégré eXtreme Scale avec la configuration personnalisée :

```
AdminApp.edit('DefaultApplication','[-SessionManagement[[true
XSEmbeddedSessionManagement
:!:!:!:custom:!:c:\XS\objectgrid.xml:!:c:\XS\objectgriddeployment.xml]]]')
```

Les caractères **:!:** envoyés sont utilisés comme délimiteurs. Les valeurs envoyées sont les suivantes :

catalogServiceName:!:*gridName*:!:custom:!:
absolutePath_to_objectGridXmlfile:!:*absolutePath_to_DeploymentXmlfile*

Lorsque vous enregistrez les modifications, l'application utilise la grille de données configurée pour la persistance des sessions sur le dispositif.

- **Pour configurer la gestion de session sur un serveur existant :**
  1. Dans la console d'administration WebSphere Application Server, cliquez sur **Serveurs** > **types de serveur** > **Serveurs d'applications WebSphere** > *server_name* > **Gestion des sessions** > **Paramètres de gestion des sessions eXtreme Scale**.
  2. Actualisez les champs pour activer la persistance des sessions.

  Les commandes wsadmin suivantes vous permettent de configurer également la gestion des sessions sur un serveur existant :

  Pour la configuration d'une grille de données eXtreme Scale distante :

```
AdminTask.configureServerSessionManagement('[-nodeName IBM-C77EE220EB6Node01 -serverName server1
-enableSessionManagement true -sessionManagementType XSRemoteSessionManagement -XSRemoteSessionManagement
[-catalogService cs0 -csGridName grid0]]')
```

  Pour une configuration de grille eXtreme Scale imbriquée :

  – La configuration par défaut, si vous utilisez les fichiers XML par défaut :

```
AdminTask.configureServerSessionManagement('[-nodeName IBM-C77EE220EB6Node01 -serverName server1
-enableSessionManagement true -sessionManagementType XSEmbeddedSessionManagement
-XSEmbeddedSessionManagement [-embeddedGridType default -objectGridXML -objectGridDeploymentXML ]]')
```

  – La configuration personnalisée, si vous utilisez des fichiers XML personnalisés :

```
AdminTask.configureServerSessionManagement('[-nodeName IBM-C77EE220EB6Node01 -serverName server1
-enableSessionManagement true -sessionManagementType XSEmbeddedSessionManagement
-XSEmbeddedSessionManagement
[-embeddedGridType custom -objectGridXML c:\XS\objectgrid.xml -objectGridDeploymentXML
c:\XS\objectgriddeployment.xml]]')
```

  Lorsque vous enregistrez les modifications, le serveur utilise la grille de données configurée pour la persistance de sessions avec toutes les applications qu'il exécute.

- Si vous souhaitez modifier d'autres aspects de la configuration des sessions HTTP, vous pouvez éditer le fichier `splicer.properties`. Vous pouvez obtenir l'emplacement du chemin du fichier `splicer.properties` en recherchant la propriété personnalisée **`sessionFilterProps`**. Si vous avez configuré la persistance de session au niveau du serveur, le nom de la propriété personnalisée est `com.ibm.websphere.xs.sessionFilterProps`. Si vous l'avez configurée au niveau de l'application, elle s'appelle `<application_name>,com.ibm.websphere.xs.sessionFilterProps`. Ces propriétés personnalisées peuvent se trouver dans l'un des emplacements suivants :
  – Dans un environnement WebSphere Application Server Network Deployment, le fichier `splicer.properties` se trouve dans le chemin du profil du gestionnaire de déploiement.

– Dans un environnement WebSphere Application Server autonome : une propriété personnalisée sur le serveur d'applications

Vous pouvez ouvrir le fichier indiqué, effectuez les modifications et synchroniser les noeuds pour propager le fichier mis à jour des propriétés vers les autres noeuds de la configuration. Tous les noeuds de serveur d'applications nécessitent que le fichier `splicer.properties` se trouve dans le chemin défini pour que les sessions persistent.

Avertissement : Si vous souhaitez activer la persistance pour les sessions qui utilisent la réécriture d'URL, affectez à la propriété **`useURLEncoding`** la valeur `true` dans le fichier `splicer.properties`.

Pour plus d'informations sur les propriétés dans le fichier `splicer.properties`, voir «Fichier `splicer.properties`», à la page 324.

Résultats

Vous avez configuré le gestionnaire de sessions HTTP pour que les sessions soient conservées vers une grille de données. Les entrées sont supprimées de la grille de données lorsque les sessions expirent. Voir Paramètres de gestion des sessions pour plus d'informations sur la mise à jour la valeur de temporisation des sessions dans la console d'administration WebSphere Application Server.

Paramètres de gestion des sessions eXtreme Scale :

Vous pouvez configurer les applications WebSphere Application Server pour utiliser WebSphere eXtreme Scale ou un WebSphere DataPower XC10 Appliance pour la persistance de session.

Vous pouvez éditer ces paramètres dans l'assistant d'installation des applications d'entreprise ou dans les pages d'information de l'application ou du serveur :

- Version 6.1 : **Applications** > **Installer une nouvelle application**
- Version 6.1 : **Applications** > **Applications d'enterprise** > *application_name*
- Version 6.1 : **Serveurs** > **Serveurs d'applications** > *server_name* > **Paramètres de conteneur Web** > **Gestion de session**
- Version 7.0 : **Applications** > **Nouvelle application** > **Nouvelle applications d'enterprise** et choisissez le chemin détaillé pour créer l'application.
- Version 7.0 : **Applications** > **Types d'application** > **Applications d'entreprise WebSphere** > *application_name* > **Propriétés du module Web** > **Gestion de sessionmanagement** > **Paramètres de gestion de session**
- Version 7.0 : **Serveurs** > **Types de serveur** > **Serveurs d'applications WebSphere** > *server_name* > **Paramètres de conteneur** > **Paramètres de gestion de session**

Activer la gestion des sessions :

Permet à la gestion des sessions d'utiliser WebSphere eXtreme Scale imbriquée ou une grille de données distante ou un WebSphere DataPower XC10 Appliance pour la persistance de session.

Gestion de la persistance des sessions par :

Indique comment la persistance des sessions est gérée. Vous pouvez sélectionner l'une des options suivantes :

- WebSphere DataPower XC10 Appliance

- Grille de données distante eXtreme Scale
- Grille de données imbriquée eXtreme Scale

Les autres paramètres que vous configurez varient en fonction du mécanisme de persistance des sessions sélectionné.

Paramètres WebSphere DataPower XC10 Appliance :

Les paramètres suivants sont spécifiques de la configuration de WebSphere DataPower XC10 Appliance pour la persistance de session.

Adresse IP ou nom d'hôte de WebSphere DataPower XC10 Appliance :

Indique l'adresse IP ou le nom d'hôte de l'appliance à utiliser pour stocker les données des sessions.

Information d'identification administrative IBM WebSphere DataPower XC10 Appliance :

Indiquez le **nom d'utilisateur** et le **mot de passe** que vous utilisez pour vous connecter à l'interface utilisateur de DataPower XC10 Appliance. Cliquez sur **Tester la connexion...** pour tester la connexion à l'appliance.

Préférences de persistance des sessions :

Indique la grille de données dans laquelle les sessions sont conservées. Vous pouvez sélectionner l'une des options suivantes :

- **Conserver les sessions dans une nouvelle grille de données sur IBM WebSphere DataPower XC10 Appliance**. Vous pouvez ensuite indiquer un **nom de grille de données**.
- **Conserver les sessions dans une grille de données existante sur IBM WebSphere DataPower XC10 Appliance**. Vous pouvez ensuite entrer ou rechercher un **nom de grille de données existant**.

Configuration d'une grille de données distante eXtreme Scale :

Les paramètres suivants s'appliquent à la configuration de la grille distante eXtreme Scale pour la persistance des sessions :

domaine de services de catalogue gérant la grille de données de session distante :

Indique le domaine de services de catalogue à utiliser pour gérer les sessions.

Si aucun domaine de services de catalogue n'est affiché ou que vous souhaitez créer un domaine de services de catalogue nouveau, cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **domaine de services de catalogue**.

Grille de données distante utilisée pour stocker les informations de session :

Indique le nom de la grille de données du domaine de services de catalogue où les informations de session doivent être stockées. La liste des grilles distantes actives est renseignée lorsque vous sélectionnez un service de catalogue. La grille de données distante doit déjà être définie dans la configuration eXtreme Scale.

Configuration d'une grille de données imbriquée eXtreme Scale :

Les paramètres suivants s'appliquent à une configuration eXtreme Scale imbriquée. Dans le scénario de configuration imbriquée eXtreme Scale, les processus eXtreme Scale sont hébergés par des processus WebSphere Application Server.

Configuration d'une grille de données imbriquée eXtreme Scale :

- **Utiliser la configuration ObjectGrid par défaut**
- **Indiquer des fichiers de configuration ObjectGrid personnalisés**

  Chemin complet du fichier `objectgrid.xml` à copier dans la configuration
  Indique le chemin complet du fichier `objectgrid.xml` correspondant à la configuration à utiliser.

  Chemin complet du fichier `objectgriddeployment.xml` à copier dans la configuration
  Indique le chemin complet du fichier `objectgriddeployment.xml` correspondant à la configuration à utiliser.

Utilisation de WebSphere eXtreme Scale pour la gestion des sessions SIP

Vous pouvez utiliser WebSphere eXtreme Scale comme mécanisme de réplication SIP (Session Initiation Protocol). Ce système se substitue en toute fiabilité au service DRS (Data Replication Service) pour la réplication des sessions SIP.

Configuration de la gestion des sessions SIP

Pour utiliser WebSphere eXtreme Scale comme mécanisme de réplication SIP, définissez la propriété personnalisée com.ibm.sip.ha.replicator.type. Dans la console d'administration, sélectionnez **Application servers** > ***mon_serveur_applications*** > **SIP container** > **Custom properties** pour chaque serveur auquel la propriété personnalisée doit être ajoutée. Entrez `com.ibm.sip.ha.replicator.type` pour le nom et `OBJECTGRID` pour la valeur.

Utilisez les propriétés ci-dessous pour personnaliser le comportement de la valeur ObjectGrid utilisée pour stocker les sessions SIP. Dans la console d'administration, cliquez sur **Application servers** > ***mon_serveur_applications*** > **SIP container** > **Custom properties** pour chaque serveur auquel la propriété personnalisée doit être ajoutée. Renseignez les champs **Nom** et **Valeur**. Les mêmes propriétés doivent être définies pour chaque serveur pour qu'il fonctionne correctement.

Tableau 23. Propriétés personnalisées pour la gestion des sessions SIP avec ObjectGrid

Propriété	Valeur	Par défaut
com.ibm.sip.ha.replicator.type	`OBJECTGRID` : utilisez ObjectGrid pour stocker les sessions SIP	
min.synchronous.replicas	Nombre minimal de fragments réplique synchrones	0
max.synchronous.replicas	Nombre maximal de fragments réplique synchrones	0
max.asynchronous.replicas	Nombre maximal de fragments réplique asynchrones	1
auto.replace.lost.shards	Pour plus d'informations, voir «Configuration de déploiements répartis», à la page 236.	true
development.mode	• `true` : active les fragments réplique sur le même noeud que les primaires • `false` : les fragments réplique doivent être sur un noeud différent que les primaires	false

Configuration du gestionnaire de sessions HTTP avec WebSphere Portal

Vous pouvez rendre persistantes des sessions HTTP depuis WebSphere Portal dans une grille de données.

Avant de commencer

Votre environnement WebSphere eXtreme Scale et WebSphere Portal doivent satisfaire aux spécifications suivantes :

- La façon dont vous installez WebSphere eXtreme Scale dépend de votre scénario de déploiement. Vous pouvez exécuter les serveurs de conteneurs qui hébergent les grilles de données en dedans ou en dehors de la cellule WebSphere Application Server :
  - Si vous exécutez des serveurs de conteneurs dans la cellule WebSphere Application Server **(scénario imbriqué)** : installez le client et le serveur WebSphere eXtreme Scale sur vos noeuds WebSphere Application Server et WebSphere Portal.
  - Si vous exécutez des serveurs de conteneurs en dehors de la cellule WebSphere Application Server **(scénario à distance)** : installez WebSphere eXtreme Scale Client sur vos noeuds WebSphere Application Server et WebSphere Portal.

  Pour plus d'informations, voir «Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client avec WebSphere Application Server», à la page 163.
- WebSphere Portal Version 7 ou suivante.
- Les portlets personnalisés doivent être configurés dans WebSphere Portal. Les portlets d'administration livrés avec WebSphere Portal ne peuvent actuellement pas être intégrés avec des grilles de données.

Pourquoi et quand exécuter cette tâche

L'introduction de WebSphere eXtreme Scale dans un environnement WebSphere Portal peut être bénéfique dans les scénarios suivants :

Important : Bien que les scénarios suivants apportent des avantages, une utilisation plus importante des processeurs au niveau de WebSphere peut résulter de l'introduction de WebSphere eXtreme Scale dans l'environnement.

- **Lorsque la persistance des sessions est requise.**

  Par exemple, si les données de session de vos portlets personnalisés doivent rester disponibles lors d'une défaillance de WebSphere Portal Server, vous pouvez rendre persistantes les sessions HTTP dans la grille de données WebSphere eXtreme Scale. Les données sont répliquées entre de nombreux serveurs, accroissant la disponibilité des données.
- **Dans une topologie avec plusieurs centres de données.**

  Si votre topologie couvre plusieurs centres de données à travers différents emplacements physiques, vous pouvez rendre persistantes les sessions HTTP de WebSphere Portal dans la grille de données WebSphere eXtreme Scale. Les sessions sont répliquées dans les grilles de données des centres de données. Si un centre de données est défaillant, les sessions sont basculées vers un autre centre de données qui a une copie des données de la grille de données.
- **Pour diminuer la mémoire requise au niveau de WebSphere Portal Server.**

En déchargeant les données de session sur un groupe de serveurs de conteneurs, un sous-ensemble des sessions se trouve sur les serveurs WebSphere Portal. Ce déchargement de données réduit la mémoire requise au niveau de WebSphere Portal Server.

Procédure

1. Raccordez l'application WebSphere Portal `wps` et les éventuels portlets personnalisés pour permettre aux sessions d'être stockées dans la grille de données.

   Vous pouvez raccorder l'application en configurant la gestion de session HTTP lorsque vous déployez l'application, ou vous pouvez utiliser des propriétés personnalisées pour raccorder automatiquement vos applications. Voir «Configuration du gestionnaire de sessions HTTP avec WebSphere Application Server», à la page 301 pour plus d'informations sur le raccordement de l'application.
2. Si vous utilisez un scénario à distance où vos serveurs de conteneurs se trouvent en dehors de WebSphere Application Server, démarrez explicitement les conteneurs eXtreme Scale distants pour les scénarios de persistance de sessions HTTP à distance. Démarrez les conteneurs avec les fichiers de configuration `XS/ObjectGrid/session/samples/objectGridStandAlone.xml` et `objectGridDeploymentStandAlone.xml`. Par exemple, vous pouvez utiliser la commande suivante :

   ```
   startOgServer.sh xsContainer1 -catalogServiceEndPoints <hôte>:<port>
   -objectgridFile XS/ObjectGrid/session/samples/objectGridStandAlone.xml -deploymentPolicyFile
   XS/ObjectGrid/session/samples/objectGridDeploymentStandAlone.xml
   ```

   Pour plus d'informations sur le démarrage des serveurs de conteneurs, voir «Démarrage des serveurs de conteneur», à la page 398. Si vous utilisez un scénario imbriqué, consultez «Configuration des serveurs de conteneurs dans WebSphere Application Server», à la page 275 pour plus d'informations sur la configuration et le démarrage des serveurs de conteneurs.
3. Redémarrez les serveurs WebSphere Portal. Pour plus d'informations, voir WebSphere Portal version 7 : Démarrage et arrêt des serveurs, des gestionnaires de déploiement et des agents de noeud.

Résultats

Vous pouvez accéder à WebSphere Portal Server ; les données de session HTTP pour les portlets personnalisés configurés sont conservées dans la grille de données.

Si l'ensemble de la grille de données qui héberge les données de sessions d'application est inaccessible à partir du client de conteneur Web, le client utilise le conteneur Web de base dans la gestion de sessions WebSphere Application Server. La grille de données peut être inaccessible dans les scénarios suivants :

- Problème de réseau entre le conteneur Web et les serveurs de conteneur distants.
- Arrêt des processus serveur de conteneur distant.

Le nombre de références de session conservées en mémoire, spécifié par le paramètre **sessionTableSize** est toujours maintenu lorsque les sessions sont stockées dans le conteneur Web de base. Les sessions les moins utilisées sont invalidés à partir du cache de session du conteneur Web lorsque la valeur **sessionTableSize** est dépassée. Si la grille de données distante devient disponible, les sessions ayant été invalidées à partir du cache du conteneur Web peuvent extraire les données de la grille de données distante et charger les données dans une nouvelle session. Si l'ensemble de la grille de données distante n'est pas

disponible et que la session est invalidée dans le cache de session, les données de session utilisateur sont perdues. Compte tenu de ce problème, n'arrêtez pas l'ensemble de la grille de données distante de production lorsque le système est chargé.

Configuration du gestionnaire de sessions HTTP pour divers serveurs d'applications

WebSphere eXtreme Scale est regroupé avec une implémentation de gestion de session qui remplace le gestionnaire de sessions par défaut pour un conteneur Web. Cette implémentation fournit l réplication de session, la haute disponibilité, améliore l'évolutivité et des options de configuration. Vous pouvez activer le gestionnaire de réplication de session WebSphere eXtreme Scale et le démarrage du conteneur intégré ObjectGrid générique.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser le gestionnaire de sessions HTTP avec d'autres serveurs d'applications qui n'exécutent pas WebSphere Application Server, WebSphere Application Server Community Edition, par exemple. Pour configurer d'autres serveurs d'applications pour qu'ils utilisent la grille de données, vous devez raccorder votre application et incorporer des fichiers WebSphere eXtreme Scale Java archive (JAR) dans votre application.

Procédure

1. Raccordez votre application de sorte qu'elle puisse utiliser le gestionnaire de sessions. Pour utiliser le gestionnaire de sessions, vous devez ajouter les déclarations de filtre appropriées aux descripteurs de déploiement Web de l'application. En outre, les paramètres de configuration du gestionnaire de sessions sont transmis au gestionnaire de sessions sous la forme de paramètres d'initialisation du contexte de servlet dans les descripteurs de déploiement. Vous disposez de trois manières de présenter ces informations dans votre application :
   - Script **addObjectGridFilter** :

     Utilisez un script de ligne de commande fourni avec eXtreme Scale pour raccorder une application avec des déclarations de filtre et une configuration sous forme de paramètres d'initialisation de contexte de servlet. Le script *rép_base_wxs*/session/bin/addObjectGridFilter.sh|bat accepte deux paramètres : le chemin absolu d'accès au fichier EAR (enterprise archive) ou au fichier WAR (web archive) à raccorder et le chemin absolu au fichier des propriétés splicer qui contient diverses propriétés de configuration. La syntaxe de ce script est la suivante :

     Windows

     ```
     addObjectGridFilter.bat <ear_or_war_file> <splicer_properties_file>
     ```

     UNIX

     ```
     addObjectGridFilter.sh <ear_or_war_file> <splicer_properties_file>
     ```

     UNIX Exemple d'utilisation de eXtreme Scale installé dans un répertoire autonome sur UNIX :

     a. `cd rép_base_wxs/session/bin`

     b. `addObjectGridFilter.sh /tmp/mySessionTest.ear rép_base_wxs/session/samples/splicer.properties`

     Le filtre de servlet qui est joint conserve les valeurs de configuration par défaut. Vous pouvez remplacer ces valeurs par défaut par des options de

configuration que vous spécifiez dans le fichier de propriétés, dans le second argument. Pour une liste des paramètres que vous pouvez utiliser, voir «Paramètres d'initialisation du contexte de servlet», à la page 321.

Vous pouvez modifier et utiliser l'exemple de fichier `splicer.properties` fourni avec l'installation d'eXtreme Scale. Vous pouvez également utiliser le script **`addObjectGridServlets`**, qui insère le gestionnaire de sessions en étendant chaque servlet. Mais le script recommandé est le script **`addObjectGridFilter`**.

- Script de génération Ant :

  WebSphere eXtreme Scale est fourni avec un fichier `build.xml` qui peut être utilisé par Apache Ant, qui est inclus dans le dossier *racine_was*`/bin` d'une installation WebSphere Application Server. Vous pouvez modifier le fichier `build.xml` pour changer les propriétés de configuration du gestionnaire de sessions. Les propriétés de configuration sont identiques aux noms de propriété dans le fichier `splicer.properties`. Une fois que le fichier `build.xml` a été modifié, appelez le processus Ant en exécutant `ant.sh`, `ws_ant.sh` (UNIX) ou `ant.bat`, `ws_ant.bat` (Windows).

- Mise à jour manuelle du descripteur Web :

  Editez le fichier `web.xml` qui est packagé avec l'application Web pour incorporer la déclaration de filtre, son mappage de servlets et les paramètres d'initialisation du contexte de servlet. N'utilisez pas cette méthode car elle est source d'erreurs possibles.

Pour une liste des paramètres que vous pouvez utiliser, voir «Paramètres d'initialisation du contexte de servlet», à la page 321.

2. Incorporez dans votre application les fichiers JAR du gestionnaire de réplication de sessions d'WebSphere eXtreme Scale. Vous pouvez incorporer les fichiers dans le répertoire `WEB-INF/lib` des modules d'application ou dans le chemin d'accès aux classes du serveur d'applications. Les fichiers JAR requis varient selon le type de conteneurs utilisés :
   - Serveurs de conteneur distants : `ogclient.jar` et `sessionobjectgrid.jar`
   - Serveurs de conteneur intégrés : `objectgrid.jar` et `sessionobjectgrid.jar`
3. Facultatif : Si vous utilisez des serveurs de conteneur distant, démarrez les serveurs de conteneur. Pour plus de détails, reportez-vous à la rubrique «Démarrage des serveurs de conteneur», à la page 398.
4. Déployez l'application. Déployez l' application à l'aide de votre procédure normale pour un serveur ou un cluster. Une fois que vous avez déployé l'application, vous pouvez la démarrer.
5. Accédez à l'application. Vous pouvez maintenant accéder à l'application, qui interagit avec le gestionnaire de sessions et WebSphere eXtreme Scale.

Que faire ensuite

Vous pouvez modifier la majorité des attributs de configuration du gestionnaire de sessions lorsque vous instrumentez votre application pour utiliser le gestionnaire de sessions. Ces attributs sont des variantes du type de réplication (synchrone ou asynchrone), la taille de la table des sessions en mémoire, etc. En dehors des attributs modifiables lors de l'instrumentation de l'application, les seuls autres attributs de configuration que vous pouvez modifier après le déploiement de l'application sont ceux liés à la topologie des clusters de serveurs WebSphere eXtreme Scale et à la manière dont leurs clients (gestionnaires de sessions) s'y connectent.

Comportement dans le scénario distant : si l'ensemble de la grille de données qui héberge les données de session d'application est inaccessible depuis le client du conteneur Web, le client utilise à la place le conteneur Web de base du serveur d'applications pour gérer les sessions. La grille de données peut être inaccessible dans les scénarios suivants :

- Problème de réseau entre le conteneur Web et les serveurs de conteneur distants
- Arrêt des processus serveur de conteneur distant

Le nombre de références de session conservées en mémoire, spécifié par le paramètre **sessionTableSize**, est toujours maintenu lorsque les sessions sont stockées dans le conteneur Web de base. Les sessions les moins utilisées sont invalidées à partir du cache de session du conteneur Web lorsque la valeur **sessionTableSize** est dépassée. Si la grille de données distante devient disponible, les sessions ayant été invalidées à partir du cache du conteneur Web peuvent extraire les données de la grille de données distante et charger les données dans une nouvelle session. Si l'ensemble de la grille de données distante n'est pas disponible et que la session est invalidée dans le cache de session, les données de session utilisateur sont perdues. Compte tenu de ce problème, n'arrêtez pas l'ensemble de la grille de données distante de production lorsque le système est chargé.

Fichiers XML de configuration du gestionnaire de sessions HTTP

Lorsque vous démarrez un serveur de conteneur qui stocke les données de session HTTP, vous pouvez utiliser les fichiers XML par défaut ou spécifier des fichiers XML personnalisés. Ces fichiers créent des noms ObjectGrid spécifiques, un nombre de répliques, etc.

Emplacement des fichiers d'exemples

Ces fichiers XML sont regroupés dans *racine_install_wxs*`/ObjectGrid/session/samples` pour une installation autonome ou *racine_was*`/optionalLibraries/ObjectGrid/session/samples` pour WebSphere eXtreme Scale installé dans une cellule WebSphere Application Server.

Package XML intégré

Si vous configurez un scénario intégré, le serveur de conteneur commence dans le groupe de serveurs de Web. Utilisez le fichier `objectGrid.xml` et le fichier `objectGridDeployment.xml` fournis par défaut. Vous pouvez mettre à jour ces fichiers pour personnaliser le comportement du gestionnaire de session HTTP.

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd" xmlns="http://ibm.com/ws/objectgrid/config">
 <objectGrids>
  <objectGrid name="session" txTimeout="30">
   <bean id="ObjectGridEventListener" className="com.ibm.ws.xs.sessionmanager.SessionHandleManager"/>
   <backingMap name="objectgridSessionMetadata" pluginCollectionRef="objectgridSessionMetadata" readOnly="false"
     lockStrategy="PESSIMISTIC" ttlEvictorType="LAST_ACCESS_TIME" timeToLive="3600" copyMode="NO_COPY"/>
   <backingMap name="objectgridSessionAttribute.*" template="true" readOnly="false" lockStrategy="PESSIMISTIC"
     ttlEvictorType="NONE" copyMode="NO_COPY"/>
       <backingMap name="objectgridSessionTTL.*" template="true" readOnly="false" lockStrategy="PESSIMISTIC"
     ttlEvictorType="LAST_ACCESS_TIME" timeToLive="3600" copyMode="NO_COPY"/>
  </objectGrid>
 </objectGrids>
    <backingMapPluginCollections>
        <backingMapPluginCollection id="objectgridSessionMetadata">
            <bean id="MapEventListener" className="com.ibm.ws.xs.sessionmanager.MetadataMapListener"/>
        </backingMapPluginCollection>
    </backingMapPluginCollections>
</objectGridConfig>
```

Figure 35. Fichier `objectGrid.xml`

Valeurs que vous pouvez modifier :

Attribut de nom ObjectGrid
La valeur doit correspondre aux valeurs suivantes dans les autres fichiers de configuration :

- La propriété **`objectGridName`** dans le fichier `splicer.properties` qui est utilisé pour raccorder l'application Web.
- L'attribut **`objectgridName`** dans le fichier `objectGridDeployment.xml`.

Si vous disposez de plusieurs applications et voulez stocker les données de session dans des grilles de données différentes, ces applications doivent avoir des valeurs d'attribut de nom ObjectGrid différentes.

7.1.1+ **Attribut ObjectGrid txTimeout**
Cette valeur détermine le nombre de secondes pendant lequel une transaction peut être ouverte avant que le serveur de conteneur déclenche la fasse expirer. La valeur par défaut est 30 secondes et elle peut être changée en fonction de l'environnement. Si la persistance de session HTTP est configurée avec une valeur de paramètre d'initialisation de contexte de servlet **`replicationInterval`** supérieure à zéro, les transactions sont traitées par lots dans une unité d'exécution. Si la propriété **`replicationInterval`** a la valeur `0`, une transaction démarre généralement lorsqu'une application Web extrait un objet HttpSession valide. La transaction valide la fin de la demande d'application Web. Si l'environnement a des demandes qui durent plus de 30 secondes, définissez cette valeur en conséquence.

Valeurs que vous ne pouvez pas changer :

ObjectGridEventListener
La ligne ObjectGridEventListener ne peut pas être modifiée et elle est utilisée en interne.

objectgridSessionMetadata
La ligne objectgridSessionMetadata fait référence à la mappe où sont stockées les métadonnées de session HTTP. Il existe une entrée pour chaque session HTTP stockée dans la grille de données dans cette mappe.

objectgridSessionTTL.*
Cette valeur ne peut pas être modifiée et elle est réservée à une utilisation future.

objectgridSessionAttribute.*
Le texte `objectgridSessionAttribute.*` définit une mappe dynamique. Cette valeur est utilisée pour créer la mappe où sont stockés les attributs de session HTTP lorsque le paramètre **`fragmentedSession`** a la valeur `true`

dans le fichier `splicer.properties`. Cette mappe dynamique s'appelle `objectgridSessionAttribute.*`. Une autre mappe est créée en fonction de ce modèle appelé `objectgridSessionAttributeEvicted` qui stocke les sessions qui ont expiré lorsque le conteneur Web n'a pas invalidé.

La ligne **MapEventListener** est interne et ne peut pas être modifiée.

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

 <objectgridDeployment objectgridName="session">
  <mapSet name="sessionMapSet" numberOfPartitions="5" minSyncReplicas="0" maxSyncReplicas="0"
    maxAsyncReplicas="1" developmentMode="false" placementStrategy="PER_CONTAINER">
             <map ref="objectgridSessionMetadata"/>
             <map ref="objectgridSessionAttribute.*"/>
             <map ref="objectgridSessionTTL.*"/>
    </mapSet>
   </objectgridDeployment>
</deploymentPolicy>
```

Figure 36. Fichier `objectGridDeployment.xml`

Valeurs que vous pouvez modifier :

Attribut de nom ObjectGrid
La valeur doit correspondre aux valeurs suivantes dans les autres fichiers de configuration :

- La propriété **objectGridName** dans le fichier `splicer.properties` qui est utilisé pour raccorder l'application Web.
- L'attribut **name** dans le fichier `objectGrid.xml`.

Si vous disposez de plusieurs applications et voulez stocker les données de session dans des grilles de données différentes, ces applications doivent avoir des valeurs d'attribut de nom ObjectGrid différentes.

Attributs d'élément mapSet
Vous pouvez changer toutes les propriétés mapSet, sauf pour l'attribut placementStrategy.

Name Peut être mis à jour avec n'importe quelle valeur.

numberOfPartitions
Spécifie le nombre de partitions primaires qui sont démarrées sur chacun des serveurs hébergeant l'application Web. Au fur et à mesure que l'on ajoute des partitions, les données sont de plus en plus réparties dans l'éventualité d'un basculement. La valeur par défaut est 5 partitions ; elle convient pour la plupart des applications.

minSyncReplicas, maxSyncReplicas et maxAsyncReplicas
Spécifient le nombre et le type des fragments réplique qui stockent les données de session HTTP. La valeur par défaut est 1 réplique asynchrone, ce qui convient pour la plupart des applications. La réplication synchrone intervient pendant le chemin de demande, ce qui peut augmenter les temps de réponse de l'application Web.

developmentMode
Informe le service de placement eXtreme Scale si les fragments réplique d'une partition peuvent être positionnés sur le même noeud que leur fragment primaire. La valeur peut être définie comme true dans un environnement de développement, mais il est conseillé de désactiver cette fonction en environnement de

production en raison des risques de pertes de données que pourrait provoquer une défaillance du noeud.

placementStrategy
Ne modifiez pas la valeur de cet attribut.

Le reste du fichier se réfère aux mêmes noms de mappes que dans le fichier `objectGrid.xml`. Ces noms ne peuvent pas être modifiés.

Valeurs non modifiables :

- L'attribut placementStrategy dans l'élément mapSet.

Package XML distant

Lorsque vous utilisez le mode distant dans lequel les conteneurs s'exécutent comme des processus autonomes, vous devez utiliser le fichier `objectGridStandAlone.xml` et le fichier `objectGridDeploymentStandAlone.xml` pour démarrer les processus. Vous pouvez modifier ces fichiers pour adapter la configuration.

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
 <objectGrids>
  <objectGrid name="session" txTimeout="30">
   <bean id="ObjectGridEventListener" className="com.ibm.ws.xs.sessionmanager.SessionHandleManager"/>
            <backingMap name="objectgridSessionMetadata" pluginCollectionRef="objectgridSessionMetadata"
       readOnly="false" lockStrategy="PESSIMISTIC" ttlEvictorType="LAST_ACCESS_TIME" timeToLive="3600"
       copyMode="COPY_TO_BYTES"/>
            <backingMap name="objectgridSessionAttribute.*" template="true" readOnly="false" lockStrategy="PESSIMISTIC"
       ttlEvictorType="NONE" copyMode="COPY_TO_BYTES"/>
            <backingMap name="objectgridSessionTTL.*" template="true" readOnly="false" lockStrategy="PESSIMISTIC"
       ttlEvictorType="LAST_ACCESS_TIME" timeToLive="3600" copyMode="COPY_TO_BYTES"/>
  </objectGrid>
 </objectGrids>
    <backingMapPluginCollections>
        <backingMapPluginCollection id="objectgridSessionMetadata">
            <bean id="MapEventListener" className="com.ibm.ws.xs.sessionmanager.MetadataMapListener"/>
        </backingMapPluginCollection>
    </backingMapPluginCollections>
</objectGridConfig>
```

Figure 37. `objectGridStandAlone.xml` file

Valeurs que vous pouvez changer :

Attribut de nom ObjectGrid
La valeur doit correspondre aux valeurs suivantes dans les autres fichiers de configuration :

- La propriété **`objectGridName`** dans le fichier `splicer.properties` qui est utilisé pour raccorder l'application Web.
- L'attribut **`objectgridName`** dans le fichier `objectGridStandAlone.xml`.

Si vous disposez de plusieurs applications et voulez stocker les données de session dans des grilles de données différentes, ces applications doivent avoir des valeurs d'attribut de nom ObjectGrid différentes.

7.1.1+ **Attribut ObjectGrid txTimeout**
Cette valeur détermine le nombre de secondes pendant lequel une transaction peut être ouverte avant que le serveur de conteneur déclenche la fasse expirer. La valeur par défaut est 30 secondes et elle peut être changée en fonction de l'environnement. Si la persistance de session HTTP est configurée avec une valeur de paramètre d'initialisation de contexte de servlet **`replicationInterval`** supérieure à zéro, les transactions sont traitées

par lots dans une unité d'exécution. Si la propriété **replicationInterval** a la valeur 0, une transaction démarre généralement lorsqu'une application Web extrait un objet HttpSession valide. La transaction valide la fin de la demande d'application Web. Si l'environnement a des demandes qui durent plus de 30 secondes, définissez cette valeur en conséquence.

Valeur que vous ne pouvez pas changer :

ObjectGridEventListener
La ligne ObjectGridEventListener ne peut pas être modifiée et elle est utilisée en interne.

objectgridSessionMetadata
La ligne objectgridSessionMetadata fait référence à la mappe où sont stockées les métadonnées de session HTTP. Il existe une entrée pour chaque session HTTP stockée dans la grille de données dans cette mappe.

objectgridSessionTTL.*
Cette valeur ne peut pas être modifiée et elle est réservée à une utilisation future.

objectgridSessionAttribute.*
Le texte objectgridSessionAttribute.* définit une mappe dynamique. Cette valeur est utilisée pour créer la mappe où sont stockés les attributs de session HTTP lorsque le paramètre **fragmentedSession** a la valeur true dans le fichier splicer.properties. Cette mappe dynamique s'appelle objectgridSessionAttribute.*. Une autre mappe est créée en fonction de ce modèle appelé objectgridSessionAttributeEvicted qui stocke les sessions qui ont expiré lorsque le conteneur Web n'a pas invalidé.

Le ligne **MetadataMapListener** est une ligne interne et elle ne peut pas être modifiée.

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

 <objectgridDeployment objectgridName="session">
  <mapSet name="sessionMapSet" numberOfPartitions="5" minSyncReplicas="0" maxSyncReplicas="0"
   maxAsyncReplicas="1" developmentMode="false" placementStrategy="PER_CONTAINER">
            <map ref="objectgridSessionMetadata"/>
            <map ref="objectgridSessionAttribute.*"/>
            <map ref="objectgridSessionTTL.*"/>
     </mapSet>
    </objectgridDeployment>
</deploymentPolicy>
```

Figure 38. objectGridDeploymentStandAlone.xml file

Valeurs que vous pouvez changer :

Attribut objectgridName
La valeur doit correspondre aux valeurs suivantes dans les autres fichiers de configuration :

- La propriété **objectGridName** dans le fichier splicer.properties qui est utilisé pour raccorder l'application Web.
- L'attribut **name** dans le fichier objectGrid.xml.

Si vous disposez de plusieurs applications et voulez stocker les données de session dans des grilles de données différentes, ces applications doivent avoir des valeurs d'attribut de nom ObjectGrid différentes.

Attributs d'élément mapSet

Vous pouvez changer toutes les propriétés mapSet, sauf pour l'attribut placementStrategy.

Name Peut être mis à jour avec n'importe quelle valeur.

numberOfPartitions

Spécifie le nombre de partitions primaires qui sont démarrées sur chacun des serveurs hébergeant l'application Web. Au fur et à mesure que l'on ajoute des partitions, les données sont de plus en plus réparties dans l'éventualité d'un basculement. La valeur par défaut est `5` partitions ; elle convient pour la plupart des applications.

minSyncReplicas, maxSyncReplicas et maxAsyncReplicas

Spécifient le nombre et le type des fragments réplique qui stockent les données de session HTTP. La valeur par défaut est 1 réplique asynchrone, ce qui convient pour la plupart des applications. La réplication synchrone intervient pendant le chemin de demande, ce qui peut augmenter les temps de réponse de l'application Web.

developmentMode

Informe le service de placement eXtreme Scale si les fragments réplique d'une partition peuvent être positionnés sur le même noeud que leur fragment primaire. La valeur peut être définie comme true dans un environnement de développement, mais il est conseillé de désactiver cette fonction en environnement de production en raison des risques de pertes de données que pourrait provoquer une défaillance du noeud.

placementStrategy

Ne modifiez pas la valeur de cet attribut.

Le reste du fichier se réfère aux mêmes noms de mappes que dans le fichier `objectGrid.xml`. Ces noms ne peuvent pas être modifiés.

Valeurs non modifiables :

- L'attribut placementStrategy dans l'élément mapSet.

Paramètres d'initialisation du contexte de servlet

La liste qui suit de paramètres d'initialisation du contexte de servlet peut être spécifiée dans le fichier splicer.properties en fonction de la méthode de raccord choisie.

Paramètres

`objectGridType`

Valeur de type chaîne `REMOTE` ou `EMBEDDED`. La valeur par défaut est `REMOTE`.

Si la valeur est `REMOTE`, les données de session sont stockées en dehors du serveur sur lequel l'application Web est exécutée.

Si la valeur est `EMBEDDED`, un conteneur intégré eXtreme Scale démarre dans le processus serveur d'applications sur lequel l'application Web s'exécute.

`objectGridName`

Valeur de chaîne qui définit le nom de l'instance ObjectGrid utilisée pour une application Web particulière. Le nom par défaut est session.

Cette propriété doit refléter le nom objectGridName dans les fichiers XML ObjectGrid et XLM de déploiement utilisés pour démarrer les serveurs de conteneur eXtreme Scale.

catalogHostPort

Le serveur de catalogues peut être contacté pour obtenir une instance ObjectGrid côté client. La valeur doit avoir le format `host:port<,host:port>`. L'hôte est le programme d'écoute sur lequel le serveur de catalogue s'exécute. Le port est le port d'écoute du processus serveur de catalogue. La longueur de cette liste peut être arbitraire et la liste n'est utilisée que pour l'amorçage. La première adresse viable qui est utilisée. Elle est facultative dans WebSphere Application Server si la propriété **catalog.services.cluster** est défini.

replicationInterval

Entier (en secondes) qui définit le temps séparant deux écritures de sessions actualisées vers la grille. La valeur par défaut est `10` secondes. Les valeurs possibles sont comprises entre 0 et 60. 0 signifie que les sessions actualisées sont écrites dans la grille pour chaque demande dès la fin de l'appel à la méthode de service du servlet. Une valeur **replicationInterval** plus élevée améliore les performances, car un moins grand nombre de mises à jour sont écrites dans la grille de données. Mais, en même temps, une valeur supérieure à 0 rend la configuration moins tolérante aux pannes.

Ce paramètre s'applique uniquement lorsque objectGridType a la valeur `REMOTE`.

sessionTableSize

Entier qui définit le nombre de références de session conservées en mémoire. La valeur par défaut est `1000`.

Ce paramètre appartient uniquement à une topologie `REMOTE`, car la topologie `EMBEDDED` a déjà les données de session dans le même groupe que le conteneur Web.

Les sessions sont expulsées de la table interne en fonction de la logique LRU (least recently used). Lorsqu'une session est expulsée de cette table, elle est invalidée dans le conteneur Web. Cependant, les données ne sont pas pour autant supprimées de la grille, ce qui permet aux demandes ultérieures de cette session de continuer à extraire les données. Cette valeur doit être supérieure à la valeur maximale du pool d'unités d'exécution du conteneur Web, ce qui réduit la contention sur le cache de session.

fragmentedSession

Valeur de type chaîne `true` ou `false`. La valeur par défaut est `true`. Ce paramètre permet de contrôler si le produit stocke les données de session en tant qu'entrée entière ou s'il stocke chaque attribut séparément.

Affectez au paramètre fragmentedSession la valeur `true` si la session d'application Web a de nombreux attributs ou des attributs avec des grandes tailles. Affectez à fragmentedSession la valeur `false` si une session a peu d'attributs, car tous les attributs sont stockés dans la même clé dans la grille de données.

Dans la précédente implémentation à base de filtres, il était fait référence à cette propriété en tant que mécanisme de persistance avec, comme valeurs possibles, ObjectGridStore (fragmentation) et ObjectGridAtomicSessionStore (non-fragmentation).

securityEnabled

Valeur de type chaîne `true` ou `false`. La valeur par défaut est `false`. Ce paramètre active la sécurité du client eXtreme Scale. Il doit correspondre au paramètre **securityEnabled** dans le fichier des propriétés sur serveur eXtreme Scale. Si les paramètres ne correspondent pas, une exception est générée.

credentialGeneratorClass

Le nom de la classe qui implémente l'interface com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator. Cette classe sert à obtenir les données d'identification des clients.

credentialGeneratorProps

Les propriétés de la classe d'implémentation CredentialGenerator. Les propriétés correspondent à l'objet avec la méthode setProperties(String). La valeur credentialGeneratorProps n'est utilisée que si la valeur de la propriété **credentialGeneratorClass** n'est pas null.

objectGridXML

L'emplacement du fichier `objectgrid.xml`. Le fichier XML intégré regroupé dans la bibliothèque eXtreme Scale est chargé automatiquement si `objectGridType=EMBEDDED` et que la propriété **objectGridXML** n'est pas définie.

objectGridDeploymentXML

Indique l'emplacement du fichier XML de stratégie de déploiement d'objectGrid. Le fichier XML intégré regroupé dans la bibliothèque eXtreme Scale est chargé automatiquement si `objectGridType=EMBEDDED` et que la propriété **objectGridDeploymentXML** n'est pas définie.

traceSpec

Spécifie la spécification de trace d'IBM WebSphere comme une valeur de chaîne. Utilisez ce paramètre pour des serveurs d'applications autres que WebSphere Application Server.

traceFile

Spécifie l'emplacement du fichier de trace sous forme de valeur de chaîne. Utilisez ce paramètre pour des serveurs d'applications autres que WebSphere Application Server.

cookieDomain

Spécifie si vous exigez que les sessions soient accessibles à travers les hôtes. Définissez la valeur avec le nom du domaine commun entre les hôtes.

reuseSessionID

A la valeur `true` si le conteneur Web sous-jacent réutilise les ID de session dans les demandes aux différents hôtes. La valeur par défaut est `false`. La valeur de cette propriété doit être la même que la valeur du conteneur Web. Si vous utilisez WebSphere Application Server et configurez la persistance de session HTTP eXtreme Scale en utilisant la console d'administration ou le scriptage de l'outil **wsadmin**, la propriété personnalisée du conteneur Web `HttpSessionIdReuse=true` est ajoutée par défaut. **reuseSessionID** a également la valeur `true`. Si vous ne voulez pas réutiliser l'ID de session, définissez la

propriété HttpSessionIdReuse=false dans la propriété personnalisée du conteneur Web avant de configurer la persistance de session eXtreme Scale.

shareSessionsAcrossWebApps
Spécifie si les sessions sont partagées entre des applications Web ; spécifiée comme valeur de chaîne true ou false. La valeur par défaut est false. La spécification de servlet indique que les sessions HTTP ne peuvent pas être partagées entre des applications Web. Une extension à la spécification de servlet est fournie pour permettre ce partage.

useURLEncoding
Affectez-lui la valeur true pour activer la réécriture d'URL. La valeur par défaut est false, ce qui indique que les cookies sont utilisés pour stocker les données de session. La valeur de ce paramètre doit être identique à celle des paramètres de conteneur Web pour la gestion des sessions.

Fichier splicer.properties

Le fichier splicer.properties contient toutes les options de configuration pour configurer un gestionnaire de sessions basé sur un filtre de servlet.

Exemple de fichier splicer.properties

Si vous décidez d'utiliser l'une des propriétés supplémentaires décrites dans ce fichier, veillez à mettre en commentaire les lignes des propriétés à activer.

```
# Fichier de propriétés qui contient toutes les options de configuration
# que le gestionnaire de sessions ObjectGrid basé sur un filtre de servlet peut être configuré
# pour utiliser.
# Ce fichier de propriétés peut être créé de sorte à conserver toutes
# les valeurs par défaut à affecter à ces paramètres de configuration
# et les paramètres individuels peuvent être remplacés à l'aide des
# propriétés de la tâche ANT, si ce fichier de propriétés est utilisé
# conjointement avec la tâche ANT filtersplicer.

# Valeur de chaîne "REMOTE" ou "EMBEDDED".  La valeur par défaut est REMOTE.
# Si elle est définie sur "REMOTE", les données de session seront stockées en dehors du
# serveur où est exécutée l'application Web. Si sa valeur est
# "EMBEDDED", un conteneur WebSphere eXtreme Scale imbriquée démarre
# dans le processus de serveur d'applications dans lequel l'application Web est exécutée.

objectGridType = REMOTE

# Valeur de chaîne qui définit le nom de l'instance  ObjectGrid
# utilisée pour une applications Web donnée. Le nom par défaut
# est session. Cette propriété doit refléter l'objectGridName dans les deux
# fichiers xml objectgrid et de déploiement utilisés pour démarrer les conteneurs eXtreme
# Scale.

objectGridName = session

# Le serveur de catalogues peut être contacté pour obtenir une instance
# ObjectGrid côté client.  La valeur doit avoir le format
# "host:port<,host:port>", où host est l'hôte d'écoute
# sur lequel le serveur de catalogue est en cours d'exécution, et le port est le
# port d'écoute du processus du serveur de catalogue.
# Cette liste peut être arbitrairement longue et n'est utilisée que pour l'amorçage.
# La première adresse valide est utilisée.  Elle est facultative dans WebSphere
# si la propriété catalog.services.cluster est définie.

# catalogHostPort = host:port<,host:port>

# Entier (secondes) qui définit la durée en secondes entre
# l'écriture de sessions actualisées dans ObjectGrid. La valeur par défaut est 10. Cette propriété
# est utilisée uniquement lorsque objectGridType a la valeur REMOTE. Les valeurs possibles sont
```

```
# comprises entre 0 et 60. 0 signifie que les sessions actualisées sont écrites
# dans l'ObjectGrid
# à la fin de l'appel à la méthode de service de servlet de chaque demande.

replicationInterval = 10

# Entier qui définit le nombre de références de session conservées
# en mémoire. La valeur par défaut est 1 000. Cette propriété est utilisée
# uniquement lorsque objectGridType a la valeur REMOTE.
# Lorsque le nombre de sessions stockées
# dans la mémoire dans le conteneur Web dépasse cette valeur, la première session
# ayant fait l'objet d'un accès
# est invalidée depuis le conteneur Web. Si une demande
# arrive pour cette session une fois qu'elle a été invalidée, une nouvelle session
# est créée (avec un nouvel ID de session reuseSessionId=false),
# remplie avec les attributs de la session invalidée. Cette value doit toujours être
# supérieure à la taille maximale du pool d'unités
# d'exécution du conteneur pour éviter les conflits dans ce cache de session.

sessionTableSize = 1000

# Valeur de type chaîne "true" ou "false". La valeur par défaut est "true".
# Permet de contrôler si nous stockons les données de session comme entrée
# intégrale ou de stocker chaque attribut séparément.
# Cette propriété s'appelle persistenceMechanism dans l'implémentation
# basée sur un filtre précédente, avec les valeurs possibles
# ObjectGridStore (fragmenté) et ObjectGridAtomicSessionStore
# (non fragmenté).

fragmentedSession = true

# Valeur de type chaîne "true" ou "false". La valeur par défaut est "false".
# Active la sécurité du client eXtreme Scale. Ce paramètre doit correspondre
# au paramètre securityEnabled dans le fichier des propriétés du serveur eXtreme
# Scale. Si les paramètres ne correspondent pas, une exception
est générée.

securityEnabled = false

# Spécifie la prise en charge de l'authentification des données
# d'identification du client.
#   Les valeurs possibles sont les suivantes :
#   Jamais : le client ne prend pas en charge l'authentification des
#   données d'identification.
#   Pris en charge* : le client prend en charge l'authentification des données
#   d'identification si et seulement si le serveur
#                la prend en charge également.
#   Obligatoire : le client requiert l'authentification des données
#   d'identification.
#   Elle est prise en charge par défaut.

# credentialAuthentication =

# Indique le nombre de tentatives d'authentification si les données
# d'identification
# ont expiré. Si la valeur est 0, aucune tentative d'authentification
# n'a lieu.

# authenticationRetryCount =

# Indique le nom de la classe qui implémente l'interface
# com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator.
# Cette classe utilisée pour obtenir les données d'identification
# des clients.

# credentialGeneratorClass =
```

```
# Spécifie les propriétés de la classe d'implémentation
# CredentialGenerator. Les propriétés sont définies dans l'objet
# avec la méthode setProperties(String).
# La valeur credentialGeneratorProps est utilisée uniquement si la
# valeur de la propriété credentialGeneratorClass est null.

# credentialGeneratorProps =

# Emplacement du fichier xml objectgrid.
# Le fichier xml pré-intégré qui est regroupé dans la
# bibliothèque eXtreme Scale
# sera automatiquement chargé si cette propriété
# n'est pas spécifiée et que objectGridType=EMBEDDED

# objectGridXML =

# Emplacement du fichier xml de stratégie de déploiement objectGrid.
# Le fichier xml pré-intégré qui est regroupé dans la
# bibliothèque eXtreme Scale
# sera automatiquement chargé si cette propriété
# n'est pas spécifiée et que objectGridType=EMBEDDED

# objectGridDeploymentXML =

# Chaîne de spécification de trace IBM WebShere,
# utile pour tous les autres serveurs d'applications, outre WebSphere.

# traceSpec =

# Chaîne d'emplacement de fichier de trace.
# utile pour tous les autres serveurs d'applications, outre WebSphere.

# traceFile=

# Cette propriété doit être définie pour que les sessions soient
# accessibles sur les hôtes. La valeur sera le nom du domaine
# commun aux hôtes.

# cookieDomain=

# A la valeur true si le conteneur Web sous-jacent
# réutilise l'ID dans les demandes à différents hôtes. La valeur
# par défaut est
# false. La valeur doit être identique à celle définie dans
# le conteneur Web.

# reuseSessionId=

# Valeur de chaîne "true" ou "false". La valeur par défaut est
# "false". Conformément à la spécification de servlet, les sessions HTTP
# ne peuvent pas être partagées dans les applications Web. Une extension
# à la spécification de servlet
# est fournie pour autoriser le partage.

# shareSessionsAcrossWebApps = false

# Affectez-lui la valeur true si vous voulez activer la réécriture
# d'URL (urlRewriting).  La valeur par défaut est
# false, ce qui signifie que les cookies seront utilisés pour stocker
# les données. La valeur doit refléter ce qui est
# défini dans les paramètres de
# conteneur Web pour la gestion de sessions.

# useURLEncoding = false
```

Configuration du fournisseur de cache dynamique pour WebSphere eXtreme Scale

L'installation et la configuration du fournisseur de cache dynamique d'eXtreme Scale varient en fonction de vos exigences et de l'environnement que vous avez configuré.

Avant de commencer

- Pour pouvoir utiliser le fournisseur de cache dynamique, WebSphere eXtreme Scale doit être installé sur les déploiements de noeud WebSphere Application Server et notamment le noeud du gestionnaire de déploiement. Pour plus d'informations, voir «Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client avec WebSphere Application Server», à la page 163.
- La sécurité globale doit être activée dans la console d'administration de WebSphere Application Server si les serveurs de catalogues de votre domaine de services de catalogue ont SSL activé ou si vous voulez utiliser SSL pour un domaine de services de catalogue avec SSL pris en charge. Vous activer SSL pour un serveur de catalogue en affectant à l'attribut la valeur `SSL-Required` dans Fichier de propriétés du serveur. Pour plus d'informations sur la configuration de la sécurité globale, voir Paramètres de sécurité globale.

Pourquoi et quand exécuter cette tâche

Pour savoir comment utiliser le fournisseur de cache dynamique d'eXtreme Scale avec IBM WebSphere Commerce, voir les rubriques suivantes dans la documentation d'IBM WebSphere Commerce :

- Activation du service de cache dynamique et mise en cache de servlet
- Activation du cache de donnés WebSphere Commerce

Si vous ne dirigez pas spécifiquement votre mise en cache vers une instance ObjectCache ou ServletCache définie, il y a de fortes chances que les appels à l'API DynamicCache soient traités par baseCache, le cache de base. Si vous souhaitez utiliser le fournisseur de cache dynamique eXtreme Scale pour JSP, les services Web ou la mise en cache des commandes, vous devez définir l'instance baseCache pour utiliser le fournisseur de cache dynamique eXtreme Scale. Les mêmes propriétés de configuration permettent de configurer l'instance baseCache. Notez que ces propriétés de configuration doivent être définies comme propriétés personnalisées Java Virtual Machine (JVM). Cette restriction s'applique à toutes les propriétés de configuration de cache décrites dans cette section, à l'exception de la mise en cache de servlet. Pour utiliser eXtreme Scale avec le fournisseur de cache dynamique pour la mise en cache de servlet, veillez à configurer l'activation dans les propriétés système et non pas dans les propriétés personnalisées.

Procédure

1. Activez le fournisseur de cache dynamique d'eXtreme Scale.
   - **WebSphere Application Server Version 7.0 et suivantes :**

     Vous pouvez configurer le service de cache dynamique pour utiliser le fournisseur de cache dynamique eXtreme Scale avec la console d'administration. Après avoir installé eXtreme Scale, le fournisseur de cache dynamique eXtreme Scale est immédiatement disponible comme option **Fournisseur de cache** dans la console d'administration. Pour plus d'informations, voir le centre de documentation WebSphere Application Server Version 7.0 : Selecting a cache service provider.
   - **WebSphere Application Server Version 6.1 :**

Utilisez une propriété personnalisée pour configurer le service de cache dynamique pour utiliser le fournisseur de cache dynamique eXtreme Scale. Vous pouvez également utiliser ces propriétés personnalisées dans WebSphere Application Server Version 7.0 et les versions suivantes. Pour créer une propriété personnalisée dans une instance de cache, cliquez sur **Ressources** > **Instances de cache** > **cache_instance_type** > **cache_instance_name** > **Propriétés personnalisées** > **New**. Si vous utilisez l'instance de cache de base, créez les propriétés personnalisées de la machine JVM.

com.ibm.ws.cache.CacheConfig.cacheProviderName
Pour utiliser le fournisseur de cache dynamique eXtreme Scale, définissez la valeur `com.ibm.ws.objectgrid.dynacache.CacheProviderImpl`. Vous pouvez créer cette propriété personnalisée dans une instance de cache dynamique ou l'instance de cache de base. Si vous choisissez de définir la propriété personnalisée dans l'instance de cache de base, toutes les autres instances de mémoire cache sur le serveur utilisent le fournisseur eXtreme Scale par défaut. Les propriétés de configuration du fournisseur de cache dynamique eXtreme Scale définies pour baseCache sont les propriétés de configuration par défaut pour toutes les instances de cache sauvegardées par eXtreme Scale. Pour remplacer l'instance de cache de base et créer une instance de cache dynamique, utilisez le fournisseur de cache dynamique par défaut, créez la propriété personnalisée com.ibm.ws.cache.CacheConfig.cacheProviderName dans l'instance de mémoire cache dynamique et définissez la valeur `default`.

2. Facultatif : Si vous utilisez des instances de cache répliqué, configurez le paramètre de réplication de la mémoire cache.

   Avec le fournisseur de cache dynamique eXtreme Scale, vous pouvez avoir des instances de cache local ou des instances de cache répliqué. Si vous utilisez uniquement des instances de cache local, vous pouvez ignorer cette étape.

   Utilisez l'une des méthodes suivantes pour configurer le cache répliqué :

   - Activer la réplication de cache à l'aide de la console d'administration. Vous pouvez activer la réplication de la mémoire cache à tout moment dans WebSphere Application Server Version 7.0. Dans WebSphere Application Server Version 6.1, vous devez créer un domaine de réplication DRS.
   - Activer la réplication de cache à l'aide de la propriété personnalisée com.ibm.ws.cache.CacheConfig.enableCacheReplication pour forcer le cache à signaler qu'il s'agit d'un cache répliqué, bien qu'un domaine de réplication DRS ne lui ait pas été affecté. Affectez la valeur `true` à cette propriété. Définissez cette propriété personnalisée dans l'instance de mémoire cache si vous utilisez une mémoire cache d'objet ou de servlet, ou sur la machine virtuelle Java si vous utilisez l'instance baseCache.

3. Facultatif : Si vous utilisez eXtreme Scale comme un cache de fragments JSP, définissez la propriété personnalisée com.ibm.ws.cache.CacheConfig.disableTemplateInvalidation avec la valeur `true` pour désactiver les invalidations basées sur les modèles lors des rechargements JSP.

4. Configurez la topologie pour le service de cache dynamique.

   Le seul paramètre de configuration requis pour le fournisseur de cache dynamique d'eXtreme Scale est la topologie du cache. Définissez cette propriété personnalisée dans l'instance de mémoire cache ou pour le service de cache

dynamique si vous utilisez l'instance baseCache. Entrez le nom de la propriété personnalisée comme suit : `com.ibm.websphere.xs.dynacache.topology`.

Les trois valeurs possibles de cette propriété sont les suivantes. Vous devez utiliser l'une des valeurs autorisées :

- `embedded`
- `embedded_partitioned`
- `remote`

Si vous utilisez des topologies intégrées ou partitionnées intégrées, vous pouvez définir la propriété personnalisée `com.ibm.ws.cache.CacheConfig.ignoreValueInInvalidationEvent` en lui affectant la valeur `true` pour réduire l'impact de la sérialisation. Définissez cette propriété personnalisée dans l'instance de mémoire cache ou la machine virtuelle Java si vous utilisez l'instance baseCache.

5. Facultatif : Si vous utilisez une topologie partitionnée intégrée, configurez le nombre de conteneurs initiaux pour le service de mise en cache dynamique.

   Vous pouvez maximiser les performances des mémoires cache qui utilisent la topologie partitionnée intégrée en configurant le nombre initial de conteneurs. Définissez la variable comme propriété système dans la machine virtuelle WebSphere Application Server Java.

   Entrez le nom de la propriété comme suit : `com.ibm.websphere.xs.dynacache.num_initial_containers`.

   La valeur recommandée pour cette propriété de configuration est un entier qui est égal ou légèrement inférieur au nombre total d'instances WebSphere Application Server qui accèdent à cette instance de cache réparti. Par exemple, si un service de cache dynamique est partagé entre les membres d'une grille de données, la valeur doit correspondre au nombre de membres de la grille de données.

   Pour des topologies intégrées ou partitionnées intégrées, vous devez utiliser la version 7.0 de WebSphere Application Server. Définissez la propriété suivante dans le processus JVM pour vous assurer que les conteneurs initiaux sont immédiatement disponibles.

   ```
   com.ibm.ws.cache.CacheConfig.createCacheAtServerStartup=true
   ```

6. Configurez la grille du service de catalogue d'eXtreme Scale.

   Lorsque vous utilisez eXtreme Scale comme le fournisseur de mémoire cache dynamique pour une instance de cache réparti, vous devez configurer un domaine de services de catalogue eXtreme Scale.

   Un même domaine de services de catalogue peut traiter plusieurs fournisseurs de services de cache dynamique s'appuyant sur eXtreme Scale.

   Un service de catalogue peut être exécuté à l'intérieur ou à l'extérieur des processus WebSphere Application Server. Depuis eXtreme Scale Version 7.1, lorsque vous utilisez la console d'administration pour configurer les domaine de services de catalogue, le cache dynamique utilise ces paramètres. Il n'est pas nécessaire de procéder à une configuration supplémentaire pour définir un service de catalogue. Pour plus d'informations, voir «Création de domaines de services de catalogue dans WebSphere Application Server», à la page 258.

7. Configurez les objets de clé personnalisés.

   Si vous utilisez des objets personnalisés comme des clés, ces objets doivent implémenter l'interface Serializable ou Externalizable. Lorsque vous utilisez des topologies intégrées ou des topologies partitionnées intégrées, vous devez placer les objets dans le chemin d'accès à la bibliothèque partagée de WebSphere, comme s'ils étaient utilisés avec le fournisseur de cache dynamique par défaut. Pour plus de détails, voir Utilisation des interfaces DistributedMap

et DistributedObjectCache pour le cache dynamique dans le Centre de documentation de WebSphere Application Server Network Deployment.

Si vous utilisez la topologie éloignée, vous devez placer les objets de clé personnalisés dans le chemin d'accès aux classes (CLASSPATH) des conteneurs eXtreme Scale autonomes. Pour plus d'informations, voir «Démarrage des serveurs de conteneur», à la page 398.

8. Facultatif : Si vous utilisez une topologie distante, configurez les serveurs de conteneureXtreme Scale.
   - **Topologie partitionnée intégrée ou intégrée :**

     Les données en mémoire cache sont stockées dans les serveurs de conteneur WebSphere eXtreme Scale. Un service de catalogue peut être exécuté à l'intérieur ou à l'extérieur des processus WebSphere Application Server. Le fournisseur eXtreme Scale crée automatiquement des conteneurs dans le processus WebSphere si vous utilisez des topologies intégrées ou partitionnées intégrées pour une instance de cache. Aucune configuration supplémentaire n'est requise pour ces topologies.
   - **Topologie distante :**

     Si vous utilisez la topologie distante, vous devez démarrer les serveurs de conteneur autonomes eXtreme Scale avant les instances WebSphere Application Server qui accèdent au démarrage de l'instance de cache. Voir les étapes de démarrage des serveurs de conteneur autonomes dans le *Guide d'administration* p pour plus d'informations. Vérifiez que tous les serveurs de conteneur d'un service de cache dynamique pointent vers les mêmes noeuds finaux de service de catalogue.

     Les fichiers de configuration XML des conteneurs de fournisseur de cache dynamique eXtreme Scale se trouvent dans le répertoire *racine_install_wxs*/customLibraries/ObjectGrid/dynacache/etc pour les installations sur WebSphere Application Server, ou le répertoire *racine_install_wxs*/ObjectGrid/dynacache/etc pour les installations autonomes. Les fichiers s'intitulent dynacache-remote-objectgrid.xml et dynacache-remote-definition.xml. Créez une copie de ces fichiers pour les modifier et les utiliser lorsque vous démarrez des conteneurs autonomes pour le fournisseur de cache dynamique eXtreme Scale. Le paramètre **numInitialContainers** dans le fichier **dynacache-remote-deployment.xml** doit correspondre au nombre de processus conteneur qui sont en cours d'exécution. Notez que l'attribut **numberOfPartitions** dans le fichier dynacache-remote-objectgrid.xml a la valeur par défaut 47.

     Remarque : L'ensemble des processus serveur conteneur doit disposer d'une mémoire suffisante pour traiter toutes les instances de cache dynamique qui sont configurées pour utiliser la topologie distante. Tous les processus WebSphere Application Server qui partagent les mêmes valeurs ou des valeurs équivalentes pour la propriété personnalisée catalog.services.cluster doivent utiliser le même ensemble de conteneurs autonomes. Le nombre de conteneurs et le nombre de serveurs sur lesquels ils résident doivent être dimensionnés de manière appropriée. Pour plus de détails, voir «Planification de la capacité de la mémoire cache dynamique», à la page 59.

     Voici une entrée de ligne de commande qui lance un conteneur autonome pour le fournisseur de cache dynamique eXtreme Scale :

     UNIX

```
startOgServer.sh container1 -objectGridFile
../dynacache/etc/dynacache-remote-objectgrid.xml -deploymentPolicyFile
../dynacache/etc/dynacache-remote-deployment.xml -catalogServiceEndPoints
MyServer1.company.com:2809
```

9. Dans le cas de topologies réparties ou intégrées, activez l'agent de dimensionnement pour améliorer les estimations d'utilisation de la mémoire.

   L'agent de dimensionnement estime l'utilisation qui sera faite de la mémoire (statistiques usedBytes). L'agent requiert Java 5 ou supérieur.

   Chargez l'agent en ajoutant l'argument suivant à la ligne de commande de la machine virtuelle Java :

   ```
   -javaagent:rép_biblio_WXS/wxssizeagent.jar
   ```

   Dans le cas d'une topologie intégrée, ajoutez l'argument à la ligne de commande du processus WebSphere Application Server.

   Dans le cas d'une topologie répartie, ajoutez l'argument à la ligne de commande des processus eXtreme Scale (conteneurs) et du processus WebSphere Application Server.

Plug-in de cache niveau 2 (L2) JPA

WebSphere eXtreme Scale inclut des plug-in de mémoire cache de niveau 2 pour les fournisseurs OpenJPA et Hibernate Java Persistence API (JPA). Lorsque vous utilisez l'un de ces plug-ins, l'application utilise l'API JPA. Une grille de données est introduite ente l'application et la base de données pour améliorer les temps de réponse.

L'utilisation d'eXtreme Scale en tant que fournisseur de cache de niveau 2 améliore les performances lors de la lecture et de l'interrogation des données et réduit la charge pesant sur la base de données. WebSphere eXtreme Scale présente plusieurs avantages par rapport aux implémentations de cache pré-intégrées car le cache est automatiquement répliqué entre tous les processus. Lorsqu'un client met une valeur en cache, tous les autres clients peuvent utiliser la valeur mise en cache en local.

Vous pouvez configurer la topologie et les propriétés pour le fournisseur de cache L2 dans le fichier `persistence.xml`. Pour plus d'informations sur la configuration de ces propriétés, voir «Propriétés de configuration du cache JPA», à la page 338.

Conseil : Le plug-in de cache L2 JPA requiert une application qui utilise les API JPA. Si vous souhaitez utiliser les API WebSphere eXtreme Scale pour accéder à une source de données JPA, utilisez le chargeur JPA. Pour plus d'informations, voir Chargeurs JPA.

Remarques relatives à la topologie cache L2 JPA

Les facteurs suivants affectent le type de topologie à configurer :

1. **Quelle quantité de données voulez-vous placer en mémoire cache ?**
   - Si les données peuvent tenir dans un seul segment de mémoire JVM, utilisez la «Topologie imbriquée», à la page 333 ou «Topologie intra-domaine», à la page 332.
   - Dans le cas contraire, utilisez la «Topologie imbriquée et partitionnée», à la page 334 ou «Topologie distante», à la page 336
2. **Quel est le taux de lecture/écriture prévu ?**

   Ce taux affecte les performances du cache L2. Chaque topologie gères différemment les opérations de lecture et d'écriture.
   - «Topologie imbriquée», à la page 333 : lecture locale, écriture distante
   - «Topologie intra-domaine», à la page 332 : lecture locale, écriture locale

- «Topologie imbriquée et partitionnée», à la page 334 : partitionnée : lecture distante, écriture distante
- «Topologie distante», à la page 336 : lecture distante, écriture distante.

Les applications qui fonctionnent principalement en lecture seule doivent utiliser des topologies intra-domaines lorsque cela est possible. Les applications qui exécutent des opération d'écriture principalement doivent utiliser des topologies intra-domaines.

3. **Quel est le pourcentage de données recherchées par rapport au pourcentage de données trouvées par une clé ?**

   Lorsque le cache des requêtes JPA est activé, les opérations d'interrogation l'utilisent. Activez ce cache pour les applications avec des taux de lecture/écriture élevés uniquement, par exemple, lorsque vous approchez de 99 % d'opérations de lecture. Si vous utilisez le cache des requêtes JPA, vous devez utiliser «Topologie imbriquée», à la page 333 ou «Topologie intra-domaine».

   L'opération de recherche par clé recherche une entité cible si l'entité cible n'a pas de relation. Si l'entité cible a des relations avec le type de recherche EAGER, ces relations sont recherchées avec l'entité cible. Dans le cache de données JPA, un petit nombre de réussites en mémoire obtient toutes les données de relation lors de la recherche de ces relations.

4. **Quel est niveau d'obsolescence toléré des données ?**

   Dans un système comportant un petit nombre de machines JVM, il existe une latence de réplication des données pour les opérations d'écriture. Le cache a pour fonction de gérer une vue de données synchronisée dans toutes les machines JVM. Lorsque vous utilisez la topologie intra-domaine, il existe un délai de réplication de données pour les opérations d'écriture. Les applications qui utilisent cette topologie doivent pouvoir tolérer les lectures obsolètes et les écritures simultanées qui peuvent remplacer les données.

7.1.1+
Topologie intra-domaine

Avec une topologie intra-domaine, les fragments primaires sont placés sur chaque serveur de conteneur dans la topologie. Ces fragments primaires contiennent l'ensemble des données de la partition. N'importe lequel de ces fragments primaires peut également exécuter des opérations d'écriture dans la mémoire cache. Cette configuration élimine le goulot d'étranglement dans la topologie intégrée dans lequel toutes les opérations d'écriture de la mémoire cache doivent passer par un fragment primaire unique.

Dans une topologie intra-domaine, aucun fragment de réplique n'est créé, même si vous avez défini des répliques dans vos fichiers de configuration. Chaque fragment primaire redondant contient une copie complète des données, de sorte qu'il peut également être considéré comme un fragment de réplique. Cette configuration utilise une partition unique, similaire à la topologie intégrée.

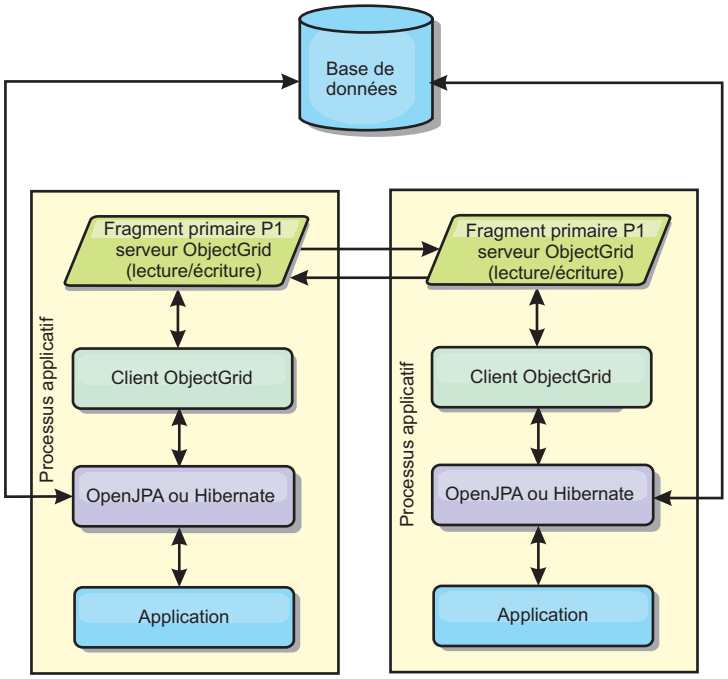


Figure 39. Topologie intra-domaine JPA

Propriétés de configuration du cache JPA associées pour la topologie intra-domaine :

```
ObjectGridName=objectgrid_name,ObjectGridType=EMBEDDED,PlacementScope=CONTAINER_SCOPE,PlacementScopeTopology=HUB | RING
```

Avantages :

- Lectures de cache et des mises à jour localement
- Simple à configurer.

Limitations :

- Cette topologie est la mieux adaptée lorsque les serveurs de conteneur peuvent contenir l'ensemble des données de la partition.
- Les fragments de réplique, même s'ils sont configurés, ne sont jamais placés, car chaque serveur de conteneur héberge un fragment primaire. Toutefois, tous les fragments primaires sont répliqués avec les autres fragments primaires, de sorte que ces fragments primaires deviennent des répliques les uns des autres.

Topologie imbriquée

Conseil : Envisagez d'utiliser une topologie intra-domaine pour obtenir de meilleures performances.

Une topologie imbriquée crée un serveur de conteneur dans l'espace de traitement de chaque application. Les plug-in OpenJPA et Hibernate lisent directement la copie en mémoire du cache et écrivent dans toutes les autres copies. Vous pouvez améliorer les performances d'écriture à l'aide de la réplication asynchrone. Cette topologie par défaut produit un résultat optimal lorsque la quantité de données mises en cache est suffisamment réduite pour être traitée par un seul processus. Avec une topologie intégrée, créez une seule partition pour les données.

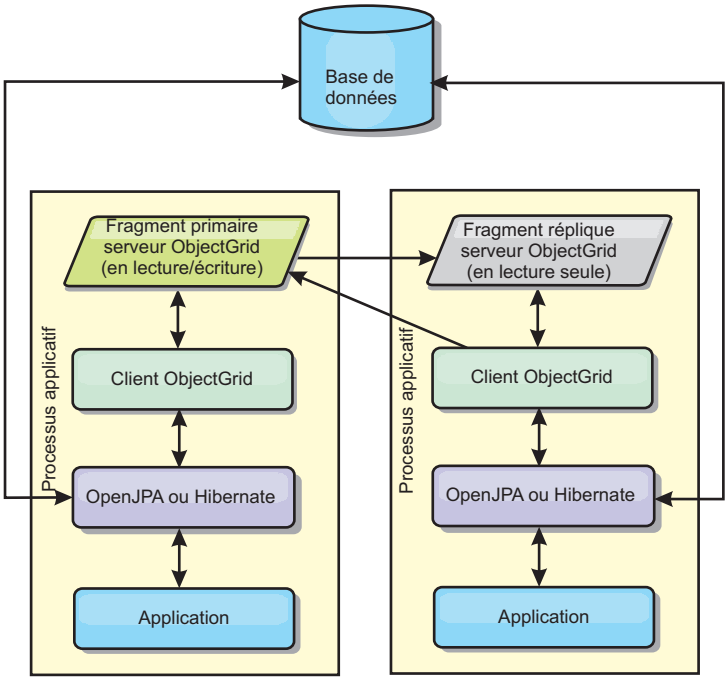


Figure 40. Topologie imbriquée JPA

Propriétés de configuration du cache JPA de la topologie intégrée :

```
ObjectGridName=objectgrid_name,ObjectGridType=EMBEDDED,MaxNumberOfReplicas=num_replicas,ReplicaMode=SYNC | ASYNC | NONE
```

Avantages :

- Toutes les lectures de cache sont des accès locaux rapides.
- Simple à configurer.

Limitations :

- La quantité de données est limitée à la taille du processus.
- Toutes les mises à jour de cache sont envoyées via un fragment primaire, ce qui crée un goulot d'étranglement.

Topologie imbriquée et partitionnée

Conseil : Envisagez d'utiliser une topologie intra-domaine pour obtenir de meilleures performances.

ATTENTION :
N'utilisez pas le cache des requêtes JPA avec un topologie partitionnée. Le cache de requêtes stocke les résultats des requêtes qui sont une collection de clés d'entité. Le cache des requêtes recherche les données d'entité dans l'antémémoire données. Comme l'antémémoire données est divisée entre plusieurs processus, ces appels supplémentaires peuvent faire perdre les avantages du cache L2.

Lorsque les données en mémoire cache sont trop volumineuses pour tenir dans un seul processus, vous pouvez utiliser la topologie partitionnée intégrée. Cette topologie divise le données dans plusieurs processus. Les données sont divisées entre les fragments primaires de sortie que chaque fragment primaire contient un sous-ensemble des données. Vous pouvez toujours utiliser cette option lorsque la latence de la base de données est élevée.

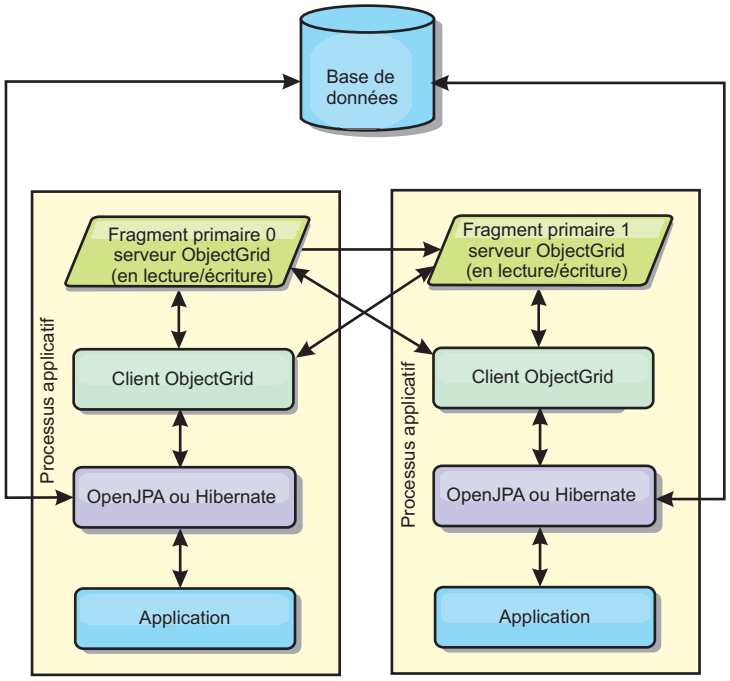


Figure 41. Topologie imbriquée et partitionnée JPA

Propriétés de configuration du cache JPA de la topologie partitionnée intégrée :

```
ObjectGridName=objectgrid_name,ObjectGridType=EMBEDDED_PARTITION,ReplicaMode=SYNC | ASYNC | NONE,
NumberOfPartitions=num_partitions,ReplicaReadEnabled=TRUE | FALSE
```

Avantages :

- Stocke de grandes quantités de données.
- Simple à configurer.
- Les mises à jour de cache sont réparties sur plusieurs processus.

Limitation :

- La plupart des lectures et des mises à jour de cache sont distantes.

Par exemple, pour mettre en cache 10 Go de données avec un maximum de 1 Go par machine JVM, 10 machines virtuelles Java sont nécessaires. Le nombre de partitions doit par conséquent être défini sur 10 ou plus. Idéalement, le nombre de partitions doit être un nombre premier où chaque fragment stocke une quantité raisonnable de mémoire. Le paramètre numberOfPartitions est généralement égal au nombre de machines virtuelles Java. Chaque machine virtuelle Java stocke une partition à l'aide de ce paramètre. Si vous activez la réplication, vous devez augmenter le nombre de machines virtuelles Java dans le système. Dans le cas contraire, chaque machine virtuelle Java stocke également une réplique de partition qui consomme autant de mémoire que la partition principale.

Consultez la rubrique relative à la définition de la taille de la mémoire et au calcul du nombre de partitions dans le *Guide d'administration* pour optimiser les performances de la configuration choisie.

Par exemple, dans un système avec quatre machines virtuelles Java et avec la valeur de paramètre numberOfPartitions 4, chaque machine virtuelle Java héberge une partition principale. Une opération de lecture a 25 pourcents de chances d'extraire des données d'une partition disponible en local, ce qui est sensiblement

plus rapide qu'à partir d'une machine virtuelle Java distante. Si une opération de lecture, telle que l'exécution d'une requête, doit extraire une collection de données impliquant une répartition égale de quatre partitions, 75 pourcents des appels sont distants et 25 pourcents sont locaux. Si le paramètre ReplicaMode est défini sur `SYNC` ou `ASYNC` et si le paramètre ReplicaReadEnabled est défini sur `true`, quatre répliques de partitions sont créées et réparties entre quatre machines virtuelles Java. Chaque machine virtuelle Java héberge une partition principale et une réplique. L'opération de lecture a désormais à 50 pourcents de chances de s'exécuter en local. L'opération de lecture qui extrait une collection de données impliquant une répartition égale de quatre partitions comporte 50 pourcents d'appels distants et 50 pourcents d'appels locaux. Les appels locaux sont considérablement plus rapides que les appels distants. Dès que des appels distants sont effectués, les performances chutent.

Topologie distante

ATTENTION :
N'utilisez pas le cache des requêtes JPA avec une topologie distante. Le cache des requêtes stocke les résultats des requêtes qui sont une collection de clés d'entité. Le cache des requêtes utilise l'antémémoire données pour rechercher toutes les données d'entité. Comme l'antémémoire données est distante, ces appels supplémentaires peuvent faire perdre les avantages du cache L2.

Conseil : Envisagez d'utiliser une topologie intra-domaine pour obtenir de meilleures performances.

Une topologie distante stocke toutes les donnés mises en cache dans un ou plusieurs processus, ce qui réduit la sollicitation de la mémoire par les processus applicatifs. Vous pouvez tirer parti de la répartition de vos données dans des processus distincts en déployant une grille de données eXtreme Scale partitionnée répliquée. Contrairement aux configurations intégrées et intégrées et partitionnées décrites dans les sections précédentes, si vous souhaitez gérer la grille de données distante, vous devez le faire indépendamment de l'application et du fournisseur JPA.

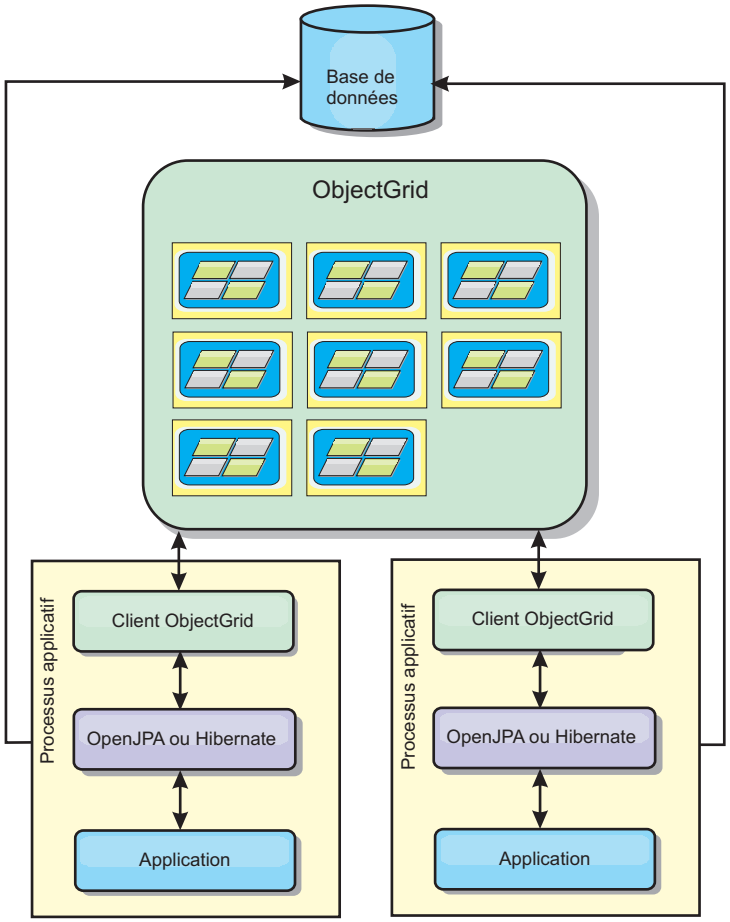


Figure 42. Topologie distante JPA

Propriétés de configuration du cache JPA de la topologie distante :

`ObjectGridName=`*`objectgrid_name`*`,ObjectGridType=REMOTE`

Le type d'ObjectGrid REMOTE ne nécessite pas de paramètres de propriété car l'ObjectGrid et la règle de déploiement sont définis distinctement de l'application JPA. Le plug-in de cache JPA se connecte à distance à un ObjectGrid éloigné existant.

Toute interaction avec la grille d'objets étant éloignée, cette topologie offre les moins bonnes performances parmi tous les types de grille d'objets.

Avantages :

- Stocke de grandes quantités de données.
- Le processus applicatif est exempt de données en cache.
- Les mises à jour de cache sont réparties sur plusieurs processus.
- Options de configuration souples.

Limitation :

- Toutes lectures et mises à jour de cache sont distantes.

Propriétés de configuration du cache JPA

WebSphere eXtreme Scale inclut des plug-in de cache de niveau 2 pour les fournisseurs Java Persistence API (JPA) OpenJPA et Hibernate. Pour configurer le plug-in de cache L2, vous devez mettre à jour les propriétés dans le fichier `persistence.xml`.

Conseil : Le plug-in de cache L2 JPA requiert une application qui utilise les API JPA. Si vous souhaitez utiliser les API WebSphere eXtreme Scale pour accéder à une source de données avec JPA, utilisez le chargeur JPA. Pour plus d'informations, voir «Configuration des chargeurs JPA», à la page 353.

Emplacement des propriétés

Vous pouvez définir ces propriétés dans le fichier `persistence.xml`. La syntaxe pour spécifier les propriétés de ce fichier varie selon que vous utilisez OpenJPA ou Hibernate :

- **OpenJPA :** Vous pouvez définir les propriétés sur le DataCache ou QueryCache

```
<property name="openjpa.DataCache"
 value="<object_grid_datacache_class(<propriété>=<valeur>,...)"/>
```

ou

```
<property name="openjpa.QueryCache"
 value="<object_grid_querycache_class(<propriété>=<valeur>,...)"/>
```

- **Hibernate :**

```
<property name="objectgrid.configuration" value="<property>=<value>,..." />
```

Topologie et propriétés par défaut

Les valeurs de propriétés par défaut suivantes sont utilisées si vous ne spécifiez pas de valeurs dans la configuration :

- **ObjectGridName** : nom de l'unité de persistance
- **ObjectGridType** : `EMBEDDED`
- **NumberOfPartitions** : 1 (ne peut pas être modifiée si le type d'ObjectGrid est `EMBEDDED`)
- **ReplicaMode** : `SYNC`
- **ReplicaReadEnabled** : `TRUE` (ne peut pas être modifiée si le type d'ObjectGrid est `EMBEDDED`)
- **MaxUsedMemory** : `TRUE`
- **MaxNumberOfReplicas** : `47` (doit être inférieur ou égal au nombre de machines virtuelles Java dans un système réparti)

Propriétés

Vous pouvez configurer des plug-in de cache JPA avec les propriétés suivantes.

ObjectGridName
Indique le nom unique d'ObjectGrid. La valeur par défaut est le nom d'unité de persistance défini. Si le nom de l'unité de persistance n'est pas disponible auprès du fournisseur, un nom généré est utilisé.

ObjectGridType
Indique le type d'ObjectGrid.

Valeurs admises :

EMBEDDED
Type de configuration par défaut et recommandée. Ses valeurs par défaut sont : `NumberOfPartitions=1`, `ReplicaMode=SYNC`, `ReplicaReadEnabled=true` et `MaxNumberOfReplicas=47`. Utilisez le paramètre **ReplicaMode** pour définir le mode de réplication et le paramètre **MaxNumberOfReplicas** pour définir le nombre maximal de fragments réplique. Si un système possède plus de 47 machines virtuelles Java, spécifiez pour **MaxNumberOfReplicas** une valeur égale au nombre de machines virtuelles Java.

EMBEDDED_PARTITION
Type à utiliser si le système doit mettre en cache une quantité de données importante sur un système réparti. Le nombre de partitions par défaut est `47` avec le mode de réplique `NONE`. Sur un petit système qui ne possède que quelques machines virtuelles Java, spécifiez pour **NumberOfPartitions** une valeur égale ou inférieure au nombre de machines virtuelles Java. Vous pouvez spécifier les valeurs **ReplicaMode**, **NumberOfPartitions** et **ReplicaReadEnabled** pour optimiser le système.

REMOTE Le cache tente de se connecter à un ObjectGrid réparti distant à partir du service de catalogue.

MaxNumberOfReplicas
Indique le nombre maximal de répliques à utiliser pour le cache. Cette valeur s'applique au type `EMBEDDED` uniquement. Ce nombre doit être égal ou supérieur au nombre de machines virtuelles Java dans un système. La valeur par défaut est `47`.

Valeurs valides : Valeurs supérieures ou égales à 1

MaxUsedMemory

Valeurs valides : `TRUE` ou `FALSE`Active l'expulsion des entrées du cache si la mémoire est soumise à des contraintes. La valeur par défaut est `TRUE` et les données sont expulsées lorsque le seuil d'utilisation des segments de mémoire de la machine virtuelle Java dépasse 70 pourcent. Vous pouvez modifier le seuil d'utilisation des segments de mémoire de la machine virtuelle Java par défaut en définissant la propriété memoryThresholdPercentage dans le fichier `objectGridServer.properties` et en plaçant ce fichier dans le chemin d'accès aux classes. Pour plus d'informations sur les expulseurs, voir les Plug-in d'expulsion d'objets du cacheinformations sur les expulseurs dans *Présentation du produit*. Pour plus d'informations sur le fichier de propriétés du serveur, voir Fichier de propriétés du serveur.

NumberOfPartitions

Valeurs valides : Valeurs supérieures ou égales à 1Indique le nombre de partitions à utiliser pour le cache. Cette propriété s'applique si la valeur d'ObjectGridType est `EMBEDDED_PARTITION`. La valeur par défaut est `47`. Pour le type `EMBEDDED`, la valeur **NumberOfPartitions** est toujours 1.

7.1.1+ **PlacementScope**
Indique la granularité d'une instance d'un groupe de mappes.

Valeurs admises :

DOMAIN_SCOPE
(Défaut) Place un fragment primaire pour chaque partition sur un serveur de conteneur dans le domaine de services de catalogue.

Les fragments de réplique pour chaque partition sont placés sur les autres serveurs de conteneur dans le domaine de services de catalogue.

CONTAINER_SCOPE
Place un fragment primaire sur chaque serveur de conteneur dans le domaine de services de catalogue

7.1.1+ **PlacementScopeTopology**
Définit la topologie de liaison du serveur de conteneur dans le domaine de services de catalogue. Cette valeur est utilisée uniquement lorsque PlacementScope a une valeur différente de `DOMAIN_SCOPE`.

Valeurs admises :

HUB (Défaut) Si la topologie de concentrateur est sélectionnée, une grille de données unique est sélectionnée pour être le concentrateur. Toutes les autres grilles de données se connectent au concentrateur. Cette topologie est assez évolutive, car branches ont une connexion unique. Le concentrateur peut devenir un goulot d'étranglement et un point unique de défaillance temporaire. Le concentrateur est transféré vers un autre serveur de conteneur lorsque il est défaillant. Cette configuration offre l'avantage de pouvoir écrire plus de code que ne le permet un seul point, le concentrateur, pour gérer toutes les collisions.

RING Si la topologie en anneau est sélectionnée, chaque grille de données est liée à deux autres grilles de données. L'ordre des liaisons n'est pas garantie. Toutefois, chaque conteneur qui démarre est probablement liée au premier conteneur et au dernier conteneur à ajouter à l'anneau. Cette topologie est la plus évolutive, mais il suffit de deux liaisons défaillantes pour qu'une déconnexion temporaire se produise. Si les serveurs de conteneurs sont défaillants, des liaisons sont établies parmi les survivants après que l'échec a été détecté.

ReplicaMode

Valeurs valides : `SYNC/ASYNC/NONE` Indique la méthode utilisée pour copier le cache vers les fragments réplique. Cette propriété s'applique si la valeur d'ObjectGridType est `EMBEDDED` ou `EMBEDDED_PARTITION`. La valeur par défaut est `NONE` pour le type `EMBEDDED_PARTITION` et `SYNC` pour le type `EMBEDDED`. Si la valeur de **ReplicaMode** est `NONE` pour le type de grille d'objets `EMBEDDED`, le type `EMBEDDED` utilise la valeur `SYNC` pour **ReplicaMode**.

ReplicaReadEnabled

Valeurs valides : `TRUE` ou `FALSE`Si cette propriété est activée, les clients lisent les valeurs à partir des fragments réplique. Cette propriété s'applique au type `EMBEDDED_PARTITION`. La valeur par défaut est FALSE pour le type `EMBEDDED_PARTITION`. Le type `EMBEDDED` affecte toujours à la propriété **ReplicaReadEnabled** la valeur `TRUE`.

writeBehind

Pour les fournisseurs Hibernate uniquement : lorsque la mise en cache en écriture différée est activée, les mises à jour sont provisoirement stockées dans une mémoire de données de portée JVM jusqu'à ce que la condition writeBehindInterval ou writeBehindMaxBatchSize soit respectée.

Avertissement : Les autres paramètres de configuration de l'écriture différée sont ignorés sauf si writeBehind est activé.

Important : Utilisez la fonction d'écriture différée avec précaution. Les configurations d'écriture différée allonge la synchronisation des données dans toutes les machines JVM et augmentent le risque de perte de données. Dans un système qui utilise la configuration d'écriture différée avec au moins quatre machines JVM, la mise à jour effectuée sur une machine virtuelle Java correspond à un délai d'environ 15 secondes avant que la mise à jour soit disponible pour les autres machines JVM. Si deux des machines virtuelles Java actualisent la même entrée, la première qui vide la mise à jour perd sa mise à jour.

Valeurs valides : TRUE ou FALSE.

Valeur par défaut : FALSE.

writeBehindInterval

Pour les fournisseurs Hibernate uniquement : spécifie l'intervalle de temps en millisecondes de vidage des mises à jour dans le cache.

Valeurs valides : supérieur ou égal à 1.

Valeur par défaut : 5000 (5 secondes).

writeBehindPoolSize

Pour les fournisseurs Hibernate uniquement : spécifie la taille maximale du pool d'unités d'exécution utilisé pour vider les mises à jour dans le cache.

Valeurs valides : supérieure ou égale à 1.

Valeur par défaut : 5.

writeBehindMaxBatchSize

Pour les fournisseurs Hibernate uniquement : spécifie la taille de lot maximale par cache de région pour le vidage des mises à jour dans le cache. Par exemple, si la taille est 1 000 et que les mises à jour stockées dans le stockage d'écriture différée d'un cache de région dépasse 1 000 entrées, les mises à jour sont vidées dans le cache, même si la condition writeBehindInterval est respectée. Les mises à jour sont vidées dans le cache à peu près dans le délai en secondes spécifié par la valeur writeBehindInterval ou lorsque la taille du stockage d'écriture différée dans chacun des caches de région dépasse 1 000 entrées. Notez que, si la condition writeBehindMaxBatchSize est remplie, seul le cache de région qui remplit cette condition vide ses mises à jour dans le stockage d'écriture différée vers le cache. Un cache de région correspond généralement à une entité ou à une requête.

Valeurs valides : supérieur ou égal à 1.

Valeur par défaut : 1000.

Configuration du plug-in de cache OpenJPA

Vous pouvez configurer les deux implémentations DataCache et QueryCache pour OpenJPA.

Avant de commencer

- Vous devez déterminer la topologie du plug-in de cache JPA à utiliser. Voir «Plug-in de cache niveau 2 (L2) JPA», à la page 331 pour plus d'informations sur les différentes configurations et propriétés à définir pour chaque topologie.
- Vous devez disposer d'une application utilisant les API JPA. Si vous souhaitez utiliser les API WebSphere eXtreme Scale pour accéder aux données avec JPA, utilisez le chargeur JPA. Pour plus d'informations, voir «Configuration des chargeurs JPA», à la page 353.

Procédure

1. Définissez les propriétés dans le fichier `persistence.xml` pour configurer le plug-in de cache OpenJPA : Vous pouvez définir ces propriétés dans l'implémentation de cache DataCache ou Query.

   Les configurations de DataCache et de QueryCache sont indépendantes l'une de l'autre. Vous pouvez activer l'une ou l'autre. Mais, si les deux configurations sont activées, la configuration de QueryCache utilisera celle de DataCache et ses propriétés à elle seront ignorées.

   ```
   <property name="openjpa.DataCache"
             value="<object_grid_datacache_class(<propriété>=<valeur>,...)"/>
   ```

   ou

   ```
   <property name="openjpa.QueryCache"
             value="<object_grid_querycache_class(<propriété>=<valeur>,...)"/>
   ```

   Remarque : Vous pouvez activer le configuration QueryCache pour les topologies intradomaines intégrées et non intégrées uniquement.

   Dans la liste de la classe de cache ObjectGrid, vous pouvez spécifier la propriété ObjectGridName, la propriété ObjectGridType ou toute autre propriétés en rapport avec des règles de déploiement simple pour personnaliser la configuration du cache. Exemple :

   ```
   <property name="openjpa.DataCache"
             value="com.ibm.websphere.objectgrid.openjpa.ObjectGridDataCache(
             ObjectGridName=BasicTestObjectGrid,ObjectGridType=EMBEDDED,
             maxNumberOfReplicas=4)"/>
   <property name="openjpa.QueryCache"
             value="com.ibm.websphere.objectgrid.openjpa.ObjectGridQueryCache()"/>
   <property name="openjpa.RemoteCommitProvider" value="sjvm"/>
   ```

   Voir «Propriétés de configuration du cache JPA», à la page 338 pour la liste des propriétés que vous pouvez définir.

2. Dans le fichier `persistence.xml`, vous devez affecter à la propriété openjpa.RemoteCommitProvider la valeur `sjvm`.

   ```
   <property name="openjpa.RemoteCommitProvider" value="sjvm"/>
   ```

3. Facultatif : Pour personnaliser davantage la grille de données utilisée par le cache, vous pouvez fournir des paramètres supplémentaires avec des fichiers XML.

   Dans la plupart des cas, la définition des propriétés du cache est amplement suffisante. Pour personnaliser davantage l'ObjectGrid utilisé par le cache, vous pouvez fournir des fichiers XML de configuration OpenJPA ObjectGrid dans le répertoire `META-INF`, similairement au fichier `persistence.xml`. Pendant l'initialisation, le cache tente de localiser ces fichiers XML et les traite s'il les trouve.

   Il existe trois types de fichiers XML de configuration OpenJPA ObjectGrid :

   - `openjpa-objectGrid.xml` (ObjectGrid configuration)

Chemin du fichier : `META-INF/openjpa-objectGrid.xml`

Ce fichier sert à personnaliser une configuration d'ObjectGrid de type EMBEDDED ou de type EMBEDDED_PARTITION. Si l'ObjectGrid est de type REMOTE, ce fichier est ignoré. Par défaut, chaque classe d'entité est mappée à sa propre configuration BackingMap désignée au sein de la configuration de l'ObjectGrid sous le nom de la classe. Ainsi, la classe d'entité com.mycompany.Employee sera mappée à la configuration BackingMap com.mycompany.Employee. La configuration BackingMap par défaut est `readOnly="false"`, `copyKey="false"`, `lockStrategy="NONE"` et `copyMode="NO_COPY"`. Vous pouvez tout à fait personnaliser des mappes de sauvegarde avec la configuration que vous choisissez. Le mot clé réservé `ALL_ENTITY_MAPS` représente tous les mappages à l'exclusion des mappages personnalisés répertoriés dans le fichier `openjpa-objectGrid.xml`. Les mappes de sauvegarde qui ne figurent pas dans ce fichier `openjpa-objectGrid.xml` utilisent la configuration par défaut. Si les mappes de sauvegarde personnalisées ne spécifient pas l'attribut ou les propriétés BackingMaps et que ces attributs sont spécifiés dans la configuration par défaut, ce sont les valeurs des attributs dans cette configuration qui s'appliquent. Par exemple, si une classe d'entité est annotée avec `timeToLive=30`, la configuration BackingMap par défaut de cette entité aura un `timeToLive=30`. Si le fichier personnalisé `openjpa-objectGrid.xml` inclut également la mappe de sauvegarde mais sans spécifier de valeur pour timeToLive value, la mappe personnalisée aura la valeur `timeToLive=30` qui est la valeur par défaut. Le fichier `openjpa-objectGrid.xml` a pour finalité de remplacer ou d'étendre la configuration par défaut.

- `openjpa-objectGridDeployment.xml` (stratégie de déploiement)

  Chemin du fichier : `META-INF/openjpa-objectGridDeployment.xml`

  Ce fichier sert à personnaliser la règle de déploiement. Lorsque celle-ci est personnalisée, si le fichier `openjpa-objectGridDeployment.xml` est fourni, la règle de déploiement par défaut est ignorée. Toutes les valeurs d'attribut de la stratégie de déploiement proviennent du fichier `openjpa-objectGridDeployment.xml` fourni.

- `openjpa-objectGrid-client-override.xml` (configuration de remplacement ObjectGrid client)

  Chemin du fichier : `META-INF/openjpa-objectGrid-client-override.xml`

  Ce fichier sert à personnaliser un ObjectGrid côté client. Par défaut, le cache de l'ObjectGrid applique une configuration par défaut de substitution des ObjectGrid par les clients, qui désactive les caches locaux (near cache). Si une application a besoin d'un cache local, elle peut fournir ce fichier en y spécifiant `numberOfBuckets="xxx"`. Le remplacement de client par défaut désactive le cache local en définissant `numberOfBuckets="0"`. Pour activer le cache local, il suffit de donner à numberOfBuckets une valeur supérieure à 0 dans le fichier `openjpa-objectGrid-client-override.xml`. Le fonctionnement du fichier `openjpa-objectGrid-client-override.xml` est semblable à celui du fichier `openjpa-objectGrid.xml`. Le fichier remplace ou étend la configuration d'ObjectGrid.

Pour personnaliser cette topologie, vous pouvez fournir le fichier XML adapté au type de l'eXtreme Scale configuré.

Pour le type EMBEDDED comme pour le type EMBEDDED_PARTITION, vous pouvez fournir n'importe lequel de ces trois fichiers XML pour personnaliser l'ObjectGrid, la règle de déploiement et la configuration de la substitution des ObjectGrid clients.

Dans le cas d'un ObjectGrid REMOTE, le cache ne crée pas d'ObjectGrid dynamique. Le cache ne contient en fait qu'un ObjectGrid côté client provenant du service de catalogue. Dans ce cas, vous pouvez fournir que le fichier `openjpa-objectGrid-client-override.xml` qui personnalisera la configuration de la substitution de l'ObjectGrid client.

4. Facultatif : (Configurations distantes uniquement) Définissez un système eXtreme Scale externe si vous voulez configurer un cache avec un type REMOTE ObjectGrid.

   Afin de pouvoir configurer un cache d'ObjectGrid de type REMOTE, vous devez configurer un système externe eXtreme Scale. Pour configurer ce système externe, vous aurez besoin des deux fichiers XML de configuration ObjectGrid et ObjectGridDeployment basés sur un fichier `persistence.xml`. Pour des exemples de ces fichiers de configuration, voir «Exemple : fichiers XML OpenJPA ObjectGrid», à la page 345.

Résultats

Configuration **EMBEDDED, EMBEDDED_PARTITION, ou intra-domaine** :

Lors du démarrage d'une application, le plug-in détecte automatiquement un service de catalogue ou en démarre un, démarre un serveur de conteneur et connecte les serveurs de conteneur au service de catalogue. Le plug-in communique alors avec le conteneur ObjectGrid et ses homologues exécutés dans d'autres processus de serveur d'applications à l'aide de la connexion client.

Configuration REMOTE :

La stratégie de déploiement est spécifiée séparément de l'application JPA. Un système ObjectGrid externe comporte le service de catalogue et les processus de serveurs de conteneur. Vous devez démarrer le service de catalogue avant les serveurs de conteneur. Pour plus d'informations, reportez-vous aux rubriques «Démarrage des serveurs autonomes», à la page 395 et «Démarrage des serveurs de conteneur», à la page 398.

Que faire ensuite

- Développez une application OpenJPA qui utilise la configuration. Pour plus d'informations, voir Exemple: Utilisation du plug-in Hibernate pour précharger les données dans le cache ObjectGrid.
- Dans un environnement de production, créez des domaines de services de catalogue pour les processus automatiquement créés pour votre configuration EMBEDDED ou EMBEDDED_PARTITION.
  - Environnement autonome :

    Si vous n'exécutez pas vos serveurs dans un processus WebSphere Application Server, les hôtes et les ports du domaine de services de catalogue sont spécifiés à l'aide du fichier de propriétés `objectGridServer.properties`. Ce fichier doit être stocké dans le chemin d'accès aux classes de l'application et la propriété **catalogServiceEndPoints** doit être définie. Le domaine de services de catalogue est démarré indépendamment des processus d'application et doit être démarré avant les processus d'application.

    Le format du fichier `objectGridServer.properties` est le suivant :

    ```
    catalogServiceEndPoints=<hostname1>:<port1>,<hostname2>:<port2>
    ```

  - Environnement WebSphere Application Server :

Lors d'une exécution à l'intérieur d'un processus WebSphere Application Server, le plug-in de cache JPA se connecte automatiquement au service de catalogue (ou au domaine de services de catalogue) qui est défini pour la cellule WebSphere Application Server.

- Si vous utilisez le type de grille d'objets EMBEDDED ou EMBEDDED_PARTITION dans un environnement Java SE, utilisez la méthode System.exit(0) à la fin du programme pour arrêter le serveur eXtreme Scale imbriqué. Sinon, le programme peut ne pas répondre.

Exemple : fichiers XML OpenJPA ObjectGrid :

Les fichiers XML OpenJPA ObjectGrid XML doivent être créés à partir de la configuration de l'unité de persistance.

Fichier persistence.xml

Voici à titre d'exemple un fichier persistence.xml représentant la configuration d'une unité de persistance :

```
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  version="1.0">
  <persistence-unit name="AnnuityGrid">
    <provider>org.apache.openjpa.persistence.PersistenceProviderImpl</provider>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistebleObject</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.FixedAnnuity</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.EquityAnnuity</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Person</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityHolder</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Address</class>
    <exclude-unlisted-classes>true</exclude-unlisted-classes>

    <properties>
    <!-- Database setting -->


    <!--  enable cache -->
      <property name="openjpa.DataCache"
        value="com.ibm.websphere.objectgrid.openjpa.ObjectGridDataCache(objectGridName=Annuity,
               objectGridType=EMBEDDED, maxNumberOfReplicas=4)" />
      <property name="openjpa.RemoteCommitProvider" value="sjvm" />
      <property name="openjpa.QueryCache"
                value="com.ibm.websphere.objectgrid.openjpa.ObjectGridQueryCache()"/>
    </properties>
  </persistence-unit>
</persistence>
```

Fichier openjpa-objectGrid.xml

Le fichier openjpa-objectGrid.xmlsert à personnaliser une configuration d'ObjectGrid de type EMBEDDED et de type EMBEDDED_PARTITION. Et voici le fichier openjpa-objectGrid.xml correspondant à ce fichier persistence.xml :

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="Annuity">
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Address" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Address" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person" />
```

```
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistebleObject"
                  readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistebleObject" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout" />
      <backingMap name="ObjectGridQueryCache" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" pluginCollectionRef="ObjectGridQueryCache"
                  evictionTriggers="MEMORY_USAGE_THRESHOLD"  />
    </objectGrid>
  </objectGrids>
  <backingMapPluginCollections>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity">
      <bean id="ObjectTransformer"
            className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Address">
      <bean id="ObjectTransformer"
            className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor">
      <bean id="ObjectTransformer"
            className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person">
      <bean id="ObjectTransformer"
            className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact">
      <bean id="ObjectTransformer"
            className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection
            id="com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistebleObject">
      <bean id="ObjectTransformer"
            className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider">
      <bean id="ObjectTransformer"
            className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout">
      <bean id="ObjectTransformer"
            className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="ObjectGridQueryCache">
      <bean id="MapIndexPlugin" className="com.ibm.websphere.objectgrid.plugins.index.HashIndex" >
          <property name="Name" type="java.lang.String"
                    value="QueryCacheKeyIndex" description="name of index"/>
          <property name="POJOKeyIndex" type="boolean" value="true" description="POJO Key Index "/>
      </bean>
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
  </backingMapPluginCollections>
</objectGridConfig>
```

Important :

1. Chaque entité est mappée à une mappe de sauvegarde qui porte le nom qualifié complet de la classe de cette entité.

Par défaut, les entités font partie de la mémoire cache de second niveau. Dans les classes Entity qui doivent être exclues de la mise en cache, vous pouvez inclure l'annotation `@DataCache(enabled=false)` dans la classe Entity à exclure du cache L2:

```
import org.apache.openjpa.persistence.DataCache;
@Entity
@DataCache(enabled=false)
public class OpenJPACacheTest { ... }
```

2. Si les classes d'entités sont dans une hiérarchie d'héritage, les classes enfants se mappent à la mappe de sauvegarde parent. La hiérarchie d'héritage partage une même mappe de sauvegarde.
3. Le mappage ObjectGridQueryCache map est indispensable pour la prise en charge de QueryCache.
4. L'ObjectTransformer de la backingMapPluginCollection de chaque mappage d'entrée doit utiliser la classe com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer.
5. L'index de clé de la backingMapPluginCollection d'un mappage d'ObjectGridQueryCache être nommé QueryCacheKeyIndex (voir l'exemple).
6. L'expulseur (evictor) est facultatif pou chaque mappage.

Fichier `openjpa-objectGridDeployment.xml`

Le fichier `openjpa-objectGridDeployment.xml` permet de personnaliser la stratégie de déploiement. Voici le fichier `openjpa-objectGridDeployment.xml` qui correspond au fichier `persistence.xml` :

`openjpa-objectGridDeployment.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
  xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
  <objectgridDeployment objectgridName="Annuity">
    <mapSet name="MAPSET_Annuity" numberOfPartitions="1" numInitialContainers="1"
            minSyncReplicas="0" maxSyncReplicas="4" maxAsyncReplicas="0"
            replicaReadEnabled="true">
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Address" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistebleObject" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout" />
      <map ref="ObjectGridQueryCache" />
    </mapSet>
  </objectgridDeployment>
</deploymentPolicy>
```

Remarque : Le mappage ObjectGridQueryCache map est indispensable pour la prise en charge de QueryCache.

Configuration du plug-in de mémoire cache Hibernate

Vous pouvez activer le cache pour utiliser le plug-in de cache Hibernate en définissant des fichiers de propriétés.

Avant de commencer

- Vous devez déterminer la topologie du plug-in de cache JPA à utiliser. Voir «Plug-in de cache niveau 2 (L2) JPA», à la page 331 pour plus d'informations sur les différentes configurations.
- Vous devez disposer d'une application utilisant les API JPA. Si vous souhaitez utiliser les API WebSphere eXtreme Scale pour accéder aux données avec JPA, utilisez le chargeur JPA. Pour plus d'informations, voir «Configuration des chargeurs JPA», à la page 353.

Procédure

1. Si vous utilisez WebSphere Application Server, placez les fichiers JAR (Java Archive) dans les emplacements appropriés.

   Le plug-in de cache Hibernate est regroupé dans le fichier `oghibernate-cache.jar` et installé dans le répertoire *racine_was*`/optionalLibraries/ObjectGrid`. Pour utiliser le plug-in de cache Hibernate, vous devez inclure le fichier `oghibernate-cache.jar` dans la bibliothèque Hibernate. Par exemple, si vous incluez la bibliothèque Hibernate dans votre application, vous devez également inclure le fichier `oghibernate-cache.jar`. Si vous définissez une bibliothèque partagée pour inclure la bibliothèque Hibernate, vous devez ajouter le fichier `oghibernate-cache.jar` dans le répertoire de cette bibliothèque partagée.

   eXtreme Scale n'installe pas le fichier `cglib.jar` dans l'environnement WebSphere Application Server. Si vous avez des applications existantes ou des bibliothèques partagées comme Hibernate, qui dépendent de `cglib.jar`, localisez le fichier `cglib.jar` et incluez-le dans le chemin d'accès aux classes. Par exemple, si votre application inclut tous les fichiers JAR de la bibliothèque Hibernate, mais exclut le fichier `cglib.jar` fourni avec Hibernate, vous devez inclure le fichier `cglib.jar` fourni par Hibernate dans votre application.

2. Définissez les propriétés dans le fichier `persistence.xml` pour configurer le plug-in de cache Hibernate.

   Syntaxe de la définition des propriétés dans le fichier `persistence.xml` :

   ```
   <property name="hibernate.cache.provider_class"
           value="com.ibm.websphere.objectgrid.hibernate.cache.ObjectGridHibernateCacheProvider" />
   <property name="hibernate.cache.use_query_cache" value="true"/>
   <property name="objectgrid.configuration" value="<property>=<value>,..." />
   <property name="objectgrid.hibernate.regionNames" value="<regionName>,.." />
   ```

   - **`hibernate.cache.provider_class`** : la valeur de la propriété **`provider_class`** est la classe com.ibm.websphere.objectgrid.hibernate.cache. ObjectGridHibernateCacheProvider.
   - **`hibernate.cache.use_query_cache`** : pour activer le cache des requêtes, affectez la valeur `true` à la propriété **`use_query_cache`**.

     Remarque : Vous pouvez activer le cache des requêtes pour les topologies intradomaines intégrées et non intégrées uniquement.
   - **`objectgrid.configuration`** : utilisez la propriété objectgrid.configuration pour définir les propriétés de configuration du cache eXtreme Scale, y compris l'attribut ObjectGridType qui indique comment placer les fragments dans la grille de données.

     Vous devez spécifier une valeur de propriété ObjectGridName unique pour éviter les conflits de nom. Les autres propriétés de configuration du cache d'eXtreme Scale sont facultatives.

     Pour activer la mise en cache en écriture différée, utilisez les attributs d'écriture différée suivants dans la propriété objectgrid.configuration. Lorsque la mise en cache en écriture différée est activée, les mises à jour sont provisoirement stockées dans une mémoire de données de portée JVM jusqu'à ce que la condition writeBehindInterval ou writeBehindMaxBatchSize soient respectées lorsque les données sont vidées dans le cache.

     ```
     writeBehind=true, writeBehindInterval=5000, writeBehindPoolSize=10, writeBehindMaxBatchSize=1000
     ```

     Avertissement : Les autres paramètres de configuration de l'écriture différée sont ignorés sauf si writeBehind est activé.

     Pour plus d'informations sur les valeurs que vous pouvez définir dans la propriété **`objectgrid.configuration`**, voir «Propriétés de configuration du cache JPA», à la page 338.

- **objectgrid.hibernate.regionNames** : la propriété objectgrid.hibernate.regionNames est facultative et doit être définie lorsque les valeurs regionNames sont définies après l'initialisation du cache eXtreme Scale. Prenons l'exemple d'une classe d'entité mappée sur une valeur regionName dont la classe d'entité n'est pas spécifiée dans le fichier `persistence.xml` ou n'est pas incluse dans le fichier de mappage Hibernate. En outre, cette classe d'entité comporte une annotation d'entité. La valeur regionName pour cette classe d'entité est ainsi résolue au moment du chargement de la classe lors de l'initialisation du cache d'eXtreme Scale. La méthode Query.setCacheRegion(String regionName) exécutée après l'initialisation du cache d'eXtreme Scale illustre également cette configuration. Dans ces situations, incluez toutes les éventuelles valeurs regionNames déterminées de façon dynamique dans la propriété objectgrid.hibernate.regionNames de sorte que le cache d'eXtreme Scale puisse préparer les mappes de sauvegarde pour toutes les valeurs regionNames.

3. Facultatif : Pour personnaliser davantage la grille de données utilisée par le cache, vous pouvez fournir des paramètres supplémentaires avec des fichiers XML.

   Dans la plupart des cas, la définition des propriétés du cache est amplement suffisante. Toutefois, si vous souhaitez peaufiner la personnalisation de la grille d'objets utilisée par le cache, vous pouvez mettre à disposition dans votre répertoire `META-INF` des fichiers XML de configuration de Hibernate ObjectGrid, à la manière du fichier `persistence.xml`. Pendant l'initialisation, le cache tente de localiser ces fichiers XML et les traite s'il les trouve.

   Il existe trois types de fichiers XML de configuration Hibernate ObjectGrid :

   - `hibernate-objectGrid.xml` (configuration ObjectGrid)

     Chemin du fichier : `META-INF/hibernate-objectGrid.xml`

     Par défaut, chaque classe d'entité est associée à une valeur regionName (par défaut au nom de la classe d'entité) mappée sur une configuration BackingMap désignée sous regionName au sein de la configuration de l'ObjectGrid. Par exemple, la classe d'entité com.mycompany.Employee est associée à une valeur regionName qui a la valeur par défaut com.mycompany.Employee BackingMap. La configuration BackingMap par défaut est readOnly="false", copyKey="false", lockStrategy="NONE" et `copyMode="NO_COPY"`. Vous pouvez tout à fait personnaliser des mappes de sauvegarde avec la configuration que vous choisissez. Le mot clé réservé ALL_ENTITY_MAPS représente tous les mappages à l'exclusion des mappages personnalisés répertoriés dans le fichier `hibernate-objectGrid.xml`. Les mappes de sauvegarde qui ne figurent pas dans ce fichier `hibernate-objectGrid.xml` utilisent la configuration par défaut.

   - `hibernate-objectGridDeployment.xml` (stratégie de déploiement)

     Chemin du fichier : `META-INF/hibernate-objectGridDeployment.xml`

     Ce fichier sert à personnaliser la règle de déploiement. Lorsque celle-ci est personnalisée, si le fichier `hibernate-objectGridDeployment.xml` est fourni, la règle de déploiement par défaut est ignorée. Toutes les valeurs d'attribut de la règle de déploiement proviennent du fichier `hibernate-objectGridDeployment.xml` fourni.

   - `hibernate-objectGrid-client-override.xml` (configuration de remplacement ObjectGrid client)

     Chemin de fichier : `META-INF/hibernate-objectGrid-client-override.xml`

     Ce fichier sert à personnaliser un ObjectGrid côté client. Par défaut, le cache de l'ObjectGrid applique une configuration par défaut de remplacement par

les clients, qui désactive le cache local. Si l'application a besoin d'un cache local, elle peut fournir ce fichier en y spécifiant `numberOfBuckets="xxx"`. Le remplacement de client par défaut désactive le cache local en définissant `numberOfBuckets="0"`. Pour activer le cache local, il suffit de donner à numberOfBuckets une valeur supérieure à 0 dans le fichier `hibernate-objectGrid-client-override.xml`. Le fonctionnement du fichier `hibernate-objectGrid-client-override.xml` est semblable à celui du fichier `hibernate-objectGrid.xml` : Il substitue ou étend la configuration par défaut des remplacements ObjectGrid par les clients.

Pour personnaliser cette topologie, vous pouvez fournir le fichier XML adapté au type de l'eXtreme Scale configuré.

Pour le type EMBEDDED comme pour le type EMBEDDED_PARTITION, vous pouvez fournir n'importe lequel de ces trois fichiers XML pour personnaliser la grille d'objets, la règle de déploiement et la configuration des remplacements par les clients ObjectGrid.

Dans le cas d'un ObjectGrid REMOTE, le cache ne crée pas d'ObjectGrid dynamique. Le cache ne contient en fait qu'un ObjectGrid côté client provenant du service de catalogue. Dans ce cas, vous ne pouvez fournir qu'un fichier `hibernate-objectGrid-client-override.xml` qui personnalisera la configuration de la substitution de l'ObjectGrid client.

4. Facultatif : (Configurations distantes uniquement) Définissez un système eXtreme Scale externe si vous voulez configurer un cache avec un type REMOTE ObjectGrid.

   Afin de pouvoir configurer un cache d'ObjectGrid de type REMOTE, vous devez configurer un système externe eXtreme Scale. Pour configurer ce système externe, vous aurez besoin des deux fichiers XML de configuration ObjectGrid et ObjectGridDeployment basés sur un fichier `persistence.xml`. Pour des exemples de ces fichiers de configuration, voir «Exemple : fichiers XML ObjectGrid Hibernate», à la page 351.

Résultats

Configuration EMBEDDED ou EMBEDDED_PARTITION :

Lors du démarrage d'une application, le plug-in détecte automatiquement un service de catalogue ou en démarre un, démarre un serveur de conteneur et connecte les serveurs de conteneur au service de catalogue. Le plug-in communique alors avec le conteneur ObjectGrid et ses homologues exécutés dans d'autres processus de serveur d'applications à l'aide de la connexion client.

Chaque entité JPA possède une mappe de sauvegarde indépendante affectée à l'aide du nom de classe de l'entité. Chaque mappe de sauvegarde possède les attributs ci-après.

- `readOnly="false"`
- `copyKey="false"`
- `lockStrategy="NONE"`
- `copyMode="NO_COPY"`

Configuration REMOTE :

La règle de déploiement est spécifiée séparément de l'application JPA. Un système ObjectGrid externe comporte le service de catalogue et les processus serveur de conteneur. Vous devez démarrer le service de catalogue avant les serveurs de

conteneur. Pour plus d'informations, reportez-vous aux rubriques «Démarrage des serveurs autonomes», à la page 395 et «Démarrage des serveurs de conteneur», à la page 398.

Que faire ensuite

- Développez une application Hibernate qui utilise la configuration. Pour plus d'informations, voir Exemple: Utilisation du plug-in Hibernate pour précharger les données dans le cache ObjectGrid.
- Dans un environnement de production, créez des domaine de services de catalogue pour vos processus automatiquement créés pour votre configuration.
  - Environnement autonome :

    Si vous n'exécutez pas vos serveurs dans un processus WebSphere Application Server, les hôtes et les ports du domaine de services de catalogue sont spécifiés à l'aide du fichier de propriétés `objectGridServer.properties`. Ce fichier doit être stocké dans le chemin d'accès aux classes de l'application et la propriété **catalogServiceEndPoints** doit être définie. Le domaine de services de catalogue est démarré indépendamment des processus d'application et doit être démarré avant les processus d'application.

    Le format du fichier `objectGridServer.properties` est le suivant :

    ```
    catalogServiceEndPoints=<hostname1>:<port1>,<hostname2>:<port2>
    ```
  - Environnement WebSphere Application Server :

    Lors d'une exécution à l'intérieur d'un processus WebSphere Application Server, le plug-in de cache JPA se connecte automatiquement au service de catalogue (ou au domaine de services de catalogue) qui est défini pour la cellule WebSphere Application Server.
- Si vous utilisez le type de grille d'objets `EMBEDDED` ou `EMBEDDED_PARTITION` dans un environnement Java SE, utilisez la méthode System.exit(0) à la fin du programme pour arrêter le serveur eXtreme Scale imbriqué. Sinon, le programme peut ne pas répondre.

Exemple : fichiers XML ObjectGrid Hibernate :

Les fichiers XML Hibernate ObjectGrid doivent être créés à partir de la configuration de l'unité de persistance.

Fichier `persistence.xml`

Voici à titre d'exemple un fichier `persistence.xml` représentant la configuration d'une unité de persistance :

```
<persistence xmlns="http://java.sun.com/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  version="1.0">
  <persistence-unit name="AnnuityGrid">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>

    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistebleObject</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.FixedAnnuity</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.EquityAnnuity</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Person</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityHolder</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Address</class>

    <exclude-unlisted-classes>true</exclude-unlisted-classes>

    <properties>
      <property name="hibernate.show_sql" value="false" />
      <property name="hibernate.connection.url" value="jdbc:db2:Annuity" />
      <property name="hibernate.connection.driver_class" value="com.ibm.db2.jcc.DB2Driver" />
      <property name="hibernate.default_schema" value="EJB30" />
```

```
      <!-- Cache -->
      <property name="hibernate.cache.provider_class"
        value="com.ibm.websphere.objectgrid.hibernate.cache.ObjectGridHibernateCacheProvider" />
      <property name="hibernate.cache.use_query_cache" value="true" />
      <property name="objectgrid.configuration" value="ObjectGridType=EMBEDDED,
               ObjectGridName=Annuity, MaxNumberOfReplicas=4" />
    </properties>
  </persistence-unit>

</persistence>
```

Fichier hibernate-objectGridDeployment.xml

Utilisez le fichier hibernate-objectGridDeployment.xml pour éventuellement personnaliser la stratégie de déploiement. Si vous fournissez ce fichier dans le répertoire META-INF/hibernate-objectGridDeployment.xml, la stratégie de déploiement par défaut est remplacée par la configuration dans ce fichier.

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
  xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
  <objectgridDeployment objectgridName="Annuity">
    <mapSet name="MAPSET_Annuity" numberOfPartitions="1" numInitialContainers="1" minSyncReplicas="0"
          maxSyncReplicas="4" maxAsyncReplicas="0" replicaReadEnabled="true">
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity" />
      <map ref="defaultCacheMap" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout" />
      <map ref="org.hibernate.cache.UpdateTimestampsCache" />
      <map ref="org.hibernate.cache.StandardQueryCache" />
    </mapSet>
  </objectgridDeployment>
</deploymentPolicy>
```

Fichier hibernate-objectGrid.xml

Si vous n'utilisez pas Hibernate avec Java Persistence API (JPA), utilisez l'exemple suivant hibernate-objectGrid.xml pour créer votre configuration Hibernate :

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="Annuity">
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity" />
      <backingMap name="defaultCacheMap" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="defaultCacheMap" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout" />
      <backingMap name="org.hibernate.cache.UpdateTimestampsCache" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="org.hibernate.cache.UpdateTimestampsCache" />
      <backingMap name="org.hibernate.cache.StandardQueryCache" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="org.hibernate.cache.StandardQueryCache" />
    </objectGrid>
  </objectGrids>
  <backingMapPluginCollections>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity">
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
```

```
    <backingMapPluginCollection id="defaultCacheMap">
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor">
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact">
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person">
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider">
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout">
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="org.hibernate.cache.UpdateTimestampsCache">
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>

    </backingMapPluginCollection>
    <backingMapPluginCollection id="org.hibernate.cache.StandardQueryCache">
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
  </backingMapPluginCollections>
</objectGridConfig>
```

Remarque : Les mappes org.hibernate.cache.UpdateTimestampsCache, org.hibernate.cache.StandardQueryCache et defaultCacheMap sont requises.

Configuration de l'intégration de base de données

Vous pouvez utiliser WebSphere eXtreme Scale pour réduire la charge sur les bases de données. Vous pouvez utiliser une API de persistance Java (JPA) entre WebSphere eXtreme Scale et la base de données pour intégrer les modifications comme chargeur.

Avant de commencer

Pour le récapitulatif des différentes topologies que vous pouvez créer avec une base de données, voir «Intégration de la base de données : caches avec écriture différée, caches en ligne et caches secondaires», à la page 17.

Configuration des chargeurs JPA

Un chargeur Java Persistence API (JPA) est une implémentation de plug-in qui utilise JPA pour interagir avec la base de données.

Avant de commencer

- Vous devez disposer d'une implémentation JPA, comme Hibernate ou OpenJPA.
- Votre base de données peut correspondre à tout programme d'arrière plan prise en charge par le fournisseur JPA choisi.
- Déterminez si vous allez utiliser le plug-in JPALoader ou JPAEntityLoader. Utilisez le plug-in JPALoader lorsque vous stockez des données à l'aide de l'API ObjectMap. Utilisez le plug-in JPAEntityLoader lorsque vous stockez des données à l'aide de l'API EntityManager.

  Remarque : Si vous utilisez les API JPA pour accéder à la source de données JPA, utilisez le plug-in de mémoire cache L2 JPA. Ce plug-in place la grille de

données entre votre application et les données source JPA, tout en continuant à utiliser une application JPA. Pour plus d'informations, voir «Plug-in de cache niveau 2 (L2) JPA», à la page 331.

Pourquoi et quand exécuter cette tâche

Pour plus d'informations sur le fonctionnement de Java Persistence API (JPA) Loader, voir Chargeurs JPA.

Procédure

1. Configurez les paramètres requis par JPA pour interagir avec une base de données.

   Les paramètres ci-après sont requis. Ces paramètres sont configurés dans le bean JPALoader ou JPAEntityLoader et le bean JPATxCallback.

   - **persistenceUnitName** : Indique le nom de l'unité de persistance. Ce paramètre est requis à deux titres : pour créer une fabrique de gestionnaire d'entités JPA et pour rechercher les métadonnées d'entité JPA dans le fichier `persistence.xml`. Cet attribut est défini dans le bean JPATxCallback.
   - **JPAPropertyFactory** : Indique la fabrique permettant de créer une mappe de propriétés de persistance pour remplacer les propriétés de persistance par défaut. Cet attribut est défini dans le bean JPATxCallback. Pour définir cet attribut, une configuration de style Spring est requise.
   - **entityClassName** : Indique le nom de classe d'entité requis pour utiliser les méthodes JPA (par exemple, EntityManager.persist, EntityManager.find, etc.). Le plug-in JPALoader requiert ce paramètre, mais ce paramètre est facultatif pour JPAEntityLoader. Pour le plug-in JPAEntityLoader, si aucun paramètre **entityClassName** n'est défini, la classe d'entités configurée dans la mappe d'entités d'ObjectGrid est utilisée. Vous devez utiliser le même nom de classe pour le gestionnaire d'entités eXtreme Scale et le fournisseur JPA. Cet attribut est défini dans le bean JPALoader ou JPAEntityLoader.
   - **preloadPartition** : Indique la partition à partir de laquelle le préchargement de la mappe démarre. Si la partition de préchargement est inférieure à zéro ou supérieure au nombre total de partitions moins 1, le préchargement de la mappe n'est pas démarré. La valeur par défaut est -1, ce qui signifie que le préchargement ne démarre pas par défaut. Cet attribut est défini dans le bean JPALoader ou JPAEntityLoader.

   Outre les quatre paramètres JPA devant être définis dans eXtreme Scale, des métadonnées JPA sont utilisées pour extraire la clé des entités JPA. Les métadonnées JPA peuvent être configurées comme annotation ou comme fichier `orm.xml` spécifié dans le fichier `persistence.xml`. Elles ne font pas partie de la configuration d'eXtreme Scale.

2. Configurez les fichiers XML de la configuration JPA.

   Pour configurer un JPALoader ou JPAEntityLoader, voir Plug-ins pour communiquer avec les bases de données.

   Configuration d'un rappel de transaction JPATxCallback avec la configuration du chargeur. L'exemple suivant est un fichier de descripteur XML ObjectGrid (`objectgrid.xml`), qui dispose d'un bean JPAEntityLoader et d'un bean JPATxCallback configurés :

   configuration d'un chargeur avec rappel - Exemple XML

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
    <objectGrids>
```

```
      <objectGrid name="JPAEM" entityMetadataXMLFile="jpaEMD.xml">
        <bean id="TransactionCallback"
           className="com.ibm.websphere.objectgrid.jpa.JPATxCallback">
           <property
              name="persistenceUnitName"
              type="java.lang.String"
              value="employeeEMPU" />
        </bean>
        <backingMap name="Employee" pluginCollectionRef="Employee" />
    </objectGrid>
  </objectGrids>

  <backingMapPluginCollections>
    <backingMapPluginCollection id="Employee">
        <bean id="Loader"
          className="com.ibm.websphere.objectgrid.jpa.JPAEntityLoader">
       <property
               name="entityClassName"
               type="java.lang.String"
               value="com.ibm.ws.objectgrid.jpa.test.entity.Employee"/>
        </bean>
    </backingMapPluginCollection>
  </backingMapPluginCollections>
</objectGridConfig>
```

Si vous souhaitez configurer un JPAPropertyFactory, vous devez utiliser une configuration de type Spring. Voici un exemple de fichier de configuration XML,`JPAEM_spring.xml` qui configure un bean Spring à utiliser pour les configurations eXtreme Scale.

configuration d'un chargeur avec une fabrique de propriétés JPA - Exemple XML

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:aop="http://www.springframework.org/schema/aop"
       xmlns:tx="http://www.springframework.org/schema/tx"
       xmlns:objectgrid="http://www.ibm.com/schema/objectgrid"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
           http://www.springframework.org/schema/beans/spring-beans-2.0.xsd">

  <objectgrid:JPAEntityLoader id="jpaLoader"
  entityClassName="com.ibm.ws.objectgrid.jpa.test.entity.Employee"/>
  <objectgrid:JPATxCallback id="jpaTxCallback" persistenceUnitName="employeeEMPU" />
</beans>
```

Le fichier de configuration XML `Objectgrid.xml` est présenté ci-après. Notez que le nom ObjectGrid est `JPAEM`, qui correspond au nom ObjectGrid dans le fichier de configuration Spring `JPAEM_spring.xml`.

Configuration du chargeur JPAEM - Exemple de XML

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="JPAEM" entityMetadataXMLFile="jpaEMD.xml">
      <bean id="TransactionCallback"
           className="{spring}jpaTxCallback"/>
        <backingMap name="Employee" pluginCollectionRef="Employee"
                    writeBehind="T4"/>
      </objectGrid>
  </objectGrids>

  <backingMapPluginCollections>
    <backingMapPluginCollection id="Employee">
       <bean id="Loader" className="{spring}jpaLoader" />
    </backingMapPluginCollection>
  </backingMapPluginCollections>
</objectGridConfig>
```

Une entité peut être annotée avec les annotations JPA et les annotations du gestionnaire d'entités d'eXtreme Scale. Chaque annotation possède un équivalent XML qui peut être utilisé. eXtreme Scale a donc ajouté l'espace de

noms Spring. Vous pouvez également les configurer à l'aide de la prise en charge de l'espace de noms Spring. Pour plus d'informations, voir Présentation de l'infrastructure Spring.

Configuration d'un programme de mise à jour de données JPA en fonction de la date/heure

Vous pouvez configurer une mise à jour de base de données en fonction de la date/heure à l'aide d'une configuration XML eXtreme Scale locale ou répartie. Vous pouvez également configurer une configuration locale à l'aide d'un programme.

Pourquoi et quand exécuter cette tâche

Pour plus d'information sur le fonctionnement du programme de mise à jour en fonction de la date/heure Java Persistence API (JPA), voir Programme de mise à jour de données JPA en fonction de la date/heure.

Procédure

Créez une configuration timeBasedDBUpdate.

- **Avec un fichier XML :**

  L'exemple suivant illustre un fichier `objectgrid.xml` qui contient une configuration timeBasedDBUpdate :

```
Programme de mise à jour
JPA en fonction de la date/heure - Exemple de XML
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
    <objectGrids>
        <objectGrid name="changeOG"
            entityMetadataXMLFile="userEMD.xml">
            <backingMap name="user" >
                <timeBasedDBUpdate timestampField="rowChgTs"
                      persistenceUnitName="userderby"
                      entityClass="com.test.UserClass"
                      mode="INVALIDATE_ONLY"
                   />
            </backingMap>
        </objectGrid>
    </objectGrids>
    <backingMapPluginCollections>
</objectGridConfig>
```

  Dans cet exemple, la mappe "user" est configurée avec une mise à jour de base de données en fonction de la date/heure. Le mode de mise à jour de base de données est INVALIDATE_ONLY et le champ d'horodatage possède la valeur rowChgTs.

  Si l'ObjectGrid réparti "changeOG" est démarré sur le serveur conteneur, une unité d'exécution de mise à jour de base de données en fonction de la date/heure est automatiquement démarrée dans la partition 0.

- **A l'aide d'un programme :**

  Si vous créez un ObjectGrid local, vous pouvez également créer un objet TimeBasedDBUpdateConfig et le définir sur l'instance BackingMap :

```
public void setTimeBasedDBUpdateConfig(TimeBasedDBUpdateConfig dbUpdateConfig);
```

  Pour plus d'informations sur la définition d'un objet sur l'instance BackingMap, voir les informations sur l'interface BackingMap dans la documentation de l'API.

  Vous pouvez également annoter le champ d'horodatage dans la classe d'entité à l'aide de l'annotation com.ibm.websphere.objectgrid.jpa.dbupdate.annotation.Timestamp. En

configurant la valeur dans la classe, vous n'avez pas besoin de configurer le champ d'horodatage dans la configuration XML.

Que faire ensuite

Démarrez le programme de mise à jour de base de données JPA en fonction de la date/heure. Pour plus d'informations, voir Démarrage du programme de mise à jour en fonction de la date/heure.

Configuration des services de données REST

Vous pouvez utiliser le service de données RESTWebSphere eXtreme Scale WebSphere Application Server, WebSphere Application Server Community Edition et Apache Tomcat.

Pourquoi et quand exécuter cette tâche

L'exemple fourni comprend le code source et les fichiers binaires compilés permettant d'exécuter une grille de données partitionnées. Cet exemple montre comment créer une grille de données simple, modéliser les données en utilisant des entités et fournit deux applications client de ligne de commande qui permettent d'ajouter et d'interroger des entités en utilisant Java ou C#.

L'exemple de client Java utilise L'API Java EntityManager pour conserver et interroger dans la grille de données. Ce client peut être exécuté dans Eclipse ou à l'aide d'un script de ligne de commande. Notez que l'exemple de client Java n'illustre pas le service de données REST, mais permet de mettre à jour les données dans la grille de façon à ce qu'un navigateur Web ou un autre client puisse lire les données.

L'exemple de client Microsoft WCF Data Services C# communique avec la grille de données eXtreme Scale via le service de données REST à l'aide de l'infrastructure .NET. Le client WCF Data Services peut être utilisé à la fois pour mettre à jour et interroger la grille de données.

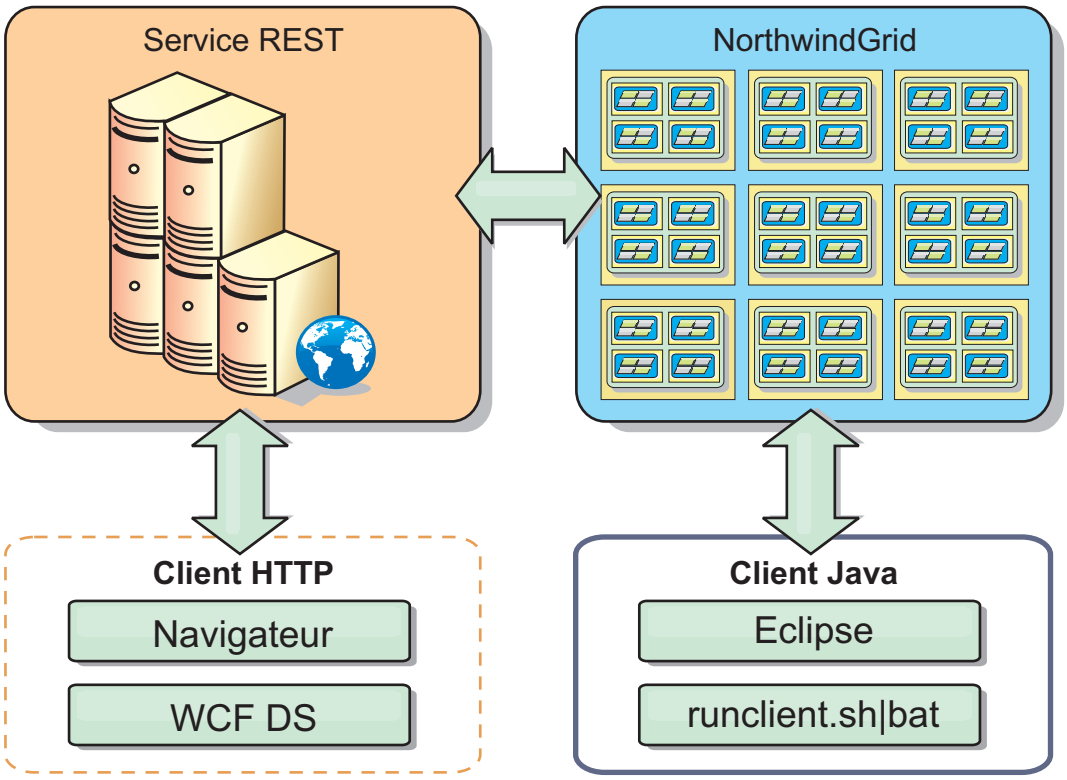


Figure 43. Exemple Mise en route de topologie. Les clients HTTP en utilisant le service de données REST et les clients Java peuvent accéder à la même grille de données.

Procédure

1. Configurez et démarrez une grille de données eXtreme Scale. Voir «Activation du service de données REST».
2. Configurez et démarrez le service de données REST dans un navigateur Web. Voir «Configuration de serveurs d'applications pour le service de données REST», à la page 367.
3. Exécutez un client pour interagir avec le service de données REST. Deux options sont disponibles :
   a. Exécutez l'exemple de client Java pour remplir la grille avec les données en utilisant l'API EntityManager et interrogez les données de la grille par le biais d'un navigateur Web et du service de données REST d'eXtreme Scale. Voir «Utilisation d'un client Java avec les services de données REST», à la page 384.
   b. Exécutez l'exemple de client WCF Data Services C#. Voir «Client WCF de Visual Studio 2008 avec le service de données REST», à la page 386.

Activation du service de données REST

Le service de données REST peut représenter les métadonnées d'entités WebSphere eXtreme Scale pour représenter chaque entité sous la forme d'un EntitySet.

Démarrage d'une exemple de grille de données eXtreme Scale

En général, avant de lancer le service de données REST, démarrez la grille de données eXtreme Scale. La procédure qui suit va démarrer un processus de service de catalogue eXtreme Scale et deux processus de serveurs conteneurs.

WebSphere eXtreme Scale peut être installé selon trois méthodes différentes :

- installation d'essai
- déploiement autonome
- déploiement intégré à WebSphere Application Server

Evolutivité du modèle de données dans eXtreme Scale

L'exemple Microsoft Northwind utilise la table OrderDetail pour établir une association plusieurs-à-plusieurs entre les commandes et les produits.

Les spécifications ORM (Object to relational mapping) comme l'ADO.NET Entity Framework et JPA (Java Persistence API) peuvent mapper les tables et les relations à l'aide d'entités. Mais cette architecture n'est pas évolutive. Pour bien fonctionner, tout doit se trouver sur la même machine ou sur un cluster coûteux de machines.

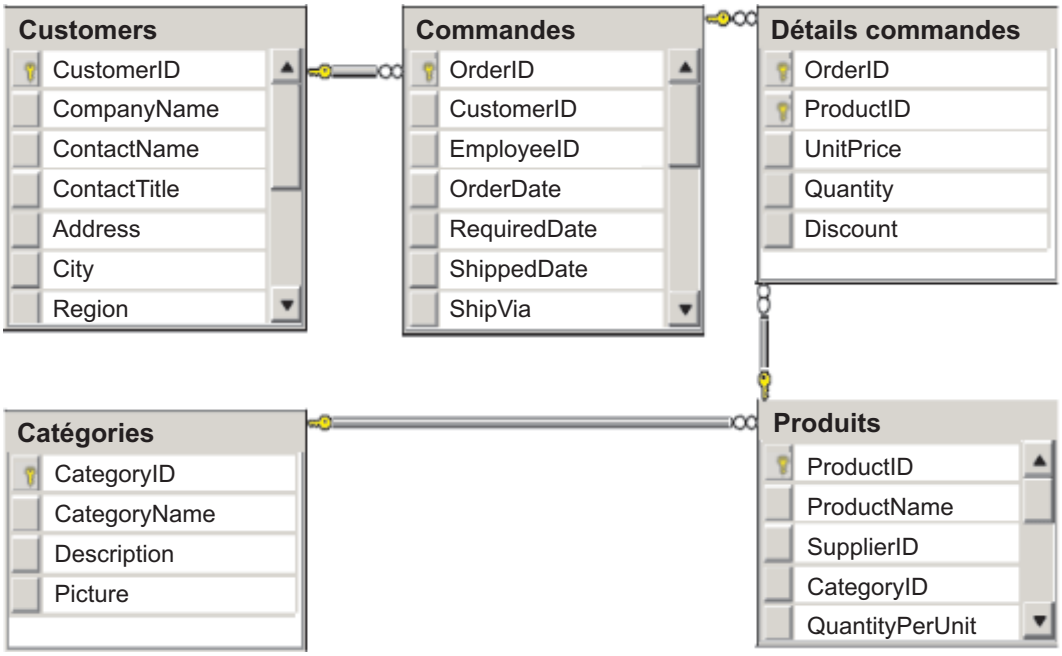


Figure 44. Schéma de l'exemple Microsoft SQL Server Northwind

Pour que puisse être créée une version évolutive de l'exemple, les entités doivent être modélisées de manière à ce que chaque entité ou chaque groupe d'entités en rapport puissent être partitionnées à partir d'une seule clé. De ce fait, les demandes peuvent être réparties entre plusieurs serveurs indépendants. Pour y arriver, les entités ont été divisées en deux arborescences : l'arborescence Customer et l'arborescence Product. Dans ce modèle, chaque arborescence peut être partitionnée de manière indépendante et peut donc croître à des rythmes différents, d'où une plus grande évolutivité.

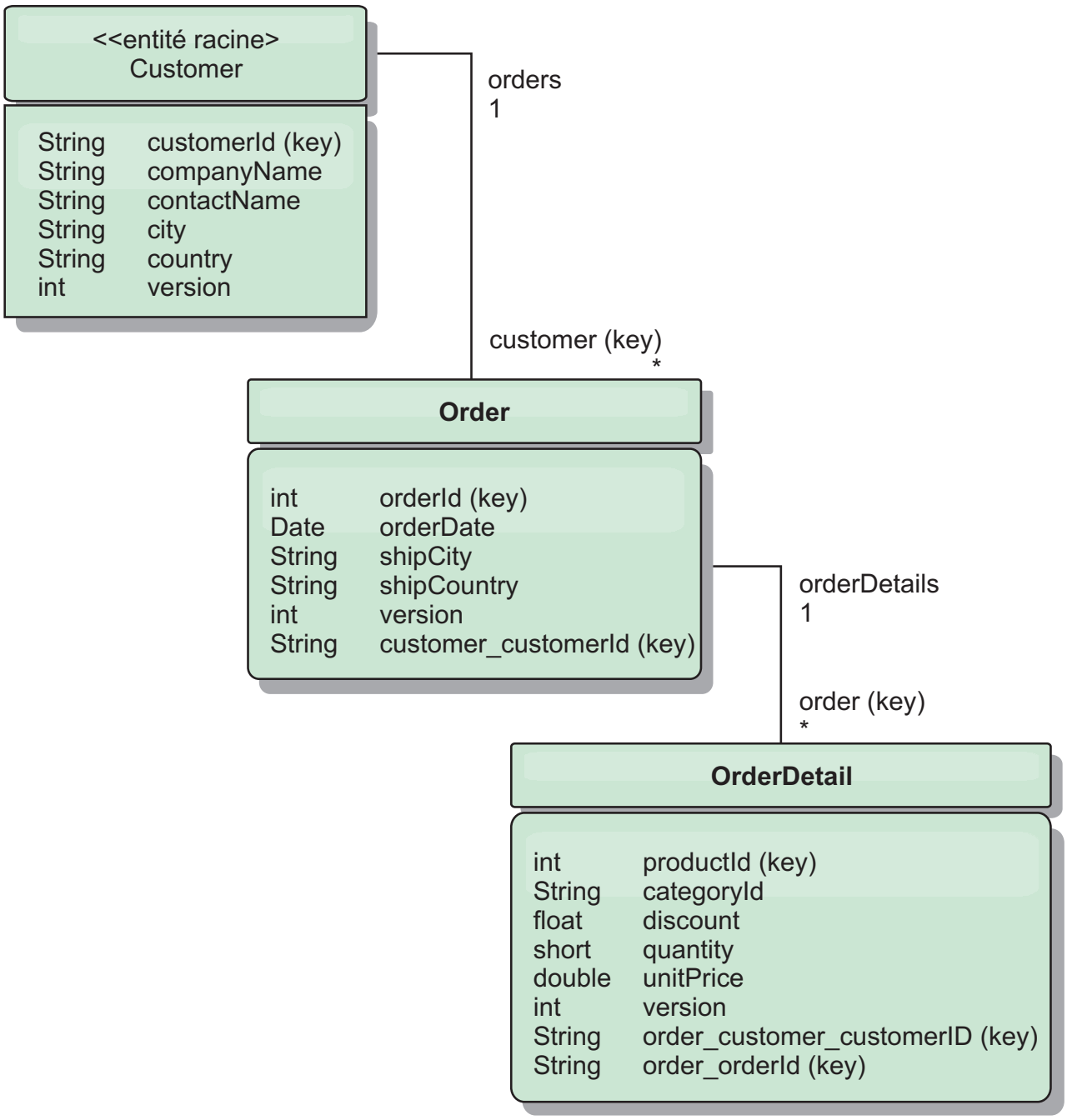


Figure 45. Schéma des entités Customer et Order

Par exemple, Order et Product ont tous les deux comme clés des entiers distincts et uniques. Et de fait, la table Order et la table Product sont deux tables réellement indépendantes l'une de l'autre. Par exemple, considérez l'effet de la taille d'un catalogue, le nombre de produits que vous vendez, avec le nombre total de commandes. A première vue, il peut sembler qu'avoir un grand nombre de produits implique d'avoir également un grand nombre de commandes, mais ce n'est pas nécessairement le cas. Si c'était vrai, il suffirait d'ajouter des produits au catalogue pour augmenter les ventes. Les commandes et les produits ont leurs propres tables indépendantes. L'on peut étendre ce concept en imaginant que les commandes et les produits aient chacun leurs propres grilles de données. Avec des grilles de données indépendantes, vous pouvez contrôler le nombre de partitions et

de serveurs, en plus de la taille de chaque grille de données séparément afin que votre application puisse évoluer. Si vous doublez la taille du catalogue, vous devez doubler la grille de données de produits, mais la grille des commandes peut rester telle quelle. L'inverse est vrai pour un afflux de commandes.

Dans le schéma, un client a zéro ou plusieurs commandes et une commande a des articles (OrderDetail), chacun avec un produit spécifique. Un produit est identifié par un ID (la clé Product) dans chaque OrderDetail. Une grille de données unique stocke les clients, les commandes et les détails des commande, Customer étant l'entité racine de la grille de données. L'on peut extraire les clients à partir de leur ID, mais l'on doit faire partir les commandes des ID clients. L'ID client est donc ajouté à la commande comme partie de sa clé. De la même manière, l'ID client et l'ID commande font partie de l'ID du détail de commande.

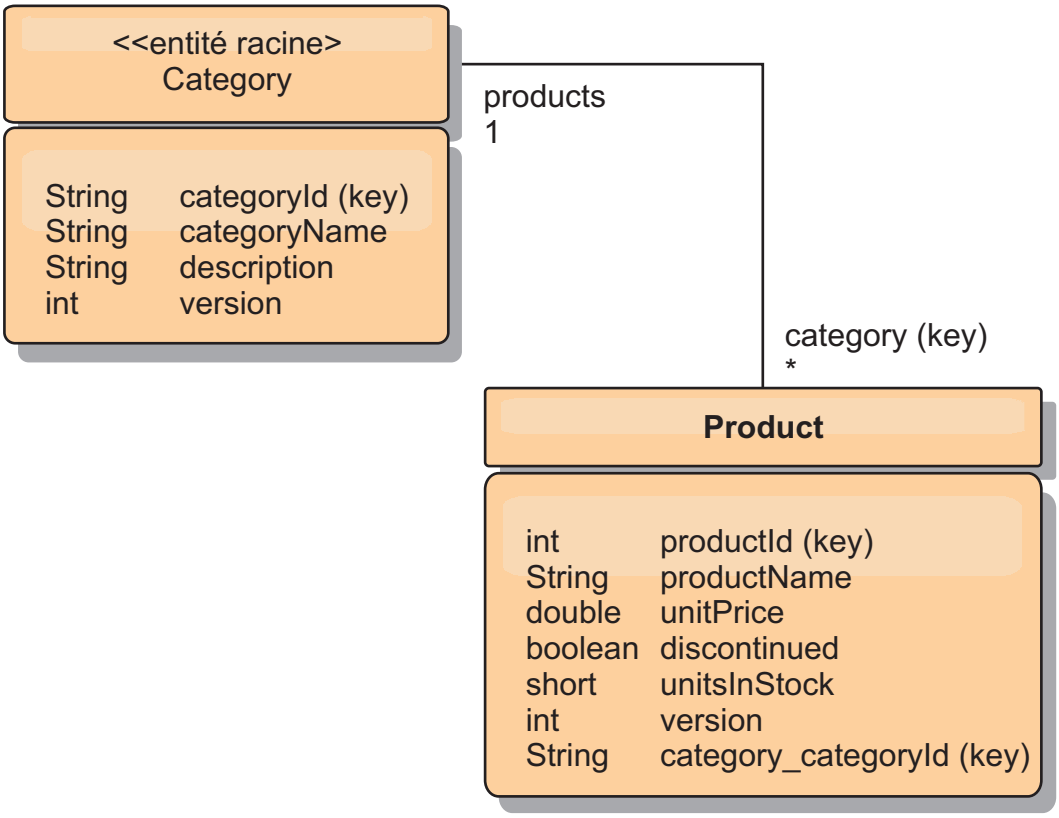


Figure 46. Schéma des entités Category et Product

Dans le schéma Category et Product, Category est la racine du schéma. Avec ce schéma, les clients peuvent rechercher des produits à partir de leur catégorie. Voir «Extraction et actualisation des données avec REST» pour d'autres détails sur les associations de clés et leur importance.

Extraction et actualisation des données avec REST

Le protocole OData requiert que toutes les entités soient adressables à partir de leur forme canonique. Cela signifie que chaque entité doit inclure la clé de l'entité racine partitionnée, la racine de schéma.

Voici un exemple de la manière d'utiliser l'association à partir d'une entité racine pour définir l'adresse d'un enfant dans :

```
/Customer('ACME')/order(100)
```

Dans WCF Data Services, l'entité enfant doit être directement adressable, c'est-à-dire que la clé dans la racine du schéma doit faire partie de la clé de l'enfant : `/Order(customer_customerId='ACME', orderId=100)`. L'on y parvient en créant une association à l'entité racine dans laquelle l'association un-à-un ou plusieurs-à-un à l'entité racine est également identifiée comme une clé. Lorsque des entités sont incluses comme faisant partie de la clé, les attributs de l'entité parent sont exposés comme propriétés de la clé.

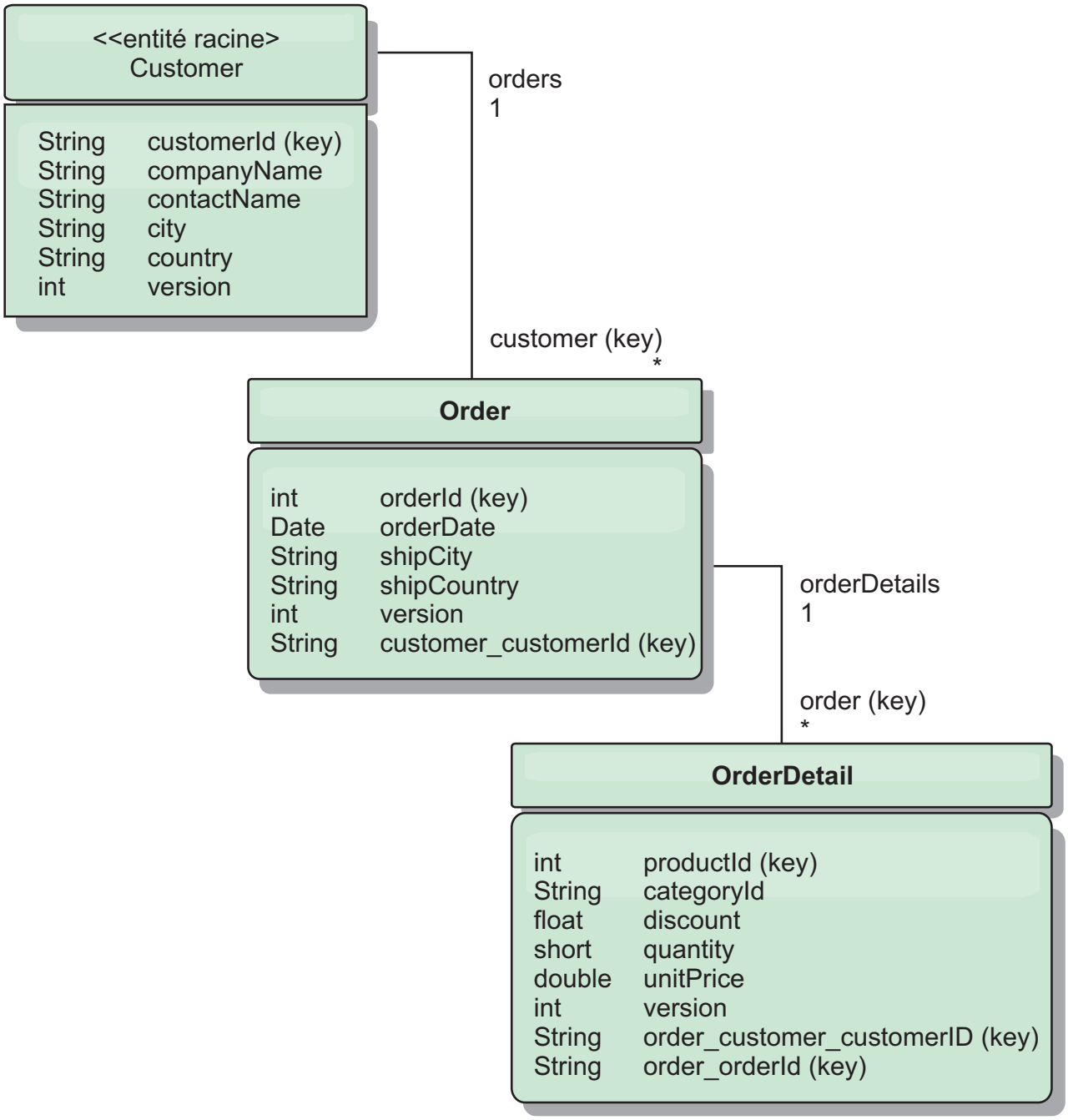


Figure 47. Schéma des entités Customer et Order

Le schéma des entités Customer/Order illustre la manière dont chaque entité est partitionnée à l'aide de Customer. L'entité Order inclut Customer comme partie de sa clé et elle est donc directement accessible. Le service de données REST expose

toutes les associations de clés comme des propriétés individuelles : Order a customer_customerId et OrderDetail a order_customer_customerId and order_orderId.

L'API EntityManager permet de trouver la commande avec l'ID du client et celui de la commande :

```
transaction.begin();
// L'on recherche l'Order à l'aide du Customer.  Nous n'incluons que l'Id
// dans la classe Customer lorsqu'on génère l'instance de clé OrderId.
Order order = (Order) em.find(Order.class,
    new OrderId(100, new Customer('ACME')));
...
transaction.commit();
```

Lorsqu'on utilise le service de données REST, la commande peut être extraite avec l'une des URL :

- `/Order(orderId=100, customer_customerId='ACME')`
- `/Customer('ACME')/orders?$filter=orderId eq 100`

L'adresse de la clé Customer est créée avec l'attribut name de l'entité Customer, un caractère de soulignement et l'attribut name de l'ID Customer : customer_customerId.

Une entité peut également inclure une entité non racine comme faisant partie de sa clé si tous les ancêtres de cette entité non racine ont des associations de clés à la racine. Dans notre exemple, OrderDetail a une association de clés à Order et Order a une association de clés à l'entité Customer racine. Avec l'API EntityManager :

```
transaction.begin();
// L'on construit une instance de clé OrderDetailId.  Elle inclut
// Order et Customer avec uniquement le jeu de clés.
Customer customerACME = new Customer("ACME");
Order order100 = new Order(100, customerACME);
OrderDetailId orderDetailKey =
    new OrderDetailId(order100, "COMP");
OrderDetail orderDetail = (OrderDetail)
    em.find(OrderDetail.class, orderDetailKey);
...
```

Le service de données REST permet l'adressage direct d'OrderDetail :

```
/OrderDetail(productId=500, order_customer_customerId='ACME', order_orderId =100)
```

L'association partant de l'entité OrderDetail vers l'entité Product a été rompue pour permettre le partitionnement indépendant des commandes et du stock de produits. L'entité OrderDetail stocke la catégorie et l'ID de produit au lieu d'une relation en dur. En découpant les deux schémas d'entités, il n'est accédé qu'à une seule partition à la fois.

Le schéma Category et Product montre que l'entité racine est Category et que chaque Product a une association à une entité Category. L'entité Category est incluse dans l'identité Product. Le service de données REST expose une propriété de clé : category_categoryId qui permet l'adressage direct de Product.

Category étant l'entité racine, dans un environnement partitionné, Category doit être connu pour que Product puisse être trouvé. Avec l'API EntityManager, la transaction doit être fixée à l'entité Category avant toute recherche de Product.

Avec l'API EntityManager :

```
transaction.begin();
// L'on crée l'entité racine Category avec uniquement la clé.  Cela
// nous permet de construire un ProductId sans avoir besoin de trouver
// d'abord la Category.  La transaction est à présent fixée
// à la partition où est stockée la Category "COMP".
Category cat = new Category("COMP");
Product product = (Product) em.find(Product.class,
    new ProductId(500, cat));
...
```

Le service de données REST permet l'adressage direct de Product :

```
/Product(productId=500, category_categoryId='COMP')
```

Démarrage d'une grille de données autonome pour les services de données REST

Suivez ces étapes pour démarrer l'exemple de grille de données du service REST WebSphere eXtreme Scale pour un déploiement eXtreme Scale autonome.

Avant de commencer

Installez la version d'évaluation de WebSphere eXtreme Scale ou le produit complet :

- Installez la version autonome du produit et appliquez tous les correctifs ultérieurs.
- Téléchargez et extrayez la version d'évaluation WebSphere eXtreme Scale 7.1 qui inclut le service de données REST WebSphere eXtreme Scale.

Pourquoi et quand exécuter cette tâche

Démarrez l'exemple de grille de données WebSphere eXtreme Scale.

Procédure

1. Démarrez le processus de service de catalogue. Ouvrez une ligne de commande ou une fenêtre de terminal, puis définissez la variable d'environnement JAVA_HOME :
   - Linux UNIX `export JAVA_HOME=base_java`
   - Windows `set JAVA_HOME=base_java`
2. `cd base_servicerest/gettingstarted`
3. Démarrez le processus de service de catalogue. Pour démarrer le service *sans* la sécurité d'eXtreme Scale, utilisez les commandes suivantes :
   - Linux UNIX `./runcat.sh`
   - Windows `runcat.bat`

   Pour démarrer le service avec la sécurité d'eXtreme Scale, utilisez les commandes suivantes :
   - Linux UNIX `./runcat_secure.sh`
   - Windows `runcat_secure.bat`
4. Démarrez deux processus de serveur conteneur. Ouvrez une autre ligne de commande ou fenêtre de terminal, puis définissez la variable d'environnement JAVA_HOME :
   - Linux UNIX `export JAVA_HOME=base_java`
   - Windows `set JAVA_HOME=base_java`

5. `cd base_servicerest/gettingstarted`
6. Démarrez un processus de serveur de conteneur :

   Pour démarrer le serveur sans la sécurité d'eXtreme Scale, utilisez les commandes suivantes :

   - Linux UNIX `./runcontainer.sh container0`
   - Windows `runcontainer.bat container0`

   Pour démarrer le serveur avec la sécurité d'eXtreme Scale, utilisez les commandes suivantes :

   - Linux UNIX `./runcontainer_secure.sh container0`
   - Windows `runcontainer_secure.bat container0`
7. Ouvrez une autre ligne de commande ou fenêtre de terminal, puis définissez la variable d'environnement JAVA_HOME :
   - Linux UNIX `export JAVA_HOME=`*`base_java`*
   - Windows `set JAVA_HOME=`*`base_java`*
8. `cd base_servicerest/gettingstarted`
9. Démarrez un second processus de serveur conteneur.

   Pour démarrer le serveur sans la sécurité d'eXtreme Scale, utilisez les commandes suivantes.

   - Linux UNIX `./runcontainer.sh container1`
   - Windows `runcontainer.bat container1`

   Pour démarrer le serveur avec la sécurité d'eXtreme Scale, utilisez les commandes suivantes.

   - Linux UNIX `./runcontainer_secure.sh container1`
   - Windows `runcontainer_secure.bat container1`

Résultats

Attendez que les conteneurs eXtreme Scale soient prêts avant de passer aux étapes suivantes. Les serveurs conteneurs sont prêts lorsque le message suivant s'affiche dans la fenêtre de terminal :

`CWOBJ1001I: Le serveur ObjectGrid `*`nom_conteneur`*` est prêt à traiter les requêtes.`

où *nom_conteneur* correspond au nom du conteneur qui a été démarré.

Démarrage d'une grille de données pour les services de données REST dans WebSphere Application Server

Appliquez la procédure exposée ici pour démarrer une grille de données exemple de service REST WebSphere eXtreme Scale autonome pour un déploiement de WebSphere eXtreme Scale intégré à WebSphere Application Server. Bien que WebSphere eXtreme Scale soit intégré à WebSphere Application Server, cette procédure démarre un processus de service de catalogue WebSphere eXtreme Scale autonome et un conteneur.

Avant de commencer

Installez le produit dans un répertoire d'installation WebSphere Application Server Version 7.0.0.5 ou suivante avec la sécurité activée. Etendez au moins un profil de serveur d'applications.

Pourquoi et quand exécuter cette tâche

Démarrage de l'exemple de grille de données WebSphere eXtreme Scale.

Procédure

1. Démarrez le processus de service de catalogue. Ouvrez une fenêtre de ligne de commande ou de terminal et définissez la variable d'environnement JAVA_HOME :
   - Linux UNIX `export JAVA_HOME=base_java`
   - Windows `set JAVA_HOME=base_java`

   `cd base_servicerest/gettingstarted`
2. Démarrez le processus de service de catalogue.

   Pour démarrer le serveur sans la sécurité d'eXtreme Scale, utilisez les commandes suivantes.
   - Linux UNIX `./runcat.sh`
   - Windows `runcat.bat`

   Pour démarrer le serveur avec la sécurité d'eXtreme Scale, utilisez les commandes suivantes :
   - Linux UNIX `./runcat_secure.sh`
   - Windows `runcat_secure.bat`
3. Démarrez deux processus de serveur conteneur. Ouvrez une autre fenêtre de ligne de commande ou de terminal et définissez la variable d'environnement JAVA_HOME :
   - Linux UNIX `export JAVA_HOME=base_java`
   - Windows `set JAVA_HOME=base_java`
4. Démarrez un processus de serveur conteneur.

   Pour démarrer le serveur sans la sécurité d'eXtreme Scale, utilisez les commandes suivantes.
   a. Ouvrez une fenêtre de ligne de commande.
   b. `cd base_servicerest/gettingstarted`
   c. Pour démarrer le serveur *sans* la sécurité d'eXtreme Scale, utilisez les commandes suivantes :
      - Linux UNIX `./runcontainer.sh container0`
      - Windows `runcontainer.bat container0`
   d. Pour démarrer le serveur avec la sécurité d'eXtreme Scale, utilisez les commandes suivantes.
      - Linux UNIX `./runcontainer_secure.sh container0`
      - Windows `runcontainer_secure.bat container0`
5. Démarrez un second processus de serveur conteneur.
   a. Ouvrez une fenêtre de ligne de commande.

b. `cd base_servicerest/gettingstarted`

c. Pour démarrer le serveur *sans* la sécurité d'eXtreme Scale, utilisez les commandes suivantes :

- Linux UNIX `./runcontainer.sh container1`
- Windows `runcontainer.bat container1`

d. Pour démarrer le serveur *avec* la sécurité d'eXtreme Scale, utilisez les commandes suivantes :

- Linux UNIX `./runcontainer_secure.sh container1`
- Windows `runcontainer_secure.bat container1`

Résultats

Attendez que les serveurs conteneurs soient prêts avant de passer aux étapes suivantes. Les serveurs conteneurs sont prêts lorsque le message suivant s'affiche :

`CWOBJ1001I: Le serveur ObjectGrid` *`nom_conteneur`* `est prêt à traiter les requêtes.`

où *nom_conteneur* correspond au nom du conteneur qui a été démarré à l'étape précédente.

Configuration de serveurs d'applications pour le service de données REST

Vous pouvez configurer des serveurs d'applications différents pour utiliser le service de données REST.

Déploiement du service de données REST sur WebSphere Application Server

Cette rubrique explique comment configurer le service de données REST WebSphere eXtreme Scale sur WebSphere Application Server ou WebSphere Application Server Network Deployment Version 6.1.0.25 et les versions suivantes. Ces instructions s'appliquent également aux déploiements où WebSphere eXtreme Scale est intégré avec le déploiement WebSphere Application Server.

Avant de commencer

Vous devez disposer de l'un des environnements suivants sur votre système pour configurer et déployer le service de données REST pour WebSphere eXtreme Scale.

- WebSphere Application Server avec le client autonome WebSphere eXtreme Scale :
  - WebSphere eXtreme Scale Trial Version 7.1 avec le service de données REST est téléchargé et extrait ou WebSphere eXtreme Scale Version 7.1.0.0 avec le groupe de correctif 2 est installé dans un répertoire autonome.
  - WebSphere Application Server Version 6.1.0.25, 7.0.0.5 ou version suivante est installé et actif.
- WebSphere Application Server intégré à WebSphere eXtreme Scale:

  WebSphere eXtreme Scale Version 7.1.0.0 avec le groupe de correctifs 2 ou suivant installé sur WebSphere Application Server Version 6.1.0.25, 7.0 ou version suivante.

Conseil : Le service de données REST d'WebSphere eXtreme Scale requiert uniquement que le client WebSphere eXtreme Scale soit installé. Il n'est pas nécessaire d'étendre le profil.

Découvrez comment activer la sécurité Java 2 dans le centre de documentation WebSphere Application Server.

Procédure

1. Configurez et démarrez une grille de données.
   a. Pour plus de détails sur la configuration d'une grille de données pour l'utiliser avec le service de données REST, voir «Démarrage d'une grille de données pour les services de données REST dans WebSphere Application Server», à la page 365.
   b. Vérifiez qu'un client peut se connecter et accéder aux entités dans la grille de données. Pour un exemple, voir «Tutoriel : Démarrer avec WebSphere eXtreme Scale», à la page 1.
2. Générez le répertoire ou le fichier JAR de configuration du service REST d'eXtreme Scale. Reportez-vous dans «Installation du service de données REST» , à la page 200 aux explications sur le packaging et le déploiement du service REST.
3. Ajoutez le fichier JAR ou le répertoire de la configuration du service de données REST au chemin d'accès aux classes du serveur d'applications :
   a. Ouvrez la console d'administration de WebSphere Application Server.
   b. Accédez à **Environnement** > **Bibliothèques partagées**
   c. Cliquez sur **Nouveau**
   d. Ajoutez les entrées suivantes dans les zones appropriées :
      - Nom : `extremescale_rest _configuration`
      - Chemin de classes : <répertoire ou fichier JAR de configuration du service REST>
   e. Cliquez sur **OK**
   f. Sauvegardez les modifications apportées à la configuration principale
4. Ajoutez le fichier JAR d'exécution client WebSphere eXtreme Scale, `wsogclient.jar` et le fichier JAR fichier JAR du service de données REST ou le répertoire d'accès au chemin d'accès aux classes du serveur. Cette étape n'est pas nécessaire si WebSphere eXtreme Scale est intégré à l'installation WebSphere Application Server.
   a. Ouvrez la console d'administration de WebSphere Application Server.
   b. Accédez à **Environnement** > **Bibliothèques partagées**.
   c. Cliquez sur **Nouveau**.
   d. Ajoutez les entrées suivantes dans les zones :
      - Nom : `extremescale_client_v71`
      - Chemin d'accès aux classes : `base_wxs/lib/wsogclient.jar`

      A faire : Ajoutez chaque chemin sur une ligne séparée.
   e. Cliquez sur **OK**.
   f. Sauvegardez les modifications apportées à la configuration principale.
5. Installez le fichier EAR du service de données REST (`wxsrestservice.ear`) sur WebSphere Application Server à partir de la console d'administration :
   a. Ouvrez la console d'administration de WebSphere Application Server.
   b. Cliquez sur **Applications** > **Nouvelle application**.

c. Accédez au fichier /lib/wxsrestservice.ear dans le système de fichiers, sélectionnez-le et cliquez sur **Suivant**.
   - Si vous utilisez WebSphere Application Server Version 7.0, cliquez sur suivant.
   - Si vous utilisez WebSphere Application Server Version 6.1, entrez un valeur de racine de contexte avec le nom /wxsrestservice et passez à l'étape suivante.

d. Choisissez l'option d'installation détaillée et cliquez sur **Suivant**.

e. Dans l'écran d'avertissements de sécurité de l'application, cliquez sur **Continuer**.

f. Sélectionnez les options d'installation par défaut, puis cliquez sur **Suivant**.

g. Choisissez un serveur pour y associer l'application et cliquez sur **Suivant**.

h. Sur la page de rechargement JSP, utilisez les valeurs par défaut, puis cliquez sur **Suivant**.

i. Dans la page des bibliothèques partagées, associez le module wxsrestservice.war aux bibliothèques partagées que vous avez définies :
   - extremescale_rest_configuration
   - extremescale_client_v71

   Conseil : Cette bibliothèque partagée est nécessaire uniquement si WebSphere eXtreme Scale n'est pas intégré à WebSphere Application Server.

j. Sur la page de mappe des relations de bibliothèques partagées, utilisez les valeurs par défaut, puis cliquez sur **Suivant**.

k. Sur la page de mappe des hôtes virtuels, utilisez les valeurs par défaut, puis cliquez sur **Suivant**.

l. Dans la page des racines de contexte des mappes, spécifiez wxsrestservice comme racine de contexte.

m. Sur l'écran récapitulatif, cliquez sur **Fin** pour terminer la installation.

n. Sauvegardez les modifications apportées à la configuration principale.

6. Démarrez l'application de service de données wxsrestservice REST :
   a. Accédez à l'application dans la console d'administration.
      - WebSphere Application Server Version 7.0 : Dans la console d'administration, cliquez sur **Applications** > **Types d'application** > **Applications WebSphere**.
      - WebSphere Application Server Version 6.1 : Dans la console d'administration, cliquez sur **Applications** > **Applications d'entreprise**.

   b. Cochez la case de l'application wxsrestservice et cliquez sur **Démarrer**.

   c. Vérifiez le fichier SystemOut.log du profil de serveur d'applications. Lorsque le service de données REST a démarré, le message suivant apparaît dans le journal SystemOut.log du profil du serveur :

      ```
      CWOBJ4000I: Le service de données REST de WebSphere eXtreme Scale a été démarré.
      ```

7. Vérifiez que le service de données REST fonctionne : Le numéro de port se trouve dans le fichier SystemOut.log dans le répertoire des journaux du profil du serveur d'applications et vous pouvez le trouver en recherchant le premier port affiché pour l'identificateur de message : SRVE0250I. Le port par défaut est 9080.

   Par exemple : http://localhost:9080/wxsrestservice/restservice/NorthwindGrid/ Résultat : le document du service AtomPub s'affiche.

Par exemple : `http://localhost:9080/wxsrestservice/restservice/NorthwindGrid/$metadata`. Le document Entity Model Data Extensions (EDMX) s'affiche.

8. Pour arrêter les processus de grille de données, utilisez `CTRL+C` dans la fenêtre de commande correspondante.

Démarrage de services de données REST avec WebSphere eXtreme Scale intégré à WebSphere Application Server 7.0 :

Cette rubrique explique comment configurer et démarrer le service de données REST d'eXtreme Scale à l'aide de WebSphere Application Server version 7.0 qui a été intégré et étendu avec WebSphere eXtreme Scale.

Avant de commencer

Vérifiez que l'exemple de grille de données autonome eXtreme Scale est démarré. Voir «Activation du service de données REST», à la page 358 pour plus d'informations sur le démarrage de la grille.

Pourquoi et quand exécuter cette tâche

Pour commencer avec le service de données REST de WebSphere eXtreme Scale REST à l'aide de WebSphere Application Server, procédez comme suit :

Procédure

1. Ajoutez au chemin d'accès aux classes le fichier JAR d'exemple de configuration du service de données REST de WebSphere eXtreme Scale :
   a. Ouvrez la console d'administration WebSphere
   b. Accédez à Environnement -> Bibliothèques partagées
   c. Cliquez sur Nouveau
   d. Ajoutez les entrées suivantes dans les champs appropriés :
      1) Nom : extremescale_gettingstarted_config
      2) Chemin d'accès aux classes
         - `base_servicerest/gettingstarted/restclient/bin`
         - `base_servicerest/gettingstarted/common/bin`

         A faire : Chaque chemin doit se trouver sur une ligne distincte.
   e. Cliquez sur **OK**
   f. Enregistrez les modifications apportées à la configuration principale.
2. Installez le fichier d'archive d'entreprise du service de données REST sur le serveur à l'aide de la console d'administration WebSphere :
   a. Ouvrez la console d'administration WebSphere
   b. Accédez à Applications -> Nouvelle application
   c. Allez au fichier `base_servicerest/lib/wxsrestservice.ear`. Sélectionnez le fichier, puis cliquez sur **Suivant**.
   d. Sélectionnez les options d'installation détaillées, puis cliquez sur **Suivant**.
   e. Dans l'écran d'avertissements de sécurité de l'application, cliquez sur **Continuer**.
   f. Sélectionnez les options d'installation par défaut, puis cliquez sur **Suivant**.
   g. Choisissez un serveur sur lequel mapper le module wxsrestservice.war, puis cliquez sur **Suivant**.

h. Sur la page de rechargement JSP, utilisez les valeurs par défaut, puis cliquez sur **Suivant**.

i. Sur la page des bibliothèques partagées, mappez le module wxsrestservice.war" sur les bibliothèques partagées suivantes, à savoir celles définies au cours de la première étape : `extremescale_ gettingstarted _config`

j. Sur la page de mappe des relations de bibliothèques partagées, utilisez les valeurs par défaut, puis cliquez sur **Suivant**.

k. Sur la page de mappe des hôtes virtuels, utilisez les valeurs par défaut, puis cliquez sur **Suivant**.

l. Sur la page de mappe de racine de contexte, définissez la racine de contexte sur : wxsrestservice.

m. Sur l'écran récapitulatif, cliquez sur **Fin** pour terminer la installation.

n. Enregistrez les modifications apportées à la configuration principale.

3. Si la grille de données eXtreme Scale a été démarrée avec la sécurité eXtreme Scale activée, définissez la propriété suivante dans le fichier `restservice_home/gettingstarted/restclient/bin/wxsRestService.properties`.

```
ogClientPropertyFile=base_servicerest/gettingstarted/security/security.ogclient.properties
```

4. Démarrez le serveur d'applications et l'application wxsrestservice du service de données REST d'eXtreme Scale.

   Une fois que l'application a démarré, ouvrez le journal SystemOut.log du serveur d'applications et vérifiez que le message suivant est présent : `CWOBJ4000I: Le service de données REST de WebSphere eXtreme Scale a été démarré.`

5. Vérifiez que le service de données REST fonctionne :

   a. Ouvrez un navigateur et rendez-vous à l'adresse suivante :

      http://localhost:9080/wxsrestservice/restservice/NorthwindGrid

      Le document de service de la NorthwindGrid s'affiche.

   b. Rendez-vous à l'adresse suivante :

      http://localhost:9080/wxsrestservice/restservice/NorthwindGrid/$metadata

      Le document Entity Model Data Extensions (EDMX) s'affiche

6. Pour arrêter les processus de grille de données, utilisez CTRL+C dans la fenêtre de commande correspondante.

Déploiement du service de données REST sur WebSphere Application Server Community Edition

Vous pouvez configurer le service de données REST eXtreme Scale sur WebSphere Application Server Community Edition Version 2.1.1.3 et les versions suivantes.

Avant de commencer

- Un JRE ou JDK IBM (recommandé) ou Sun, version 5 ou ultérieure, est installé et la variable d'environnement JAVA_HOME est définie.
- Téléchargez et installez WebSphere Application Server Community Edition version 2.1.1.3 ou plus récente dans le répertoire `racine_wasce`, par exemple le répertoire `/opt/IBM/wasce`. Pour plus d'informations sur version 2.1.1 ou (autres versions), lisez les instructions d'installation.

Procédure

1. Configurez et démarrez une grille de données.

a. Pour plus d'informations sur la configuration d'une grille de données eXtreme Scale pour l'utiliser avec le service de données REST, voir «Démarrage d'une grille de données autonome pour les services de données REST», à la page 364.

b. Vérifiez qu'un client eXtreme Scale parvient à se connecter aux entités de la grille et à y accéder. Pour un exemple, voir «Tutoriel : Démarrer avec WebSphere eXtreme Scale», à la page 1.

2. Générez le répertoire ou le fichier JAR de configuration du service REST d'eXtreme Scale. Pour les détails, voir dans «Installation du service de données REST», à la page 200 les explications sur le packaging et le déploiement.

3. Démarrez le serveur WebSphere Application Server Community Edition :

a. Pour démarrer le serveur sans la sécurité Java SE activée, exécutez la commande suivante :

UNIX Linux `racine_wasce/bin/startup.sh`

Windows `racine_wasce/bin/startup.bat`

b. Pour démarrer le serveur avec la sécurité Java SE activée, effectuez les étapes suivantes : UNIX Linux

1) Open a command-line or terminal window and run the following copy command (or copy the contents of the specified policy file into your existing policy): `cp base_servicerest/gettingstarted/wasce/geronimo.policy wasce_root/bin`

2) Modifiez le fichier `racine_wasce/bin/setenv.sh`.

3) Après la ligne qui contient `"WASCE_JAVA_HOME="`, ajoutez la ligne suivante : `export JAVA_OPTS="-Djava.security.manager -Djava.security.policy=geronimo.policy"`.

Windows

1) Ouvrez une fenêtre de ligne de commande et exécutez la commande de copie suivante ou copiez le contenu du fichier de règles défini vers la stratégie existante :

```
copy base_servicerest\gettingstarted\wasce\geronimo.policy\bin
```

2) Modifiez le fichier `racine_wasce/bin/setenv.bat`.

3) Après la ligne qui contient "set WASCE_JAVA_HOME=", ajoutez la ligne suivante :

```
set JAVA_OPTS="-Djava.security.manager
-Djava.security.policy=geronimo.policy"
```

4. Ajoutez le fichier JAR d'exécution de client ObjectGrid au référentiel WebSphere Application Server Community Edition :

a. Ouvrez et connectez-vous à la console d'administration de WebSphere Application Server Community Edition. L'URL par défaut est `http://localhost:8080/console`, l'ID utilisateur par défaut est `system` et le mot de passe, `manager`.

b. Cliquez sur le lien **Référentiel** situé dans la partie gauche de la fenêtre de la console, dans le dossier **Services**.

c. Dans la section **Ajouter une archive au référentiel**, entrez les éléments suivants dans les zones de texte :

Tableau 24. Ajout d'une archive au référentiel

Zone de texte	Valeur
Fichier	`base_wxs/lib/ogclient.jar`

Tableau 24. Ajout d'une archive au référentiel (suite)

Zone de texte	Valeur
Groupe	com.ibm.websphere.xs
Artefact	ogclient
Version	7.1
Type	JAR

d. Cliquez sur le bouton Installer

Reportez-vous à la note technique suivante pour des détails sur les différentes manières dont les dépendances de classes et de bibliothèques peuvent être configurées : Specifying external dependencies to applications running on WebSphere Application Server Community Edition.

5. Déployez vers le serveur WebSphere Application Server Community Edition le module du service de données REST, le fichier `wxsrestservice.war`, et démarrez-le.
   a. Copiez et éditez l'exemple de fichier XML de plan de déploiement : `base_servicerest/gettingstarted/wasce/geronimo-web.xml`, afin d'inclure les dépendances de chemin au répertoire ou au fichier JAR de configuration de votre service de données REST. Voir la section pour un exemple de définition du chemin d'accès aux classes afin d'y inclure votre fichier `wxsRestService.properties` ainsi que d'autres fichiers de configuration et classes de métadonnées.
   b. Ouvrez et connectez-vous à la console d'administration de WebSphere Application Server Community Edition.

      Conseil : L'URL par défaut est `http://localhost:8080/console`. L'ID utilisateur par défaut est `system` et le mot de passe, `manager`.
   c. Cliquez sur le lien **Déployer nouveau** situé dans la partie gauche de la fenêtre de la console.
   d. Entrez les valeurs suivantes dans les zones de texte de la page **Installer de nouvelles applications** :

Tableau 25. Installer de nouvelles applications

Zone de texte	Valeur
Archive	`base_servicerest/lib/wxsrestservice.war`
Plan	`base_servicerest/gettingstarted/wasce/geronimo-web.xml`

      Conseil : Utilisez le chemin du fichier `geronimo-web.xml` que vous avez copié et édité au point 3.
   e. Cliquez sur le bouton Installer. La page de la console indique alors que l'application a été installée et démarrée.
   f. Examinez le journal de sortie du système WebSphere Application Server Community Edition ou la console pour déterminer si le service de données REST a démarré. Le message suivant doit apparaître :

      ```
      CWOBJ4000I: Le service de données REST de WebSphere eXtreme Scale a été démarré.
      ```

6. Démarrez le serveur WebSphere Application Server Community Edition en exécutant la commande suivante :
   - UNIX Linux `racine_wasce/bin/startup.sh`

- Windows `racine_wasce/bin/startup.bat`

7. Installez sur le serveur WebSphere Application Server Community Edition le service de données REST d'eXtreme Scale et l'exemple fourni :
   a. Ajoutez le fichier JAR d'exécution de client ObjectGrid au référentiel WebSphere Application Server Community Edition :
      1) Ouvrez et connectez-vous à la console d'administration de WebSphere Application Server Community Edition. L'URL par défaut est `http://localhost:8080/console`. L'ID utilisateur par défaut est `system` et le mot de passe, `manager`.
      2) Cliquez sur le lien **Référentiel** à gauche de la fenêtre de la console, dans le dossier Services.
      3) Dans la section **Ajouter une archive au référentiel**, entrez les éléments suivants dans les zones de texte :

Tableau 26. Ajout d'une archive au référentiel

Zone de texte	Valeur
Fichier	base_wxs/lib/ogclient.jar
Groupe	com.ibm.websphere.xs
Artefact	ogclient
Version	7.1
Type	JAR

      4) Cliquez sur le bouton Installer.

         Conseil : Reportez-vous à la note technique suivante pour des détails sur les différentes manières dont les dépendances de classes et de bibliothèques peuvent être configurées : Specifying external dependencies to applications running on WebSphere Application Server Community Edition.
   b. Déployez le module de service de données REST, `wxsrestservice.war`, vers le serveur WebSphere Application Server Community Edition.
      1) Editez l'exemple de fichier XML de déploiement `base_servicerest/gettingstarted/wasce/geronimo-web.xml` pour inclure les dépendances de chemin d'accès dans les répertoires du chemin d'accès aux classes de l'exemple Mise en route :
         - Modifiez le chemin "classesDirs" des deux GBeans du client Mise en route :

         Le chemin de "classesDirs" pour le bean géré GettingStarted_Client_SharedLib doit avoir la valeur `base_servicerest/gettingstarted/restclient/bin`.

         Le chemin de "classesDirs" pour le bean géré GettingStarted_Common_SharedLib doit avoir la valeur `base_servicerest/gettingstarted/restclient/bin`.
      2) Ouvrez et connectez-vous à la console d'administration de WebSphere Application Server Community Edition.
      3) Cliquez sur le lien **Déployer nouveau** situé dans la partie gauche de la fenêtre de la console.
      4) Entrez les valeurs suivantes dans les zones de texte de la page **Installer de nouvelles applications** :

Tableau 27. Install New Applications

Zone de texte	Valeur
Archive	*base_servicerest*/lib/wxsrestservice.war
Plan	*base_servicerest*/gettingstarted/wasce/geronimo-web.xml

5) Cliquez sur le bouton **Installer**.

   La page de la console indique alors que l'application a été installée et démarrée.

6) Recherchez le message ci-après sur la console ou dans le journal de sortie système de WebSphere Application Server Community Edition pour vérifier que le service de données REST a bien démarré :

   ```
   CWOBJ4000I: Le service de données REST de WebSphere eXtreme Scale a été démarré.
   ```

8. Vérifiez que le service de données REST fonctionne :

   Ouvrez un navigateur Web et accédez à l'URL `http://<hôte>:<port>/<racine_contexte>/restservice/<nom_grille>`

   Le port par défaut de WebSphere Application Server Community Edition est 8080 et il est défini à l'aide de la propriété HTTPPort dans le fichier `/var/config/config-substitutions.properties`.

   Par exemple : http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/

Résultats

Le document de service AtomPub est affiché.

Démarrage du service de données REST sur WebSphere Application Server Community Edition. :

Nous allons expliquer comment configurer et démarrer le service de données REST d'eXtreme Scale avec WebSphere Application Server Community Edition.

Avant de commencer

Vérifiez que la grille d'exemple est bien démarrée. Voir «Activation du service de données REST», à la page 358 pour savoir comment démarrer la grille.

Procédure

1. Téléchargez et installez WebSphere Application Server Community Edition version 2.1.1.3 ou plus récente dans le répertoire `racine_wasce`, par exemple le répertoire `/opt/IBM/wasce`.
2. Démarrez le serveur WebSphere Application Server Community Edition en exécutant la commande suivante :
   - Linux UNIX `racine_wasce/bin/startup.sh`
   - Windows `racine_wasce/bin/startup.bat`
3. Si la grille eXtreme Scale a été démarrée avec la sécurité d'eXtreme Scale activée, définissez les propriétés suivantes dans le fichier `base_servicerest/gettingstarted/restclient/bin/wxsRestService.properties`.

```
ogClientPropertyFile=base_servicerest/gettingstarted/security/security.ogclient.properties
loginType=none
```

4. Installez sur le serveur WebSphere Application Server Community Edition le service de données REST d'eXtreme Scale et l'exemple fourni :
   a. Ajoutez le fichier JAR d'exécution du client ObjectGrid au référentiel WebSphere Application Server Community Edition :
      1) Ouvrez et connectez-vous à la console d'administration de WebSphere Application Server Community Edition.

         Conseil : L'URL par défaut est `http://localhost:8080/console`. L'ID utilisateur par défaut est `system` et le mot de passe `manager`.
      2) Cliquez sur **Référentiel** dans le dossier Services.
      3) Dans la section **Ajouter une archive au référentiel**, entrez les éléments suivants dans les zones de texte :

Tableau 28. Archivage dans le référentiel

Zone de texte	Valeur
Fichier	base_wxs/lib/ogclient.jar
Groupe	com.ibm.websphere.xs
Artefact	ogclient
Version	7.0
Type	jar

      4) Cliquez sur le bouton Installer.

         Conseil : Reportez-vous à la note technique suivante pour des explications détaillées sur les différentes méthodes de configuration des dépendances de classes et des bibliothèques : Specifying external dependencies to applications running on WebSphere Application Server Community Edition.
   b. Déployez vers le serveur WebSphere Application Server Community Edition le module du service de données REST, le fichier `wxsrestservice.war`.
      1) Editez l'exemple de fichier XML de déploiement `base_servicerest/gettingstarted/wasce/geronimo-web.xml` pour inclure les dépendances de chemin d'accès dans les répertoires du chemin d'accès aux classes de l'exemple Mise en route :

         Modifiez le chemin classesDirs des deux GBeans du client Mise en route :
         - Le chemin de "classesDirs" pour le bean géré GettingStarted_Client_SharedLib doit avoir la valeur `base_servicerest/gettingstarted/restclient/bin`.
         - Le chemin de "classesDirs" pour le bean géré GettingStarted_Common_SharedLib doit avoir la valeur `base_servicerest/gettingstarted/restclient/bin`.
      2) Ouvrez et connectez-vous à la console d'administration de WebSphere Application Server Community Edition.

         Conseil : L'URL par défaut est `http://localhost:8080/console`. L'ID utilisateur par défaut est `system` et le mot de passe `manager`.
      3) Cliquez sur **Déployer nouveau**.
      4) Entrez les valeurs suivantes dans les zones de texte de la page **Installer de nouvelles applications** :

Tableau 29. Valeurs d'installation

Zone de texte	Valeur
Archive	`base_servicerest/lib/wxsrestservice.war`
Plan	`base_servicerest/gettingstarted/wasce/geronimo-web.xml`

5) Cliquez sur le bouton Installer.

   La page de la console indique que l'application est installée et a démarré.

6) Recherchez le message ci-après sur la console ou dans le journal de sortie système de WebSphere Application Server Community Edition pour vérifier que le service de données REST a bien démarré :

   `CWOBJ4000I : Le service de données REST de WebSphere eXtreme Scale a été démarré.`

5. Vérifiez que le service de données REST fonctionne :
   a. Ouvrez le lien suivant dans une fenêtre de navigateur : `http://localhost:8080/wxsrestservice/restservice/NorthwindGrid`. Le document de service de la grille NorthwindGrid s'affiche.
   b. Ouvrez le lien suivant dans une fenêtre de navigateur : `http://localhost:8080/wxsrestservice/restservice/NorthwindGrid`. Le document Entity Model Data Extensions (EDMX) s'affiche.
6. Pour arrêter les processus de grille, utilisez `CTRL+C` dans les fenêtres de commande respectives.
7. Pour arrêter WebSphere Application Server Community Edition, utilisez la commande suivante :
   - UNIX Linux `racine_wasce/bin/shutdown.sh`
   - Windows `racine_wasce\bin\shutdown.bat`

   Conseil : L'ID utilisateur par défaut est `system` et le mot de passe `manager`. Si vous utilisez un port personnalisé, utilisez l'option `-port`.

Déploiement du service de données REST sur Apache Tomcat

Cette rubrique explique comment configurer le service de données REST WebSphere eXtreme Scale sur Apache Tomcat Version 5.5 et les versions suivantes.

Pourquoi et quand exécuter cette tâche

- Un JRE ou JDK IBM ou Sun, version 5 ou ultérieure est installé et la variable d'environnement JAVA_HOME est spécifiée.
- Apache Tomcat Version 5.5 ou ultérieure est installé. Voir Apache Tomcat pour savoir comment installer Tomcat.
- Installation autonome WebSphere eXtreme Scale.

Procédure

1. Si vous utilisez un JDK ou un JRE Sun, installez l'ORB IBM dans Tomcat :
   a. Tomcat version 5.5 :

      Copiez tous les fichiers JAR depuis :

      le répertoire *base_wxs*`/lib/endorsed`

      vers :

      le répertoire *racine_tomcat*`/common/endorsed`

b. Tomcat version 6.0 :

Créez un répertoire "endorsed" :

UNIX Linux `mkdir racine_tomcat/endorsed`

Windows `md racine_tomcat/endorsed`

Copiez tous les fichiers JAR de :

`base_wxs/lib/endorsed`

vers :

`racine_tomcat/common/endorsed`

2. Configurez et démarrez une grille de données.
   a. Pour plus d'informations sur la configuration d'une grille de données pour l'utiliser avec le service de données REST, voir Chapitre 6, «configuration», à la page 223.
   b. Vérifiez qu'un client eXtreme Scale arrive à se connecter aux entités de la grille et à y accéder. Pour un exemple, voir «Configuration des services de données REST», à la page 357.
3. Générez le répertoire ou le fichier JAR de configuration du service REST d'eXtreme Scale. Pour les détails, voir dans «Installation du service de données REST», à la page 200 les explications sur le packaging et le déploiement.
4. Déployez le module du service de données REST : wxsrestservice.war sur le serveur Tomcat.

   Copiez le fichier `wxsrestservice.war` depuis :

   rép_base_servicerest`/lib`

   vers :

   racine_tomcat`/webapps`
5. Ajoutez le fichier JAR d'exécution du client ObjectGrid et le fichier JAR de l'application dans le chemin d'accès aux classes partagé, dans Tomcat :
   a. Editez le fichier *racine_tomcat*`/conf/catalina.properties`.
   b. Ajoutez les noms de chemin suivants à la fin de la propriété shared.loader, en les séparant par une virgule :
      - *rép_base_wxs*`/lib/ogclient.jar`
      - *rép_base_servicerest*`/gettingstarted/restclient/bin`
      - *rép_base_servicerest*`/gettingstarted/common/bin`
6. Si vous utilisez la sécurité Java 2, ajoutez les droits de sécurité au fichier de règles tomcat :
   - Si vous utilisez Tomcat version 5.5 :

     Fusionnez le contenu de l'exemple de fichier de `règles catalina` 5.5 qui se trouve dans

     rép_base_servicerest`/gettingstarted/tomcat/catalina-5_5.policy` avec le fichier *racine_tomcat*`/conf/catalina.policy`.
   - Si vous utilisez Tomcat version 6.0 :

     Fusionnez le contenu de l'exemple de fichier de `règles catalina` 6.0 qui se trouve dans

     rép_base_servicerest`/gettingstarted/tomcat/catalina-6_0.policy` avec le fichier *racine_tomcat*`/conf/catalina.policy`.
7. Démarrez le serveur Tomcat :
   - **Si vous utilisez Tomcat 5.5 sous UNIX or Windows, ou la distribution ZIP de Tomcat 6.0 :**

a. `cd racine_tomcat/bin`
b. Démarrez le serveur :
   - Sans la sécurité Java 2 activée :

     UNIX Linux `./catalina.sh run`

     Windows `catalina.bat run`
   - Avec la sécurité Java 2 activée :

     UNIX Linux `./catalina.sh run -security`

     Windows `catalina.bat run -security`
c. Les journaux d'Apache Tomcat sont affichés sur la console. Lorsque le service de données REST a correctement démarré, le message suivant est affiché dans la console d'administration :

   ```
   CWOBJ4000I: Le service de données REST de WebSphere eXtreme Scale a été démarré.
   ```

- **Si vous utilisez Tomcat 6.0 sous Windows à l'aide de la distribution du programme d'installation Windows :**
  a. `cd /bin`
  b. Démarrez l'outil de configuration d'Apache Tomcat 6 :

     ```
     tomcat6w.exe
     ```
  c. Pour activer la sécurité Java 2 : (facultatif) :

     Ajoutez les entrées suivantes aux options Java dans la page Java de la fenêtre des propriétés d'Apache Tomcat 6 :

     ```
     -Djava.security.manager
     -Djava.security.policy=\conf\catalina.policy
     ```
  d. Cliquez sur le bouton Démarrer de la fenêtre de propriétés d'Apache Tomcat 6 pour démarrer le serveur Tomcat.
  e. Consultez les journaux suivants pour vérifier que le serveur Tomcat a été correctement démarré :
     - *racine_tomcat*`/bin/catalina.log`

       Affiche le statut du moteur du serveur Tomcat
     - *racine_tomcat*`/bin/stdout.log`

       Affiche le journal de la sortie système.
  f. Si le service de données REST est correctement démarré, le message suivant est affiché dans le journal de la sortie système :

     ```
     CWOBJ4000I: Le service de données REST de WebSphere eXtreme Scale a été démarré.
     ```

8. Vérifiez que le service de données REST fonctionne bien.

   Ouvrez un navigateur Web et accédez à l'adresse URL suivante :

   `http://`*`host:port/racine_contexte`*`/restservice/`*`nom_grille`*

   Le port par défaut pour Tomcat est 8080 et il est configuré dans le fichier *racine_tomcat*`/conf/server.xml` dans l'élément <Connector>.

   Par exemple :

   ```
   http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/
   ```

Résultats

Le document de service AtomPub est affiché.

Démarrage de services de données REST dans Apache Tomcat :

Cette rubrique explique comment configurer et démarrer le service de données REST d'eXtreme Scale avec Apache Tomcat version 5.5 ou plus récente.

Avant de commencer

Vérifiez que l'exemple de grille de données eXtreme Scale est démarré. Voir «Activation du service de données REST», à la page 358 pour plus d'informations sur le démarrage de la grille.

Procédure

1. Téléchargez et installez dans racine_tomcat Apache Tomcat version 5.5 ou ultérieure. Par exemple : `/opt/tomcat`
2. Installez sur le serveur Tomcat le service de données REST d'eXtreme Scale et l'exemple fourni :
   a. Si vous utilisez un JRE ou un JDK Sun, vous devez installer l'ORB IBM sur Tomcat :
      - Pour Tomcat version 5.5

        Copiez tous les fichiers JAR de :

        `base_wxs/lib/endorsed`

        vers

        `racine_tomcat/common/endorsed`
      - Pour Tomcat version 6.0
        1) Créez un répertoire "validé".
           – UNIX Linux `mkdir racine_tomcat/endorsed`
           – Windows `md racine_tomcat/endorsed`
        2) Copiez tous les fichiers JAR de :

           `base_wxs/lib/endorsed`

           vers

           `racine_tomcat/endorsed`

   b. Déployez le module de service de données REST : wxsrestservice.war vers le serveur Tomcat.

      Copiez le fichier wxsrestservice.war depuis :

      `base_servicerest/lib`

      vers :

      `racine_tomcat/webapps`

   c. Ajoutez le fichier JAR d'exécution client ObjectGrid et le fichier JAR d'application au chemin d'accès aux classes dans Tomcat :
      1) Modifiez le fichier `racine_tomcat/conf/catalina.properties`.
      2) Ajoutez les noms de chemins suivants à la fin de la propriété shared.loader sous forme de liste séparée par des virgules :
         - base_wxs/lib/ogclient.jar
         - base_servicerest/gettingstarted/restclient/bin
         - base_servicerest/gettingstarted/common/bin

Important : Le séparateur de chemin doit être une barre **oblique**.

3. Si la grille de données eXtreme Scale a été démarrée avec la sécurité eXtreme Scale activée, définissez la propriété suivante dans le fichier `restservice_home/gettingstarted/restclient/bin/wxsRestService.properties`.

```
ogClientPropertyFile=base_servicerest/gettingstarted/security/security.ogclient.properties
loginType=none
```

4. Démarrez le serveur Tomcat avec le service de données REST :
   - Si vous utilisez Tomcat 5.5 sous UNIX ou Windows, ou Tomcat 6.0 sous UNIX :
     a. `cd racine_tomcat/bin`
     b. Démarrez le serveur :
        - UNIX Linux `./catalina.sh run`
        - Windows `catalina.bat run`
     c. La console affiche ensuite les journaux d'Apache Tomcat. Au démarrage du service de données REST, le message suivant s'affiche dans la console d'administration :

        `CWOBJ4000I : Le service de données REST de WebSphere eXtreme Scale a été démarré.`
   - Si vous utilisez Tomcat 6.0 sous Windows :
     a. `cd racine_tomcat/bin`
     b. Démarrez l'outil de configuration Apache Tomcat 6 avec la commande suivante : `tomcat6w.exe`
     c. Dans la fenêtre des propriétés d'Apache Tomcat 6, cliquez sur le bouton Démarrer pour démarrer le serveur Tomcat.
     d. Révisez les journaux suivants pour vérifier que le serveur Tomcat a démarré correctement :
        - `racine_tomcat/bin/catalina.log`

          Affiche l'état du moteur du serveur Tomcat.
        - `racine_tomcat/bin/stdout.log`

          Affiche le journal de sortie système.
     e. Au démarrage du service de données REST, le message suivant s'affiche dans le journal de sortie système : `CWOBJ4000I : Le service de données REST de WebSphere eXtreme Scale a été démarré.`
5. Vérifiez que le service de données REST fonctionne :
   a. Ouvrez un navigateur et accédez à :

      http://localhost:8080/wxsrestservice/restservice/NorthwindGrid

      Le document de service pour NorthwindGrid s'affiche.
   b. Accédez à :

      http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/$metadata

      Le document Entity Model Data Extensions (EDMX) s'affiche.
6. Pour arrêter les processus de grille de données, utilisez CTRL+C dans la fenêtre de commande correspondante.
7. Pour arrêter Tomcat, utilisez CTRL +C dans la fenêtre depuis laquelle vous l'avez démarré.

Configuration des navigateurs Web pour accéder aux flux ATOM du service de données REST

Par défaut, le service de données REST d'eXtreme Scale crée des flux ATOM lorsqu'il utilise un navigateur Web. Le format des flux ATOM risque de ne pas être compatible avec les navigateurs plus anciens. Il y a également le risque d'une mauvaise interprétation des données qui ne seront pas affichées comme des données XML. Vous pouvez configurer Internet Explorer Version 8 et Firefox Version 3 pour afficher les flux ATOM et les données XML dans le navigateur.

Pourquoi et quand exécuter cette tâche

Par défaut, le service de données REST d'eXtreme Scale crée des flux ATOM lorsqu'il utilise un navigateur Web. Le format des flux ATOM risque de ne pas être compatible avec les autres navigateurs ou d'être interprété de sorte que les données ne puissent pas être consultées comme des données XML. Pour les anciens navigateurs, vous serez invité à sauvegarder les fichiers sur le disque. Une fois que les fichiers ont été téléchargés, utilisez votre lecteur XML favori pour consulter les fichiers. Le XML généré n'étant pas formaté pour être affiché, tout est imprimé sur une seule ligne. La plupart des programmes de lecture XML, tels qu'Eclipse, prennent en charge le reformatage du XML dans un format lisible.

Pour les navigateurs modernes, tels que Microsoft Internet Explorer Version 8 et Firefox Version 3, les fichiers XML ATOM peuvent être affichés de manière native dans le navigateur. Les rubriques ci-après fournissent des détails sur la manière de configurer Internet Explorer Version 8 et Firefox Version 3 pour afficher les flux ATOM et le XML dans le navigateur.

Procédure

Configuration d'Internet Explorer Version 8

- Pour permettre à Internet Explorer de lire les flux ATOM que le service de données REST génère, procédez comme suit :
  1. Cliquez sur **Outils** > **Options Internet**
  2. Sélectionnez l'onglet **Contenu**
  3. Cliquez sur le bouton **Paramètres** de la section **Flux et composants Web Slice**
  4. Désélectionnez la case "Activer le mode Lecture du flux"
  5. Cliquez sur **OK** pour retourner au navigateur.
  6. Redémarrez Internet Explorer.

Configuration de Firefox Version 3

- Firefox n'affiche pas automatiquement les pages avec le type de contenu suivant : application/atom+xml. La première fois qu'une page est affichée, Firefox vous invite à sauvegarder le fichier. Pour afficher la page, ouvrez le fichier avec Firefox, comme suit :
  1. Dans la boîte de dialogue de sélection de l'application, sélectionnez le bouton d'option "Ouvrir avec" et cliquez sur le bouton **Parcourir**.
  2. Accédez au répertoire d'installation de Firefox. Par exemple : `C:\Program Files\Mozilla Firefox`
  3. Sélectionnez `firefox.exe`, puis cliquez sur le bouton **OK**.
  4. Cochez la case "Toujours utiliser ce programme pour ouvrir ce type de fichier".

5. Cliquez sur le bouton **OK**.
6. Firefox affiche ensuite la page XML ATOM dans une nouvelle fenêtre ou page de navigateur

- Firefox affiche automatiquement les flux ATOM dans un format lisible. Toutefois, les flux créés par le service de données REST incluent XML. Firefox ne peut pas afficher le XML à moins que vous ne désactiviez le présentateur de flux. Contrairement à Internet Explorer, dans Firefox, le plug-in d'affichage des flux ATOM doit être édité de manière explicite. Pour configurer Firefox afin qu'il puisse lire les flux ATOM comme des fichiers XML, procédez comme suit :
  1. Ouvrez le fichier suivant dans un éditeur de texte : `<firefoxInstallRoot>\components\FeedConverter.js`. Dans le chemin d'accès, `<firefoxInstallRoot>` correspond au répertoire principal dans lequel Firefox est installé.

     Pour les systèmes d'exploitation Windows, le répertoire par défaut est le suivant : `C:\Program Files\Mozilla Firefox`.
  2. Recherchez le fragment de code similaire au suivant :

     ```
     // montre la page de flux si elle n'a pas été reniflée
     et que nous avons un document,
     // ou un document, un titre et un lien ou un ID
     if (result.doc && (!this._sniffed ||
         (result.doc.title && (result.doc.link || result.doc.id)))) {
     ```
  3. Placez les deux lignes commençant par `if` et `result` en commentaire, en les précédant de deux barres obliques (//).
  4. Ajoutez l'instruction suivante au fragment de code : `if(0) {`.
  5. Le texte résultant doit ressembler au suivant :

     ```
     // montre la page de flux si elle n'a pas été reniflée
     et que nous avons un document,
     // ou un document, un titre et un lien ou un ID
     //if (result.doc && (!this._sniffed ||
     //    (result.doc.title && (result.doc.link || result.doc.id)))) {
     if(0) {
     ```
  6. Enregistrez le fichier.
  7. Redémarrez Firefox
  8. Firefox peut maintenant afficher automatiquement tous les flux dans le navigateur.
- Testez votre configuration en essayant quelques URL.

Exemple

Cette section décrit certains exemples d'URL qui peuvent être utilisés pour afficher les données qui ont été ajoutées par l'exemple d'initiation fourni avec le service de données REST. Avant d'utiliser les URL suivantes, ajoutez le fichier par défaut à l'exemple de grille de données eXtreme Scale en utilisant l'exemple de client Java ou l'exemple de client Visual Studio WCF Data.

Dans les exemples qui suivent, l'on part du principe que le port utilisé est le 8080, mais cela peut varier. Reportez-vous à la section pour des explications détaillées sur la manière de configurer le service de données REST sur différents serveurs d'applications.

- Visualiser un seul client dont l'ID est "ACME" :

  ```
  http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Customer('ACME')
  ```
- Visualiser toutes les commandes du client "ACME" :

  ```
  http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Customer('ACME')/orders
  ```
- Visualiser le client "ACME" et les commandes :

```
http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Customer('ACME')?$expand=orders
```

- Visualiser la commande 1000 du client "ACME" :

```
http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Order(orderId=1000,customer_customerId='ACME')
```

- Visualiser la commande 1000 du client "ACME" et le Customer qui lui est associé :

```
http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/
Order(orderId=1000,customer_customerId='ACME')?$expand=customer
```

- Visualiser la commande 1000 du client "ACME" et le Customer et les OrderDetails associés à ce client :

```
http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/
Order(orderId=1000,customer_customerId='ACME')?$expand=customer,orderDetails
```

- Visualiser toutes les commandes du client "ACME" pour le mois d'octobre 2009 (GMT) :

```
http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/
Customer(customerId='ACME')/orders?$filter=orderDate
ge datetime'2009-10-01T00:00:00'
and orderDate lt datetime'2009-11-01T00:00:00'
```

- Visualiser les trois premières commandes et les trois premiers orderDetails du client "ACME" pour le mois d'octobre 2009 (GMT) :

```
http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/
Customer(customerId='ACME')/orders?$filter=orderDate
ge datetime'2009-10-01T00:00:00'
and orderDate lt datetime'2009-11-01T00:00:00'
&$orderby=orderDate&$top=3&$expand=orderDetails
```

Utilisation d'un client Java avec les services de données REST

L'application client Java utilise l'API EntityManager d'eXtreme Scale pour insérer des données dans la grille.

Pourquoi et quand exécuter cette tâche

Les sections précédentes expliquent comment créer une grille de données eXtreme Scale et configurer et démarrer le service de données eXtreme Scale REST. L'application client Java utilise l'API EntityManager d'eXtreme Scale pour insérer des données dans la grille. Elle ne montre pas comment utiliser les interfaces REST. Ce client a pour fonction de montrer comme l'API EntityManager est utilisée pour interagir avec la grille de données eXtreme Scale et de modifier les données dans la grille. Pour afficher des données dans la grille avec le service de données REST, utilisez un navigateur Web ou l'application client Visual Studio 2008.

Procédure

Pour ajouter rapidement du contenu à la grille de données eXtreme Scale, exécutez la commande suivante :

1. Ouvrez une ligne de commande ou une fenêtre de terminal, puis définissez la variable d'environnement JAVA_HOME :
   - Linux UNIX `export JAVA_HOME=base_java`
   - Windows `set JAVA_HOME=base_java`
2. `cd base_servicerest/gettingstarted`
3. Insérez des données dans la grille. Les données insérées seront extraites ultérieurement à l'aide d'un navigateur Web et du service de données REST.

   Si la grille de données a été démarrée *sans*eXtreme Scale la sécurité, utilisez les commandes suivantes.

- UNIX Linux `./runclient.sh load default`
- Windows `runclient.bat load default`

Si la grille de données a été démarrée *sans*eXtreme Scale la sécurité, utilisez les commandes suivantes.

- UNIX Linux `./runclient_secure.sh load default`
- Windows `runclient_secure.bat load default`

Pour un client Java, utilisez la syntaxe suivante :

- UNIX Linux `runclient.sh` *`commande`*
- Windows `runclient.bat` *`commande`*

Les commandes suivantes sont disponibles :

- `load default`

  Charge un ensemble prédéfini d'entités Customer, Category et Product dans la grille de données et crée un ensemble aléatoire de commandes pour chaque client.
- `load category` *`IDcatégorie Nomcatégorie IDpremierproduit nbre_produits`*

  Crée une catégorie de produits et un nombre fixe d'entités Product dans la grille de données. Le paramètre firstProductId identifie le numéro d'identification du premier produit et chaque produit suivant se voit affecter l'identifiant suivant jusqu'à ce que le nombre spécifié de produits soit créé.
- `load customer` *`companyCode contactNamecompanyName numOrders firstOrderIdshipCity maxItems discountPct`*

  Charge un nouveau client dans la grille de données et crée un groupe fixe d'entités Order pour un produit aléatoire chargé actuellement dans la grille. Le nombre de commandes est déterminé par le paramètre <numOrders>. Chaque commande sera dotée d'un nombre aléatoire d'entités OrderDetail jusqu'à <maxItems>
- `display customer` *`companyCode`*

  Affiche une entité Customer et les entités Order et OrderDetail associées.
- `display category` *`categoryId`*

  Affiche une entité de produit Category et les entités Product associées.

Résultats

- `runclient.bat load default`
- `runclient.bat load customer IBM "John Doe" "IBM Corporation" 5 5000 Rochester 5 0.05`
- `runclient.bat load category 5 "Household Items" 100 5`
- `runclient.bat display customer IBM`
- `runclient.bat display category 5`

Exécution et génération de l'exemple de données de la grille et du client Java avec Eclipse

L'échantillon de démarrage de service de données REST peut être mis à jour et amélioré à l'aide d'Eclipse. Pour plus d'informations sur la configuration de votre environnement Eclipse, voir le document : `base_servicerest/gettingstarted/ECLIPSE_README.txt`.

Une fois que le projet WXSRestGettingStarted a été importé dans Eclipse et que sa génération s'effectue correctement, l'échantillon se recompile automatiquement et les fichiers script utilisés pour démarrer le serveur conteneur et le client sélectionnent automatiquement les fichiers de classes et les fichiers XML. Le service de données REST détecte automatiquement toute modification, car le serveur Web est configuré pour effectuer une lecture automatique des répertoires de construction Eclipse.

Important : En cas de modification des fichiers source ou de configuration, le serveur de conteneur eXtreme Scale et l'application du service de données REST doivent tous deux être redémarrés. Le serveur conteneur eXtreme Scale doit être démarré avant l'application Web du service de données REST.

Client WCF de Visual Studio 2008 avec le service de données REST

L'exemple Mise en route du service de données REST d'eXtreme Scale inclut un client WCF Data Services qui peut interagir avec le service de données REST. L'exemple est écrit comme une application de ligne de commande dans C#.

Configuration logicielle requise

L'exemple de client WCF Data Services C# requiert la configuration suivante :

- Système d'exploitation
  - Microsoft Windows XP
  - Microsoft Windows Server 2003
  - Microsoft Windows Server 2008
  - Microsoft Windows Vista
- Microsoft Visual Studio 2008 avec Service Pack 1

  Conseil : Pour les configurations matérielle et logicielle supplémentaires requises, voir le lien précédent.
- Microsoft .NET Framework 3.5 Service Pack 1
- Microsoft Support : La mise à jour pour .NET Framework 3.5 Service Pack 1 est disponible

Génération et exécution du client Mise en route

L'exemple de client des services de données WCF inclut un projet et une solution Visual Studio 2008, ainsi que le code source permettant d'exécuter l'exemple. L'exemple doit être chargé dans Visual Studio 2008 et compilé dans un programme exécutable sous Windows pour pouvoir être exécuté. Pour générer et exécuter l'exemple, voir le document texte : `base_servicerest/gettingstarted/VS2008_README.txt`.

Syntaxe des commandes du client WCF Data Services C#

Windows `WXSRESTGettingStarted.exe <URL du service> <commande>`

L'<URL du service> est l'URL du service de données REST d'eXtreme Scale configuré dans la section .

Les commandes suivantes sont disponibles :

- `load default`

Charge un ensemble prédéfini d'entités Customer, Category et Product dans la grille de données et crée un ensemble aléatoire de commandes pour chaque client.

- `load category <categoryId> <categoryName> <firstProductId> <numProducts>`

  Crée une catégorie de produits et un nombre fixe d'entités Product dans la grille de données. Le paramètre firstProductId identifie l'identificateur du premier produit et chaque produit suivant reçoit le prochain ID jusqu'à ce que le nombre de produits spécifié soit créé.
- `load customer <companyCode> <contactName> <companyName> <numOrders> <firstOrderId> <shipCity> <maxItems> <discountPct>`

  Charge un nouveau client dans la grille de données et crée un groupe fixe d'entités Order pour un produit aléatoire déjà chargé actuellement dans la grille. Le nombre de commandes est déterminé par le paramètre <numOrders>. Chaque commande contient un nombre aléatoire d'entités OrderDetail, inférieur à la valeur <maxItems>
- `display customer <companyCode>`

  Affiche un entité Customer et les entités Order et OrderDetail associées.
- `display category <categoryId>`

  Affiche l'entité Category d'un produit et les entités Product associées.
- `unload`

  Supprime toutes les entités chargées à l'aide de la commande "default load".

Les exemples suivants illustrent diverses commandes.

- `WXSRestGettingStarted.exe http://localhost:8080/wxsrestservice/restservice/NorthwindGrid load default`
- `WXSRestGettingStarted.exe http://localhost:8080/wxsrestservice/restservice/NorthwindGrid load customer`
- `IBM "John Doe" "IBM Corporation" 5 5000 Rochester 5 0.05`
- `WXSRestGettingStarted.exe http://localhost:8080/wxsrestservice/restservice/NorthwindGrid load category 5 "Household Items" 100 5`
- `WXSRestGettingStarted.exe http://localhost:8080/wxsrestservice/restservice/NorthwindGrid display customer IBM`
- `WXSRestGettingStarted.exe http://localhost:8080/wxsrestservice/restservice/NorthwindGrid display category 5`

Configuration des serveurs pour OSGi

WebSphere eXtreme Scale inclut un ensemble OSGi de serveur qui permet de démarrer et de configurer les serveurs et les conteneurs dans une infrastructure OSGi. Les rubriques de configuration expliquent comment utiliser l'ensemble de serveur eXtreme Scale, le service OSGi Blueprint et la configuration eXtreme Scale pour exécuter des serveurs eXtreme Scale dans une infrastructure OSGi Eclipse Equinox.

Pourquoi et quand exécuter cette tâche

Vous devez exécuter les tâches suivantes pour démarrer un serveur eXtreme Scale dans Eclipse Equinox:

Procédure

1. Créez un ensemble OSGi qui stockera les plug-ins eXtreme Scale en les exposant comme services et mettez à jour le fichier XML descripteur d'ObjectGrid pour référencer les services.
2. Configurez OSGi pour démarrer le serveur de conteneur eXtreme Scale.
3. Installez et redémarrez l'ensemble de serveur eXtreme Scale dans l'infrastructure OSGi.
4. Installez et démarrez l'ensemble OSGi qui contient les plug-ins eXtreme Scale.

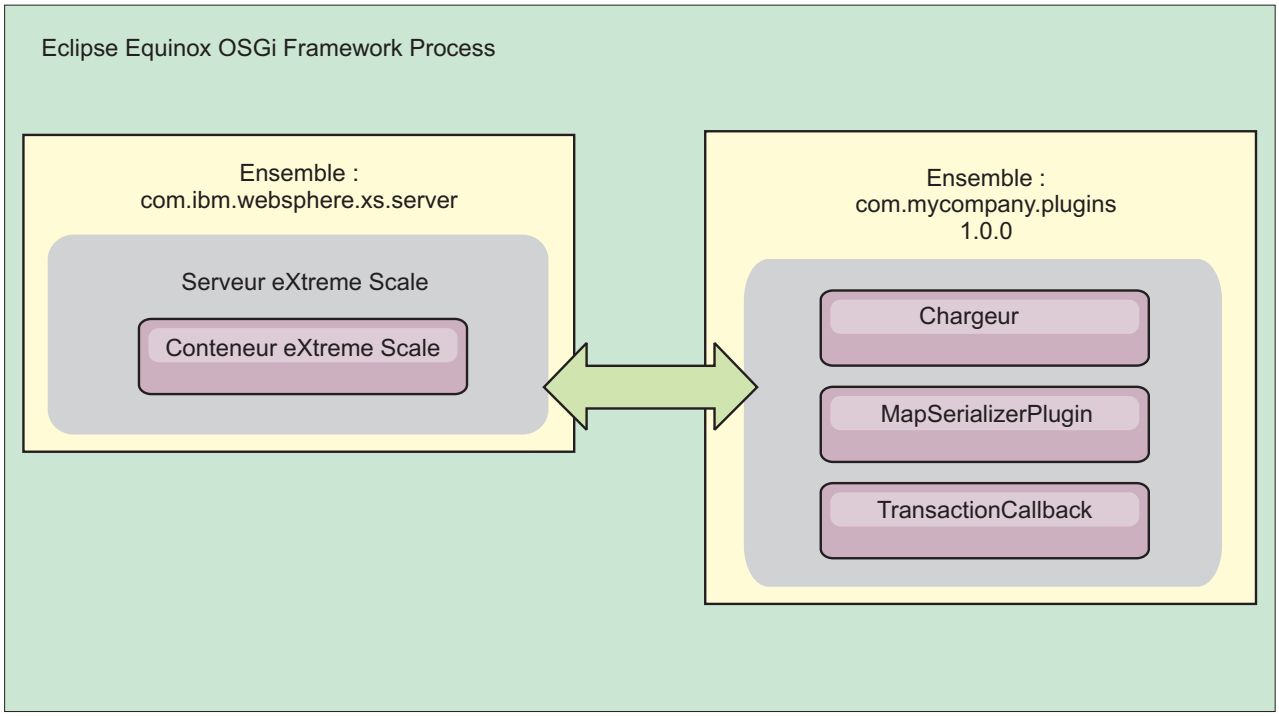


Figure 48. Processus Eclipse Equinox d'installation et de démarrage des ensembles OSGi avec des plug-ins eXtreme Scale

Configuration des plug-ins eXtreme Scale avec OSGi Blueprint

Tous les plug-ins eXtreme Scale ObjectGrid et BackingMap peuvent être définis comme beans et services OSGi en utilisant le service OSGi Blueprint disponible avec Eclipse Gemini ou Apache Aries.

Avant de commencer

Pour pouvoir configurer vos plug-ins comme services OSGi, vous devez regrouper les plug-ins dans un ensemble OSGi et connaître les concepts de base des plug-ins requis. L'ensemble doit importer les modules client ou serveur WebSphere eXtreme Scale et d'autres packages dépendants nécessaires aux plug-ins ou créer une dépendance d'ensemble dans les ensembles de serveur ou de client eXtreme Scale. Cette rubrique explique comment configurer le fichier XML Blueprint XML pour créer des beans de plug-in et les exposer comme services OSGi pour que eXtreme Scale les utilise.

Pourquoi et quand exécuter cette tâche

Les beans et services sont définis dans un fichier XML Blueprint et le conteneur Blueprint découvre, crée et interconnecte les beans et les expose comme services. Le processus rend les beans accessibles aux autres ensembles OSGi, y compris les ensembles de serveur et de client eXtreme Scale.

Lors de la création de services de plug-in personnalisés pour les utiliser avec eXtreme Scale, l'ensemble qui doit héberger les plug-ins doit être configuré pour utiliser Blueprint. En outre, un fichier XML Blueprint doit être créé et stocké dans l'ensemble. Lisez la rubrique relative à la création d'applications OSGi avec la spécification Blueprint Container qui décrit de manière générale la spécification.

Procédure

1. Créez un fichier XML Blueprint. Attribuez-lui un nom de votre choix. Toutefois, vous devez inclure l'espace de nom Blueprint :

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0">
   ...
   </blueprint>
   ```

2. Créez des définitions de bean dans le fichier XML Blueprint pour chaque plug-in eXtreme Scale.

   Les beans sont définis en utilisant l'élément `<bean>`, ils peuvent être connectés à d'autres références de bean et ils peuvent inclure des paramètres d'initialisation.

   Important : Lors de la définition d'un bean, vous devez utiliser la portée correcte. Blueprint prend en charge les portées singleton et prototype. eXtreme Scale prend également en charge une portée de fragment personnalisée.

   Définissez la plupart des plug-ins eXtreme Scale comme prototype ou beans à portée de fragment, car tous les beans doivent être uniques pour chaque fragment ObjectGrid ou instance BackingMap auquel ou à laquelle ils sont associés. Les beans à portée de fragment peuvent être utiles lorsque vous utilisez les beans dans d'autres contextes pour pouvoir extraire l'instance correcte.

   Pour définir un bean à portée prototype, utilisez l'attribut `scope="prototype"` sur le bean :

   ```
   <bean id="myPluginBean" class="com.mycompany.MyBean" scope="prototype">
   ...
   </bean>
   ```

   Pour définir un beans à portée de fragment, vous devez ajouter l'espace de nom `objectgrid` au schéma XML et utiliser l'attribut `scope="objectgrid:shard"` sur le bean :

   ```
   <?xml version="1.0" encoding="UTF-8"?>

   <blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
              xmlns:objectgrid="http://www.ibm.com/schema/objectgrid"

       xsi:schemaLocation="http://www.ibm.com/schema/objectgrid
                   http://www.ibm.com/schema/objectgrid/objectgrid.xsd">

       <bean id="myPluginBean" class="com.mycompany.MyBean"
      scope="objectgrid:shard">
       ...
       </bean>

   ...
   ```

3. Créez des définitions de bean PluginServiceFactory pour chaque bean de plug-in. Tous les beans eXtreme Scale doivent avoir un bean PluginServiceFactory défini pour que la portée de bean correcte puisse être appliquée. eXtreme Scale inclut une fabrique BlueprintServiceFactory que vous pouvez utiliser. Elle contient deux propriétés que vous devez définir. Vous devez affecter à la propriété blueprintContainer la référence `blueprintContainer` et attribuer à la propriété beanId le nom de l'identificateur du bean. Lorsque eXtreme Scale recherche le service pour instancier les beans appropriés, le serveur recherche l'instance du composant bean en utilisant le conteneur Blueprint.

   ```
   bean id="myPluginBeanFactory"
       class="com.ibm.websphere.objectgrid.plugins.osgi.BluePrintServiceFactory">
       <property name="blueprintContainer" ref="blueprintContainer"/>
       <property name="beanId" value="myPluginBean" />
   </bean>
   ```

4. Créez un gestionnaire de service pour chaque bean PluginServiceFactory. Chaque gestionnaire de service expose le bean PluginServiceFactory en utilisant l'élément `<service>`. L'élément de service identifie le nom à exposer à OSGi, la référence au bean PluginServiceFactory et l'interface à exposer, ainsi que le classement du service. eXtreme Scale utilise le classement du gestionnaire de service pour effectuer des mises à niveau de service lorsque la grille eXtreme Scale est active. Si le classement n'est pas défini, l'infrastructure OSGi utilise le classement 0 par défaut. Consultez la rubrique relative à la mise à jour des classements de service pour plus d'informations.

   Blueprint contient diverses options de configuration des gestionnaires de service. Pour définir un gestionnaire de service simple pour un bean PluginServiceFactory, créez un élément `<service>` pour chaque bean PluginServiceFactory :

   ```
   <service ref="myPluginBeanFactory"
       interface="com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactory"
       ranking="1">
   </service>
   ```

5. Stockez le fichier XML Blueprint dans l'ensemble de plug-ins. Le fichier XML Blueprint doit être stocké dans le répertoire `OSGI-INF/blueprint` du conteneur Blueprint pour être découvert.

   Pour stocker le fichier XML Blueprint dans un répertoire différent, vous devez définir l'en-tête de manifeste Bundle-Blueprint suivant :

   ```
   Bundle-Blueprint: OSGI-INF/blueprint.xml
   ```

Résultats

Les plug-ins eXtreme Scale sont maintenant configurés pour être exposés dans un conteneur OSGi Blueprint. En outre, le fichier XML descripteur ObjectGrid est configuré pour référencer les plug-ins en utilisant le service OSGi Blueprint.

Configuration des serveurs avec OSGi Blueprint

Vous pouvez configurer les serveurs de conteneur WebSphere eXtreme Scale en utilisant un fichier XML OSGi Blueprint qui permet de simplifier le regroupement et le développement d'ensembles de serveur autonomes.

Avant de commencer

Cette rubrique suppose que vous avez exécuté les tâches suivantes :

- L'infrastructure OSGi Eclipse Equinox a été installée et démarrée avec le conteneur Eclipse Gemini ou Apache Aries Blueprint.

- L'ensemble de serveur eXtreme Scale a été installé et démarré.
- L'ensemble de plug-ins dynamiques eXtreme Scale a été créé.
- Le fichier XML descripteur eXtreme Scale ObjectGrid et le fichier XML de stratégie de déploiement ont été créés.

Pourquoi et quand exécuter cette tâche

Cette tâche explique comment configurer un serveur eXtreme Scale avec un conteneur en utilisant un fichier XML Blueprint. Le résultat de la procédure est un ensemble de conteneur. Lorsque l'ensemble de conteneur est démarré, l'ensemble de serveur eXtreme Scale suit l'ensemble, analyse le fichier XML de serveur et démarre un serveur et un conteneur.

Un ensemble de conteneur peut être éventuellement combiné à l'application et aux plug-ins eXtreme Scale lorsque des mises à jour de plug-ins dynamiques ne sont pas nécessaires ou que les plug-ins ne prennent pas en charge la mise à jour dynamique.

Procédure

1. Créez un fichier XML Blueprint avec l'espace de nom `objectgrid` inclut. Vous pouvez affecter le nom de votre choix au fichier. Toutefois, il doit inclure l'espace de nom Blueprint :

   ```
   <?xml version="1.0" encoding="UTF-8"?>

   <blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
              xmlns:objectgrid="http://www.ibm.com/schema/objectgrid"
              xsi:schemaLocation="http://www.ibm.com/schema/objectgrid
                   http://www.ibm.com/schema/objectgrid/objectgrid.xsd">
   ...
   </blueprint>
   ```

2. Ajoutez la définition XML du serveur eXtreme Scale avec les propriétés de serveur appropriées. Voir le fichier XML descripteur Spring pour plus d'informations sur toutes les propriétés de configuration disponibles. Voir l'exemple suivant de définition de fichier XML :

   ```
   objectgrid:server
       id="xsServer"
      tracespec="ObjectGridOSGi=all=enabled"
      tracefile="logs/osgi/wxsserver/trace.log"
       jmxport="1199"
      listenerPort="2909">
       <objectgrid:catalog host="catserver1.mycompany.com" port="2809" />
       <objectgrid:catalog host="catserver2.mycompany.com" port="2809" />
   </objectgrid:server>
   ```

3. Ajoutez la définition XML du conteneur eXtreme Scale avec la référence à la définition de serveur et les fichiers XML descripteur d'ObjectGrid et de déploiement d'ObjectGrid regroupés dans l'ensemble. Par exemple :

   ```
   <objectgrid:container id="container"
       objectgridxml="/META-INF/objectGrid.xml"
       deploymentxml="/META-INF/objectGridDeployment.xml"
       server="xsServer" />
   ```

4. Stockez le fichier XML Blueprint dans l'ensemble de conteneur. Le fichier XML Blueprint doit être stocké dans le répertoire `OSGI-INF/blueprint` du conteneur Blueprint pour être trouvé.

   Pour stocker le fichier XML Blueprint dans un répertoire différent, vous devez définir l'en-tête du manifeste Bundle-Blueprint. Par exemple :

   ```
   Bundle-Blueprint: OSGI-INF/blueprint.xml
   ```

5. Regroupez les fichiers dans un fichier JAR d'ensemble unique. Voir l'exemple suivant de hiérarchie de répertoires d'ensemble :

```
MyBundle.jar
    /META-INF/manifest.mf
    /META-INF/objectGrid.xml
    /META-INF/objectGridDeployment.xml
    /OSGI-INF/blueprint/blueprint.xml
```

Résultats

Un ensemble de conteneur eXtreme Scale est maintenant créé et peut être installé dans Eclipse Equinox. Lorsque l'ensemble de conteneur est démarré, l'environnement d'exécution du serveur eXtreme Scale dans l'ensemble de serveur eXtreme Scale démarre automatiquement le serveur eXtreme Scale de singleton en utilisant les paramètres définis dans l'ensemble et démarre un serveur de conteneur. L'ensemble peut être arrêté et démarré, ce qui arrête et redémarre le conteneur. Le serveur est un singleton et ne s'arrête pas lorsque l'ensemble est démarré pour la première fois.

Configuration des serveurs avec l'administration de configuration OSGI

Vous pouvez utiliser le service d'administration de configuration (config admin) OSGi pour configurer les serveurs de conteneur WebSphere eXtreme Scale.

Pourquoi et quand exécuter cette tâche

Pour configurer un serveur, le PID (Persistent Identifier) ManagedService, com.ibm.websphere.xs.server, est défini pour faire référence au fichier des propriétés du serveur ObjectGrid dans le système de fichiers. Pour configurer un conteneur, le PID ManagedServiceFactory, com.ibm.websphere.xs.container, est défini pour faire référence au fichier XML de déploiement ObjectGrid et le fichier XML de stratégie de déploiement dans le système de fichiers.

Lorsque les deux PID sont définis dans le service config admin, le service de serveur eXtreme Scale initialise automatiquement le serveur et démarre le conteneur avec les fichiers de configuration spécifiés. Les PID config admin persistent dans le répertoire de configuration OSGi. Si la configuration n'est pas effacée, les paramètres sont conservés entre les redémarrages d'infrastructure.

Il existe plusieurs utilitaires tiers pour définir les propriétés config admin. Voici des exemples d'outils pris en charge par le produit :

- Le client de ligne de commande Luminis OSGi Configuration Admin permet de configurer depuis la ligne de commande.
- Apache Felix File Install permet de définir les paramètres PID config admin dans des fichiers de propriétés standard.

Pour configurer les serveurs de conteneur eXtreme Scale avec le client de ligne de commande OSGi Configuration Administration pour Luminis, procédez comme suit :

Procédure

1. Créez un PID de service géré pour le fichier de propriété de serveur ObjectGrid dans la console OSGi en exécutant les commandes suivantes :

```
osgi> cm create com.ibm.websphere.xs.server
osgi> cm put com.ibm.websphere.xs.server objectgrid.server.props /mypath/server.properties
```

2. Créez un PID (persistence identifier) de fabrique de service géré pour le conteneur ObjectGrid dans la console OSGi en exécutant les commandes suivantes.

   Avertissement : Utilisez le PID créé avec la commande **createf** config admin. Le PID utilisé dans l'exemple suivant n'est fourni qu'à titre d'exemple.

```
osgi> cm createf com.ibm.websphere.xs.container
PID: com.ibm.websphere.xs.container-123456789-0
osgi> cm put com.ibm.websphere.xs.container-123456789-0 objectgridFile /mypath/objectGrid.xml
osgi> cm put com.ibm.websphere.xs.container-123456789-0 deploymentPolicyFile /mypath/deployment.xml
```

Résultats

Maintenant, les serveurs de conteneur eXtreme Scale sont configurés pour démarrer dans une infrastructure OSGi Eclipse Equinox.

Que faire ensuite

Les serveurs de conteneur peuvent être également créés à l'aide d'un programme en utilisant l'ServerFactory et des activateurs de regroupement OSGi. Pour plus d'informations sur l'utilisation de l'API ServerFactory, voir la documentation de l'API.

Chapitre 7. Administration

L'administration et l'exploitation de l'environnement du produit consiste à démarrer et arrêter des serveurs, gérer la disponibilité de la grille de données et récupérer à partir de scénarios de défaillance de centre de données. Une fois que vous avez configuré les serveurs de catalogue et les serveurs de conteneur, vous pouvez démarrer et arrêter les serveurs à l'aide de diverses méthodes. La méthode que vous utilisez pour démarrer et arrêter les serveurs varie selon que vous utilisez une topologie intégrée, une topologie autonome ou une topologie exécutée dans WebSphere Application Server.

Démarrage et arrêt des serveurs sécurisés

Vous pouvez démarrer et arrêter les serveurs de catalogue autonome et de conteneur avec les scripts **startOgServer** et **stopOgServer** ou l'API de serveur intégré.

Avant de commencer

Si vous démarrez ou arrêtez les serveurs dans un environnement autonome qui utilise un fournisseur de sécurité client externe, vous devez définir la variable d'environnement *CLIENT_AUTH_LIB* avant d'exécuter le script **startOgServer** ou **stopOgServer**. Pour plus d'informations sur la définition de cette variable, voir «Démarrage des serveurs sécurisés dans un environnement autonome», à la page 530.

Démarrage des serveurs autonomes

Lorsque vous exécutez une configuration autonome, l'environnement se compose de serveurs de catalogue, de serveurs de conteneur et de processus client. Les serveurs WebSphere eXtreme Scale peuvent être également intégrés à des applications Java existantes en utilisant l'API Embedded Server. Vous devez manuellement configurer et démarrer ces processus.

Avant de commencer

Vous pouvez démarrer des serveurs WebSphere eXtreme Scale dans un environnement dans lequel WebSphere Application Server n'est pas installé. Si vous utilisez WebSphere Application Server, voir «Configuration de WebSphere eXtreme Scale avec WebSphere Application Server», à la page 257.

Démarrage d'un service de catalogue autonome

Vous devez démarrer le service de catalogue manuellement si vous utilisez un environnement WebSphere eXtreme Scale réparti qui n'est pas exécuté dans WebSphere Application Server.

Avant de commencer

- Si vous utilisez WebSphere Application Server, le service de catalogue démarre automatiquement dans les processus existants. Pour plus d'informations, voir Démarrage du processus du service de catalogue dans un environnement WebSphere Application Server.

Pourquoi et quand exécuter cette tâche

Démarrage du serveur de catalogue avec le script **startOgServer**. Lorsque vous appelez la commande de démarrage, utilisez le script **startOgServer.sh** sur les plateformes Unix ou **startOgServer.bat** sous Windows.

Le service de catalogue peut être exécuté dans un seul processus ou il peut inclure plusieurs serveurs de catalogue afin de constituer un domaine de services de catalogue. Un domaine de services de catalogue est obligatoire dans un environnement de production pour la haute disponibilité. Pour plus d'informations sur les domaines de services de catalogue, voir les informations sur les domaine de services de catalogue dans *Présentation du produit*. Vous pouvez également spécifier des paramètres supplémentaires au script pour associer l'ORB (Object Request Broker) à un hôte et un port spécifiques, spécifier le domaine ou activer la sécurité.

Procédure

- **Démarrez un processus de serveur de catalogue.**

  Pour démarrer un serveur de catalogues, entrez les commandes suivantes à partir de la ligne de commande :

  1. Accédez au répertoire `bin`.

     ```
     cd objectgridRoot/bin
     ```

  2. Exécutez la commande **startOgServer**.

     ```
     startOgServer.bat|sh catalogServer
     ```

  Pour une liste de tous les paramètres de ligne de commande disponibles, voir «Script **startOgServer**», à la page 401. N'utilisez pas une seule machine virtuelle Java (JVM) pour exécuter le service de catalogue dans un environnement de production. Si le service de catalogue échoue, aucun nouveau client ne peut être acheminé vers l'instance eXtreme Scale déployée et aucune nouvelle instance ObjectGrid ne peut être ajoutée au domaine. Pour ces motifs, vous devez démarrer un ensemble de machines virtuelles Java pour pouvoir exécuter un domaine de services de catalogue.

- **Démarrez un domaine de services de catalogue constitué de plusieurs noeud finaux.**

  Pour démarrer un ensemble de serveurs afin d'exécuter un service de catalogue, vous devez utiliser l'option **-catalogServiceEndPoints** sur le script `startOgServer`. Cet argument accepte une liste de noeuds finaux de services de catalogue dans le format *serverName*:*hostName*:*clientPort*:*peerPort*. L'exemple suivant indique comment démarrer la première des trois machines virtuelles Java pour héberger un service de catalogue :

  1. Accédez au répertoire bin.

     ```
     cd racine_install_wxs/bin
     ```

  2. Exécutez la commande **startOgServer**.

     ```
     startOgServer.bat|sh cs1 -catalogServiceEndPoints
     cs1:MyServer1.company.com:6601:6602,
     cs2:MyServer2.company.com:6601:6602,
     cs3:MyServer3.company.com:6601:6602
     ```

     Dans ce cas, le serveur `cs1` de l'hôte `MyServer1.company.com` est démarré. Ce nom de serveur est le premier argument transmis au script. Lors de l'initialisation du serveur `cs1`, les paramètres catalogServiceEndpoints sont examinés pour déterminer les ports alloués pour ce processus. La liste est également utilisée pour permettre au serveur `cs1` d'accepter les connexions des autres serveurs : `cs2` et `cs3`.

3. Pour démarrer les serveurs de catalogues restants de la liste, transmettez les arguments ci-après au script startOgServer. Démarrage du serveur `cs2` sur l'hôte `MyServer2.company.com`.

   ```
   startOgServer.bat|sh cs2 -catalogServiceEndPoints
   cs1:MyServer1.company.com:6601:6602,
   cs2:MyServer2.company.com:6601:6602,
   cs3:MyServer3.company.com:6601:6602
   ```

   Démarrage du serveur `cs3` sur `MyServer3.company.com` :

   ```
   startOgServer.bat|sh cs3 -catalogServiceEndPoints
   cs1:MyServer1.company.com:6601:6602,
   cs2:MyServer2.company.com:6601:6602,
   cs3:MyServer3.company.com:6601:6602
   ```

   Important : Démarrez au moins deux serveurs de catalogue en même temps.

   Vous devez démarrer les serveurs de catalogues qui se trouvent dans une grille de données en parallèle, car chaque serveur s'interrompt pour attendre que les autres serveurs de catalogues rejoignent le groupe central. Un serveur de catalogue qui est configuré pour une grille de données ne démarre pas tant qu'il n'a pas identifié les autres membres du groupe. Le serveur de catalogues arrive à expiration si aucun autre serveur ne devient disponible.

- **Liez l'ORB à un hôte et un port spécifiques.**

  En dehors des ports définis dans l'argument **`catalogServiceEndpoints`**, chaque service de catalogue utilise également un ORB (Object Request Broker) pour accepter les connexions des clients et des conteneurs. Par défaut, l'ORB écoute sur le port `2809` du système hôte local. Si vous souhaitez associer l'ORB à un hôte et un port spécifiques sur la machine virtuelle Java d'un service de catalogue, utilisez les arguments **`-listenerHost`** et **`-listenerPort`**. L'exemple suivant montre comment démarrer le serveur de catalogue d'une machine virtuelle Java unique avec son ORB associé au port `7000` sur `MyServer1.company.com` :

  ```
  startOgServer.sh catalogServer -listenerHost MyServer1.company.com
  -listenerPort 7000
  ```

  Chacun des conteneurs et clients eXtreme Scale doit être fourni avec des données de point de contact d'ORB de service de catalogue. Les clients n'ont besoin que d'un sous-ensemble de ces données, mais vous devez utiliser au moins deux points de contact pour la haute disponibilité.

- Facultatif : **Nom du domaine de services de catalogue**

  Un nom de domaine de services de catalogue n'est pas requis lors du démarrage d'un service de catalogue. Toutefois, si vous utilisez la réplication multimaître ou plusieurs domaine de services de catalogue dans un même ensemble de processus, vous devez définir un nom de domaine de services de catalogue unique. Le nom de domaine par défaut est `DefaultDomain`. Pour affecter un nom à votre domaine, utilisez l'option **`-domain`**. L'exemple ci-après montre comment démarrer la machine virtuelle Java d'un service de catalogue avec le nom de domaine `myDomain`.

  ```
  startOgServer.sh catalogServer -domain myDomain
  ```

  Pour plus d'informations sur la configuration de la réplication multimaître, voir «Configuration de plusieurs topologies de centres de données», à la page 281.

- **Démarrez un service de catalogue sécurisé.** Pour plus d'informations, voir «Démarrage des serveurs sécurisés dans un environnement autonome», à la page 530.
- **Démarrez le service de catalogue à l'aide d'un programme.**

  Tout paramètre JVM qui est marqué par la méthode CatalogServerProperties.setCatalogServer peut héberger le service de catalogue pour eXtreme Scale. Cette méthode indique à l'environnement d'exécution du serveur eXtreme Scale d'instancier le service de catalogue lorsque le serveur est démarré. Le code qui suit montre comment instancier le serveur de catalogue eXtreme Scale :

  ```
  CatalogServerProperties catalogServerProperties =
   ServerFactory.getCatalogProperties();
  catalogServerProperties.setCatalogServer(true);

  //La méthode getInstance() démarre le service de catalogue.
  Server server = ServerFactory.getInstance();
  ```

  Pour plus d'informations sur le démarrage des serveurs à l'aide d'un programme, voir «Utilisation de l'API de serveur embarqué pour démarrer et arrêter les serveurs», à la page 410.

Démarrage des serveurs de conteneur

Vous pouvez démarrer les serveurs de conteneur depuis la ligne de commande en utilisant une topologie de déploiement ou un fichier `server.properties`.

Pourquoi et quand exécuter cette tâche

Pour démarrer un processus de conteneur, vous avez besoin d'un fichier ObjectGrid XML. Ce fichier spécifie quels serveurs eXtreme Scale sont hébergés par le conteneur. Vérifiez que votre conteneur est équipé pour héberger chaque ObjectGrid dans le fichier XML que vous lui transmettez. Toutes les classes que ces ObjectGrids requièrent doivent se trouver dans le chemin d'accès aux classes pour le conteneur. Pour plus d'informations sur le fichier XMLObjectGrid, voir Fichier `objectGrid.xsd`.

Procédure

- **Démarrez le processus de conteneur depuis la ligne de commande.**
  1. A partir de la ligne debvommande, accédez au répertoire bin :

     ```
     cd racine_install_wxs/bin
     ```

  2. Exécutez la commande suivante :

     ```
     startOgServer.sh c0 -objectGridFile ../xml/companyGrid.xml
     -catalogServiceEndPoints MyServer1.company.com:2809
     ```

  Important : Sur le conteneur, l'option **-catalogServiceEndPoints** est utilisée pour référencer l'hôte et le port de la fonction ORB sur le service de catalogue. Le service de catalogue utilise les options **-listenerHost** et **-listenerPort** pour spécifier l'hôte et le port de la fonction ORB ou accepte la liaison par défaut. Lorsque vous démarrez un conteneur, utilisez l'option **-catalogServiceEndPoints** pour référencer les valeurs transmises aux options **-listenerHost** et **-listenerPort** sur le service de catalogue. Si les options **-listenerHost** et **-listenerPort** ne sont pas utilisées quand le service de catalogue est démarré, la fonction ORB est liée au port 2809 sur le système hôte local pour le service de catalogue. N'utilisez pas l'option **-catalogServiceEndPoints** pour référencer les hôtes et les ports transmis à l'option **-catalogServiceEndPoints** sur le service de catalogue. Sur le service de catalogue, l'option **-catalogServiceEndPoints** est utilisée pour spécifier les ports nécessaires pour une configuration de serveur

statique.
Ce processus est identifié par c0, le premier argument transmis au script. Utilisez le fichier companyGrid.xml pour démarrer le conteneur. Si votre fonction ORB de serveur de catalogue est exécutée sur un hôte différent que celui du conteneur ou qu'elle utilise un autre port que celui par défaut, vous devez utiliser l'argument **-catalogServiceEndPoints** pour vous connecter à la fonction ORB. Pour cet exemple, partez du principe qu'un unique service de catalogue est exécuté sur le port 2809 sur MyServer1.company.com

- **Démarrez le conteneur à l'aide d'une règle de déploiement.**

  Sans être nécessaire, une stratégie est recommandée pendant le démarrage du conteneur. La règle de déploiement est utilisée pour configurer le partitionnement et la réplication pour eXtreme Scale. La règle de déploiement peut également être utilisée pour influencer le comportement de positionnement. Comme l'exemple précédent ne fournit pas de fichier de règle de déploiement, l'exemple reçoit toutes les valeurs par défaut en ce qui concerne la réplication, le partitionnement et le positionnement. Donc, les mappes dans le CompanyGrid se trouvent dans un mapSet. Le mapSet n'est ni partitionné ni répliqué. Pour plus d'informations sur les fichiers de règle de déploiement, voir Fichier XML du descripteur de la règle de déploiement. L'exemple suivant utilise le fichier companyGridDpReplication.xml pour démarrer une machine virtuelle Java de conteneur, c0 :

  1. A partir de la ligne de commande, accédez au répertoire bin :

     ```
     cd racine_install_wxs/bin
     ```

  2. Exécutez la commande suivante :

     ```
     startOgServer.sh c0 -objectGridFile ../xml/companyGrid.xml
     -deploymentPolicyFile ../xml/companyGridDpReplication.xml
     -catalogServiceEndPoints MyServer1.company.com:2809
     ```

  Remarque : Si vous avez des classes Java stockées dans un répertoire spécifique, au lieu d'altérer le script StartOgServer, vous pouvez lancer le serveur avec des arguments, comme suit : -jvmArgs -cp C:\ . . . \DirectoryPOJOs\POJOs.jar . Dans le fichier companyGridDpReplication.xml, un seul groupe de mappes contient toutes les mappes. Ce mapSet est divisé en 10 partitions. Chaque partition a une réplique synchrone et aucune réplique asynchrone. Tout conteneur utilisant la règle de déploiement companyGridDpReplication.xml combinée au fichier XML ObjectGrid companyGrid.xml est également capable d'héberger des fragments de CompanyGrid. Démarrez une autre machine virtuelle Java de conteneur, c1 :

  1. A partir de la ligne de commande, accédez au répertoire bin :

     ```
     cd racine_install_wxs/bin
     ```

  2. Exécutez la commande suivante :

     ```
     startOgServer.sh c1 -objectGridFile ../xml/companyGrid.xml
     -deploymentPolicyFile ../xml/companyGridDpReplication.xml
     -catalogServiceEndPoints MyServer1.company.com:2809
     ```

  Chaque règle de déploiement contient au moins un élément objectgridDeployment. Quand un conteneur est démarré, il publie sa règle de déploiement sur le service de catalogue. Le service de catalogue examine chaque élément objectgridDeployment. Si l'attribut objectgridName correspond à l'attribut objectgridName d'un élément objectgridDeployment précédemment reçu, l'élément objectgridDeployment le plus récent est ignoré. Le premier élément objectgridDeployment reçu pour un attribut objectgridName spécifique est utilisé comme élément maître. Par exemple, partons du principe que la machine virtuelle Java c2 utilise une règle de déploiement qui divise le mapSet en nombre différent de partitions :

companyGridDpReplicationModified.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy
  ../deploymentPolicy.xsd"
    xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

    <objectgridDeployment objectgridName="CompanyGrid">
        <mapSet name="mapSet1" numberOfPartitions="5"
            minSyncReplicas="1" maxSyncReplicas="1"
            maxAsyncReplicas="0">
            <map ref="Customer" />
            <map ref="Item" />
            <map ref="OrderLine" />
            <map ref="Order" />
        </mapSet>
    </objectgridDeployment>

</deploymentPolicy>
```

Vous pouvez maintenant démarrer une troisième machine virtuelle Java, c2 :

1. A partir de la ligne de commande, accédez au répertoire bin :

   ```
   cd racine_install_wxs/bin
   ```

2. Exécutez la commande suivante :

   ```
   startOgServer.sh c2 -objectGridFile ../xml/companyGrid.xml
   -deploymentPolicyFile ../xml/companyGridDpReplicationModified.xml
   -catalogServiceEndPoints MyServer1.company.com:2809
   ```

Le conteneur sur la machine virtuelle Java c2 est démarré avec une règle de déploiement qui spécifie 5 partitions pour le mapSet1. Cependant, le service de catalogue contient déjà la copie maître de l'objectgridDeployment pour le CompanyGrid. Quand la machine virtuelle Java c0 a été démarrée, elle a spécifié que 10 partitions existent pour ce mapSet. Comme il s'agit du premier conteneur à démarrer et publier sa règle de déploiement, cette dernière devient la stratégie maître. En conséquence, toute valeur d'attribut objectgridDeployment égale à `CompanyGrid` dans une règle de déploiement suivante est ignorée.

- **Démarrez un conteneur à l'aide d'un fichier de propriétés de serveur.**

  Vous pouvez utiliser un fichier de propriétés de serveur pour configurer la fonction de trace et la sécurité sur un conteneur. Exécutez les commandes suivantes pour démarrer un conteneur c3 avec un fichier de propriétés de serveur.

  1. A partir de la ligne de commande, accédez au répertoire bin :

     ```
     cd racine_install_wxs/bin
     ```

  2. Exécutez la commande suivante :

     ```
     startOgServer.sh c3 -objectGridFile ../xml/companyGrid.xml
     -deploymentPolicyFile ../xml/companyGridDpReplicationModified.xml
     -catalogServiceEndPoints MyServer1.company.com:2809
     -serverProps ../serverProps/server.properties
     ```

  Voici un exemple de fichier `server.properties` :

  server.properties

  ```
  workingDirectory=
  traceSpec=*=all=disabled
  systemStreamToFileEnabled=true
  enableMBeans=true
  memoryThresholdPercentage=50
  ```

  Il s'agit d'un fichier de propriétés de serveur de base dans lequel la sécurité n'est pas activée. Pour plus d'informations concernant le fichier `server.properties`, voir Fichier de propriétés du serveur.

- **Démarrez un serveur de conteneur à l'aide d'un programme.**

  Pour plus d'informations sur le démarrage des serveurs de conteneur à l'aide d'un programme, voir «Utilisation de l'API de serveur embarqué pour démarrer et arrêter les serveurs», à la page 410.

Script `startOgServer`

Le script **`startOgServer`** arrête les serveurs de catalogue et de conteneur. Vous pouvez utiliser divers paramètres lorsque vous démarrez vos serveurs pour activer la trace, spécifiez des numéros de port, etc.

Rôle

Vous pouvez utiliser le script **`startOgServer`** pour démarrer les serveurs.

Placement

Le script **`startOgServer`** se trouve dans le répertoire `bin` du répertoire root, par exemple :

```
cd racine_install_wxs/bin
```

Remarque : Si des classes Java sont stockées dans un répertoire spécifique, au lieu de modifier le script startOgServer, vous pouvez lancer le serveur avec des arguments, comme suit : `-jvmArgs -cp C:\ . . . \DirectoryPOJOs\POJOs.jar`

.

Syntaxe des serveurs de catalogues

Pour démarrer un serveur de catalogues :

Windows

```
startOgServer.bat <server> [options]
```

UNIX

```
startOgServer.sh <server>[options]
```

Pour démarrer un serveur de catalogues configuré par défaut, utilisez les commandes suivantes :

Windows

```
startOgServer.bat catalogServer
```

UNIX

```
startOgServer.sh catalogServer
```

Options de démarrage des serveurs de catalogues

Les paramètres suivants sont tous facultatifs.

Paramètres de démarrage d'un serveur de catalogues :

`-catalogServiceEndPoints <serverName:hostName:clientPort:peerPort>`
Dans le conteneur, fait référence à l'hôte et au port ORB (Object Request Broker) dans le service de catalogue. Chaque attribut est défini comme suit :

serverName
Spécifie un nom permettant d'identifier le processus que vous lancez.

hostName
Spécifie le nom d'hôte de l'ordinateur sur lequel le serveur est lancé.

clientPort
Spécifie le port utilisé pour la communication de service de catalogue homologue.

peerPort
Cette value est identique à haManagerPort. Spécifie le port utilisé pour la communication de service de catalogue homologue.

L'exemple suivant démarre le serveur de catalogue, cs1, qui se trouve dans le même domaine de services de catalogue que les serveurs cs2 et cs3 :

```
startOgServer.bat|sh cs1 -catalogServiceEndPoints
cs1:MyServer1.company.com:6601:6602,
cs2:MyServer2.company.com:6601:6602,
cs3:MyServer3.company.com:6601:6602
```

-clusterSecurityFile <fichier_xml_sécurité_cluster>
Indique le fichier `objectGridSecurity.xml` sur le disque dur, qui décrit les propriétés de sécurité communes à tous les serveurs (y compris les serveurs de catalogue et les serveurs de conteneur). L'un des exemples de propriété est la configuration de l'authentificateur qui représente le registre d'utilisateurs et le mécanisme d'authentification.

Exemple :`/opt/xs/ogsecurity.xml`

-clusterSecurityUrl <URL du xml de la sécurité du cluster>
Indique le fichier `objectGridSecurity.xml` comme URL du fichier sur le disque dur ou sur le réseau, qui décrit les propriétés de sécurité communes à tous les serveurs, y compris les serveurs de catalogue et les serveurs de conteneur. L'un des exemples de propriété est la configuration de l'authentificateur qui représente le registre d'utilisateurs et le mécanisme d'authentification.

Exemple :`file:///opt/xs/ogsecurity.xml`

-domain <nom de domaine>
Indique le nom du domaine de services de catalogue du serveur de catalogue. Le domaine de services de catalogue crée un groupe de serveurs de catalogue à haute disponibilité. Chaque serveur de catalogue pour un seul domaine doit spécifier la même valeur pour le paramètre **-domain**.

-JMXConnectorPort <port>
Définit le port SSL (Secure Sockets Layer) auquel se connecte le service Java Management Extensions (JMX).

-haManagerPort <port>
Synonyme avec port homologue. Indique le numéro de port utilisé par le gestionnaire de haute disponibilité. Si cette propriété n'est pas définie, le service de catalogue génère automatiquement un port disponible. Cette propriété s'applique à la fois au serveur conteneur et au service de catalogue. (Requis pour les environnements WebSphere Application Server uniquement.)

-JMXServicePort <port>
Spécifie le numéro du port sur lequel le serveur MBean écoute les communications avec Java Management Extensions (JMX). Vous devez utiliser un numéro de port différent pour chaque machine virtuelle Java dans votre configuration. Si vous voulez utiliser JMX/RMI, définissez explicitement

l'option **-JMXServicePort** et le numéro de port, même si vous souhaitez utiliser la valeur de port par défaut. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue.

Valeur par défaut : `1099`

-jvmArgs <arguments de la machine virtuelle Java>
Indique un ensemble d'arguments de machine virtuelle Java. Chaque option après l'option **-jvmArgs** est utilisée pour démarrer la machine JVM (Java virtual machine) du serveur. Si le paramètre **-jvmArgs** est utilisé, vérifiez qu'il s'agit du dernier argument de script facultatif spécifié.

Exemple :-jvmArgs `-Xms256M -Xmx1G`

-listenerHost <nom d'hôte>

Indique le nom d'hôte auquel l'ORB (Object Request Broker) se connecte pour communiquer avec IIOP (Internet Inter-ORB Protocol). La valeur doit être un nom qualifié complet de domaine ou une adresse IP. Si la configuration implique plusieurs cartes réseau, configurez l'hôte du programme d'écoute et le port d'écoute pour que l'ORB (Object Request Broker) dans la machine JVM connaisse l'adresse IP à laquelle se connecter. Si vous ne définissez pas l'adresse IP à utiliser, des symptômes (délais de connexion, défaillances inhabituelles d'API et clients qui semblent se bloquer) apparaissent.**Valeur par défaut :** `localhost`

-listenerPort <port>
Indique le numéro de port auquel se connecte l'ORB (Object Request Broker). Ce paramètre configure les conteneurs et les clients pour communiquer avec le service de catalogue via l'ORB. Dans WebSphere Application Server, le port d'écoute est hérité par la configuration de port BOOTSTRAP_ADDRESS. Cette propriété s'applique au serveur de conteneur et au service de catalogue.**Valeur par défaut :** `2809`

-quorum true|false
Active le quorum sur le serveur de catalogue. Pour plus d'informations, voir Quorums de serveurs de catalogue.

-script <fichier script>
Indique l'emplacement d'un script personnalisé pour les commandes que vous spécifiez pour démarrer les serveurs de catalogue ou les conteneurs, puis définir des paramètres ou effectuer des modifications en fonction des besoins.

-serverProps <fichier de propriétés du serveur>
Indique le fichier de propriétés du serveur qui contient les propriétés de sécurité spécifiques au serveur. Le nom de fichier spécifié pour cette propriété correspond simplement à un chemin classique, tel que `c:/tmp/og/catalogserver.props`.

-traceSpec <spécification de la trace>
Indique une chaîne qui spécifie la portée de la trace qui est activée au démarrage du serveur.

Exemple :

- `ObjectGrid=all=enabled`
- `ObjectGrid*=all=enabled`

-traceFile <fichier de trace>
Indique le chemin d'un fichier dans lequel les informations de trace doivent être sauvegardées.

Exemple : `../logs/c4Trace.log`

-timeout <secondes>
Indique un nombre de secondes avant que le démarrage du serveur n'arrive à expiration.

Syntaxe des serveurs conteneurs Windows

```
startOgServer.bat <serveur> -objectgridFile <fichier xml>
-deploymentPolicyFile <fichier xml> [options]
```

Windows

```
startOgServer.bat <serveur> -objectgridUrl <URL du xml>
-deploymentPolicyUrl <URL du xml> [options]
```

UNIX

```
startOgServer.sh <server> -objectgridFile <fichier xml>
-deploymentPolicyFile <fichier xml> [options]
```

UNIX

```
 startOgServer.sh <serveur> -objectgridUrl <URL du xml>
-deploymentPolicyUrl <URL du xml> [options]
```

Options des serveurs conteneurs

-catalogServiceEndPoints<nomHôte:port,nomHôte:port>
Spécifie l'hôte ORB (Object Request Broker) et le numéro de port dans le service de catalogue.

Valeur par défaut : localhost:2809

-deploymentPolicyFile <fichier xml de la règle de déploiement>
Indique le chemin d'accès au fichier de la stratégie de déploiement sur le disque dur. La stratégie de déploiement est utilisée pour configurer le partitionnement et la réplication. La règle de déploiement peut également être utilisée pour influencer le comportement de positionnement.

Exemple : `../xml/SimpleDP.xml`

-deploymentPolicyUrl <url de la règle de déploiement>
Indique l'URL du fichier de la stratégie de déploiement sur le disque dur ou sur le réseau. La stratégie de déploiement est utilisée pour configurer le partitionnement et la réplication. La règle de déploiement peut également être utilisée pour influencer le comportement de positionnement.

Exemple : `file://xml/SimpleDP.xml`

-JMXConnectorPort <port>
Définit le port SSL (Secure Sockets Layer) auquel se connecte le service Java Management Extensions (JMX).

-JMXServicePort <port>

Spécifie le numéro du port sur lequel le serveur MBean écoute les communications avec Java Management Extensions (JMX). Vous devez utiliser un numéro de port différent pour chaque machine virtuelle Java dans votre configuration. Si vous voulez utiliser JMX/RMI, définissez explicitement l'option **-JMXServicePort** et le numéro de port, même si vous souhaitez utiliser la valeur de port par défaut. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. **Valeur par défaut :** `1099`

-jvmArgs <arguments de la machine virtuelle Java>
Indique un ensemble d'arguments de machine virtuelle Java. Chaque option

après l'option **-jvmArgs** est utilisée pour démarrer la machine JVM (Java virtual machine) du serveur. Si le paramètre **-jvmArgs** est utilisé, vérifiez qu'il s'agit du dernier argument de script facultatif spécifié.

Exemple :-jvmArgs `-Xms256M -Xmx1G`

-listenerHost <nom d'hôte>

Indique le nom d'hôte auquel l'ORB (Object Request Broker) se connecte pour communiquer avec IIOP (Internet Inter-ORB Protocol). La valeur doit être un nom qualifié complet de domaine ou une adresse IP. Si la configuration implique plusieurs cartes réseau, configurez l'hôte du programme d'écoute et le port d'écoute pour que l'ORB (Object Request Broker) dans la machine JVM connaisse l'adresse IP à laquelle se connecter. Si vous ne définissez pas l'adresse IP à utiliser, des symptômes (délais de connexion, défaillances inhabituelles d'API et clients qui semblent se bloquer) apparaissent.**Valeur par défaut :** `localhost`

-listenerPort <port>

Indique le numéro de port auquel se connecte l'ORB (Object Request Broker). Ce paramètre configure les conteneurs et les clients pour communiquer avec le service de catalogue via l'ORB. Dans WebSphere Application Server, le port d'écoute est hérité par la configuration de port BOOTSTRAP_ADDRESS. Cette propriété s'applique au serveur de conteneur et au service de catalogue.**Valeur par défaut :** `2809`

-objectgridFile <Fichier XML du descripteur d'ObjectGrid>

Indique le chemin d'accès au fichier du descripteur d'ObjectGrid. Le fichier XML ObjectGrid spécifie les serveurs eXtreme Scale hébergés par le conteneur.

-objectgridUrl <URL du descripteur d'ObjectGrid>

Spécifie une URL pour le fichier descripteur ObjectGrid. Le fichier XML ObjectGrid spécifie les serveurs eXtreme Scale hébergés par le conteneur.

-script <fichier script>

Indique l'emplacement d'un script personnalisé pour les commandes que vous spécifiez pour démarrer les serveurs de catalogue ou les conteneurs, puis définir des paramètres ou effectuer des modifications en fonction des besoins.

-serverProps <fichier de propriétés du serveur>

Indique le chemin d'accès au fichier de propriétés du serveur.

Exemple :`../security/server.props`

-timeout <secondes>

Indique un nombre de secondes avant que le démarrage du serveur n'arrive à expiration.

-traceFile <fichier de trace>

Indique le chemin d'un fichier dans lequel les informations de trace doivent être sauvegardées.

Exemple : `../logs/c4Trace.log`

-traceSpec <spécification de la trace>

Indique une chaîne qui spécifie la portée de la trace qui est activée au démarrage du serveur.

Exemple :

- `ObjectGrid=all=enabled`
- `ObjectGrid*=all=enabled`

-zone <nom de zone>
Indique la zone à utiliser pour tous les conteneurs du serveur. Voir les «Routage par zone préférée», à la page 243 informations sur les zones dans *Présentation du produit* pour plus d'informations sur la configuration de zones.

Arrêt des serveurs autonomes

Utilisez le script `stopOgServer` pour arrêter les processus serveur eXtreme Scale.

Pourquoi et quand exécuter cette tâche

Exécutez le script **stopOgServer** en accédant au répertoire `bin` :

```
cd racine_install_wxs/bin
```

Procédure

- **Arrêtez un serveur de conteneur.**

  Utilisez le script **stopOgServer** pour arrêter le serveur de conteneur. Utilisez cette commande uniquement lorsque vous arrêtez un seul serveur de conteneur. Si vous exécutez la commande d'arrêt du serveur de catalogue unique sur plusieurs serveurs de conteneur à la suite, des problèmes de performances et de désabonnement se produisent pour le placement des fragments.

  ```
  stopOgServer containerServer -catalogServiceEndPoints MyServer1.company.com:2809
  ```

  Avertissement : L'option **-catalogServiceEndPoints** doit correspondre à la valeur de l'option **-catalogServiceEndPoints** utilisée pour démarrer le conteneur. S'il n'a pas été fait usage de **-catalogServiceEndPoints** pour démarrer le conteneur, les valeurs par défaut seront probablement localhost ou le nom d'hôte et 2809 pour le port ORB de connexion au service de catalogue. Autrement, utilisez les valeurs envoyées à **-listenerHost** et **-listenerPort** dans le service de catalogue. Si les options **-listenerHost** et **-listenerPort** ne sont pas utilisées lors du démarrage du service de catalogue, l'ORB se lie au port 2809 sur localhost pour le service de catalogue.

- **Arrêtez plusieurs serveurs de conteneur.**

  Pour éviter les problèmes de //désabonnement et performances pour le placement des fragments lorsque vous voulez arrêter plusieurs serveurs de conteneur en même temps, utilisez la syntaxe de commande suivante. Séparez les serveurs de conteneur avec une virgule :

  ```
  stopOgServer containerServer0,containerServer1,containerServer2
  -catalogServiceEndPoints MyServer1.company.com:2809
  ```

  Si vous souhaitez arrêter tous les conteneurs dans une zone ou un hôte, vous pouvez utiliser le paramètre **-teardown**. Pour plus d'informations, voir «Arrêt propre des serveurs avec l'utilitaire **xscmd**», à la page 409.

- **Arrêtez les serveurs de catalogue.**

  Exécutez le script **stopOgServer** pour arrêter le serveur de catalogue.

  ```
  stopOgServer.sh serveurCatalogue -catalogServiceEndPoints MyServer1.company.com:2809
  ```

Avertissement : Lorsque vous arrêtez un service de catalogue, utilisez l'option **-catalogServiceEndPoints** pour référencer l'hôte ORB (Object Request Broker) et le port dans le service de catalogue. Le service de catalogue utilise les options **-listenerHost** et **-listenerPort** pour spécifier l'hôte et le port pour la liaison ORB ou accepte la liaison par défaut. Si les options **-listenerHost** et **-listenerPort** ne sont pas utilisées quand le service de catalogue est démarré, la fonction ORB est liée au port 2809 sur le système hôte local pour le service de catalogue. L'option **-catalogServiceEndPoints** pour arrêter un service de catalogue est différente de l'option que vous avez utilisée pour le démarrer.

Démarrer un service de catalogue requiert des ports d'accès homologues et des ports d'accès clients si les ports par défaut n'ont pas été utilisés. En revanche, l'arrêt d'un service de catalogue ne requiert que le port de l'ORB.

- **Arrêtez le serveur de la console Web.** Pour arrêter le serveur de la console Web, exécutez le script **stopConsoleServer.bat|sh**. Ce script se trouve dans le répertoire *racine_install_wxs*/ObjectGrid/bin de votre installation. Pour plus d'informations, voir «Démarrage et consignation sur la console Web», à la page 443.
- **Activez la fonction de trace pour le processus d'arrêt du serveur.**

  Si un conteneur ne parvient pas à s'arrêter, vous pouvez activer la fonction de trace pour vous aider dans le débogage du problème. Pour activer la fonction de trace au cours de l'arrêt d'un serveur, ajoutez les paramètres **-traceSpec** et **-traceFile** pour les commandes d'arrêt. Le paramètre **-traceSpec** spécifie le type de trace et le paramètre **-traceFile** spécifie le chemin d'accès et le nom du fichier à créer et à utiliser pour les données de trace.

  1. A partir de la ligne de commande, accédez au répertoire bin.

     ```
     cd racine_install_wxs/bin
     ```

  2. Exécutez le script **stopOgServer** avec la fonction de trace activée.

     ```
     stopOgServer.sh c4 -catalogServiceEndPoints MyServer1.company.com:2809
      -traceFile ../logs/c4Trace.log -traceSpec ObjectGrid=all=enabled
     ```

     Une fois la trace obtenue, recherchez les erreurs relatives aux conflits de ports, aux classes manquantes, aux fichiers XML manquants ou incorrects, ou toute trace de pile. Suggestions de spécifications de trace au démarrage :

     - `ObjectGrid=all=enabled`
     - `ObjectGrid*=all=enabled`

     Pour connaître toutes les options de spécification de trace, voir «Options de trace», à la page 537.
- **Arrêtez les serveurs embarqués à l'aide d'un programme.**

  Pour plus d'informations sur cette opération, voir «Utilisation de l'API de serveur embarqué pour démarrer et arrêter les serveurs», à la page 410.

Script stopOgServer

Le script **stopOgServer** arrête les serveurs de catalogue et de conteneur.

Rôle

Utilisez le script **stopOgServer** pour arrêter un serveur. Vous devez indiquer le nom du serveur et ses noeuds finaux de service de catalogue.

Emplacement

Le script **stopOgServer** se trouve dans le répertoire `bin` du répertoire `root` , par exemple :

```
cd racine_install_wxs/bin
```

Syntaxe

Pour arrêter un serveur de catalogue ou de conteneur : Windows

```
stopOgServer.bat <server_name> -catalogServiceEndPoints
<csHost:csListenerPort,csHost:csListenerPort> [options]
```

UNIX

```
stopOgServer.sh <server_name> -catalogServiceEndPoints
<csHost:csListenerPort,csHost:csListenerPort> [options]
```

Options

-catalogServiceEndPoints <csHost:csListenerPort, csHost:csListenerPort...>
Spécifie l'hôte ORB (Object Request Broker) et le numéro de port.

Pour les serveurs de conteneur : la liste des noeuds finaux du service de catalogue doit être identique à la liste qui a été utilisé pour démarrer le serveur de conteneur. Si vous n'avez pas spécifié cette option lorsque vous avez démarré le serveur de conteneur, utilisez la valeur par défaut `localhost:2809`.

Pour les serveurs de catalogues : si vous arrêtez le service de catalogue, utilisez les valeurs que vous avez indiquées pour les options **-listenerHost** et **-listenerPort** lorsque vous avez démarré le service de catalogue. Si vous n'avez pas spécifié ces options lorsque vous avez démarré le serveur de catalogue, utilisez la valeur par défaut `localhost:2809`. La valeur **-catalogServiceEndPoints** que vous utilisez lorsque vous arrêtez le service de catalogue est différente lorsque vous démarrez le service de catalogue.

-clientSecurityFile <fichier de propriétés du serveur>
Indique le chemin d'accès au fichier de propriétés client qui définit les propriétés de sécurité du client. Voir Fichier de propriétés du client pour plus d'informations sur les paramètres de sécurité dans ce fichier.

-traceSpec <spécification de la trace>
Indique une chaîne qui spécifie la portée de la trace qui est activée au démarrage du serveur.

Exemple :

- `ObjectGrid=all=enabled`
- `ObjectGrid*=all=enabled`

-traceFile <fichier de trace>
Indique le chemin d'un fichier dans lequel les informations de trace doivent être sauvegardées.

Exemple : `../logs/c4Trace.log`

-jvmArgs <arguments de la machine virtuelle Java>
Indique un ensemble d'arguments de machine virtuelle Java. Chaque option après l'option **-jvmArgs** est utilisée pour démarrer la machine JVM (Java virtual machine) du serveur. Si le paramètre **-jvmArgs** est utilisé, vérifiez qu'il s'agit du dernier argument de script facultatif spécifié.

Exemple :-jvmArgs `-Xms256M -Xmx1G`

Arrêt propre des serveurs avec l'utilitaire xscmd

Vous pouvez utiliser l'utilitaire **inattendue** avec la commande **-c teardown** pour arrêter une liste ou un groupe de serveurs de catalogue et de conteneur. Cette commande simplifie l'arrêt d'une partie ou de la totalité d'une grille de données en évitant que le service de catalogue ait a exécuter des actions de placement et de récupération inutiles qui sont généralement exécutées lorsque des processus sont arrêtés.

Procédure

- Arrêter les serveurs d'une liste de serveurs.

  Fournissez une liste de serveurs après le paramètre **-teardown** :

  ```
  xscmd –c teardown
  ```

- Arrêter tous les serveurs dans une zone donnée.

  Utilisez le paramètre **-z** et fournissez le nom de la zone. Le serveur de catalogue détermine les serveurs qui sont en cours d'exécution dans la zone et l'utilitaire **xscmd** demande une liste de serveurs dans la zone sélectionnée pour arrêter les serveurs :

  ```
  xscmd –c teardown –z zone_name
  ```

- Arrêter tous les serveurs sur un hôte.

  Utilisez le paramètres **-hf** et fournissez le nom de l'hôte. Par exemple, pour arrêter tous les serveurs sur `myhost.mycompany.com`, entrez `-hf myhost.mycompany.com`. Le serveur de catalogue détermine les serveurs qui sont en cours d'exécution sur l'hôte et l'utilitaire **xscmd** demande une liste de serveurs sur hôte sélectionné pour arrêter les serveurs :

  ```
  xscmd –teardown –hf <host_name>
  ```

Démarrage et arrêt des serveurs dans un environnement WebSphere Application Server

Les serveurs de catalogue et de conteneur peuvent être démarrés automatiquement dans un environnement WebSphere Application Server ou WebSphere Application Server Network Deployment.

Avant de commencer

Configurer les serveurs de catalogue et les serveurs de conteneur pour s'exécuter sur WebSphere Application Server:

- «Configuration du service de catalogue dans WebSphere Application Server», à la page 257
- «Configuration des serveurs de conteneurs dans WebSphere Application Server», à la page 275

Pourquoi et quand exécuter cette tâche

Le cycle de vie des serveurs de catalogue et de conteneur dans WebSphere Application Server est lié au processus dans lequel ces serveurs s'exécutent.

Procédure

- **Démarrez des services de catalogue dans WebSphere Application Server** :

  Le cycle de vie d'un serveur de catalogue est lié au processus WebSphere Application Server. Après avoir configuré le domaine de services de catalogue dans WebSphere Application Server, redémarrez chaque serveur que vous avez

défini pour ce domaine. Le service de catalogue démarre automatiquement sur les serveurs que vous avez associés au domaine de services de catalogue. Le service de catalogue peut également démarrer automatiquement dans les scénarios suivants, en fonction de l'édition WebSphere Application Server :

  - **Base WebSphere Application Server** : vous pouvez configurer votre application pour démarrer automatiquement un serveur de conteneur et le service de catalogue. Cette fonctionnalité simplifie le test des unités dans les environnements de développement comme Rational Application Developer car vous n'avez pas besoin de démarrer explicitement un service de catalogue. Pour plus d'informations, voir «Configuration des applications WebSphere Application Server pour démarrer automatiquement les serveurs de conteneur», à la page 275.
  - **WebSphere Application Server Network Deployment** : le service de catalogue démarre automatiquement dans le processus du gestionnaire de déploiement si le noeud du gestionnaire de déploiement a WebSphere eXtreme Scale est installé et que le profil du gestionnaire de déploiement est étendu. Pour plus d'informations, voir «Configuration du service de catalogue dans WebSphere Application Server», à la page 257..

- **Démarrez les serveurs de conteneur dans WebSphere Application Server** :

  Le cycle de vie d'un serveur de catalogue est lié à l'application WebSphere Application Server. Lorsque vous démarrez l'application configurée, les serveurs de conteneur démarrent également.

- **Arrêt de l'ensemble d'une grille de données de serveurs** :

  Vous pouvez arrêter les serveurs de catalogue et de conteneur en arrêtant les applications et les serveurs d'applications associés. Toutefois, vous pouvez également arrêter l'ensemble d'une grille de données avec l'utilitaire **xscmd** ou des beans gérés :

  - **Dans l'utilitaire xscmd :**

    Voir «Arrêt propre des serveurs avec l'utilitaire **xscmd**», à la page 409 pour plus d'informations sur l'arrêt de l'ensemble d'une grille de données.

  - **Avec des beans gérés :**

    utilisez l'opération tearDownServers sur le bean géré PlacementServiceMBean.

Utilisation de l'API de serveur embarqué pour démarrer et arrêter les serveurs

Avec WebSphere eXtreme Scale, vous pouvez utiliser une interface de programmation d'application pour gérer le cycle de vie des conteneurs et serveurs imbriqués. Vous pouvez configurer le serveur à l'aide d'un programme avec les options que vous pouvez également configurer avec la ligne de commande ou les propriétés de serveur incluses dans un fichier. Vous pouvez configurer le serveur imbriqué pour en faire un serveur conteneur et/ou un service de catalogue.

Avant de commencer

Vous devez disposer d'une méthode permettant d'exécuter du code depuis une machine virtuelle Java existante. les classes eXtreme Scale doivent être disponibles dans l'arborescence du chargeur de classe.

Pourquoi et quand exécuter cette tâche

Vous pouvez effectuer de nombreuses tâches d'administration à l'aide de l'API d'administration. L'API est couramment utilisée comme serveur interne pour

stocker l'état d'une application Web. Le serveur Web peut démarrer un serveur WebSphere eXtreme Scale imbriqué et signaler le serveur conteneur au service de catalogue. Le serveur est ensuite ajouté comme membre d'une grille répartie plus importante. Cette utilisation peut offrir des possibilité d'évolution et une haute disponibilité à un fichier de données qui reste sinon volatile.

Vous pouvez contrôler à l'aide d'un programme le cycle de vie complet d'un serveur eXtreme Scale imbriqué. Les exemples sont aussi génériques que possible et n'illustrent que des exemples de code spécifiques aux étapes présentées.

Procédure

1. Procurez-vous l'objet ServerProperties de la classe ServerFactory et configurez les options nécessaires.

   Chaque serveur eXtreme Scale possède un ensemble de propriétés configurables. Lorsqu'un serveur est démarré à partir de la ligne de commande, ces propriétés reçoivent les valeurs par défaut, mais vous pouvez remplacer plusieurs propriétés en fournissant un fichier ou une source externe. Dans la portée imbriquée, vous pouvez directement définir les propriétés avec un objet ServerProperties. Vous devez définir ces propriétés avant d'obtenir une instance de serveur de la classe ServerFactory. L'exemple de fragment de code ci-après obtient un objet ServerProperties, définit le champ CatalogServiceBootStrap et initialise plusieurs paramètres de serveur facultatifs. Pour une liste des paramètres configurables, reportez-vous à la documentation de l'API.

   ```
   ServerProperties props = ServerFactory.getServerProperties();
   props.setCatalogServiceBootstrap("host:port"); // requis pour se connecter
   à un service de catalogue spécifique
   props.setServerName("ServerOne"); // nommez le serveur
   props.setTraceSpecification("com.ibm.ws.objectgrid=all=enabled"); // Définit la spécification
   de trace
   ```

2. Si vous souhaitez que le serveur soit un service de catalogue, procurez-vous l'objet CatalogServerProperties.

   Chaque serveur imbriqué peut être un service de catalogue, un serveur conteneur ou un serveur conteneur et un service de catalogue. L'exemple ci-après obtient l'objet CatalogServerProperties, active l'option de service de catalogue et configure divers paramètres de service de catalogue.

   ```
   CatalogServerProperties catalogProps = ServerFactory.getCatalogProperties();
   catalogProps.setCatalogServer(true); // false par défaut ; doit être défini comme service de catalogue
   catalogProps.setQuorum(true); // active/désactive le quorum
   ```

3. Procurez-vous une instance Server à partir de la classe ServerFactory. L'instance Server est un singleton de portée processus chargé de gérer l'appartenance dans la grille. Une fois que cette instance a été instanciée, ce processus est connecté et devient hautement disponible pour les autres serveurs de la grille. L'exemple suivant montre comment créer l'instance Server :

   ```
   Server server = ServerFactory.getInstance();
   ```

   Si nous considérons l'exemple précédent, la classe ServerFactory fournit une méthode statique qui renvoie une instance Server. La classe ServerFactory est prévue pour être la seule interface permettant d'obtenir une instance Server. Par conséquent la classe garantit que l'instance est un singleton ou une instance pour chaque machine virtuelle Java ou chargeur de classe isolé. La méthode getInstance initialise l'instance Server. Vous devez configurer toutes les propriétés du serveur avant d'initialiser l'instance. La classe Server est chargée de créer des instances Container. Vous pouvez utiliser à la fois la classe ServerFactory et la classe Server pour gérer le cycle de vie de l'instance Server imbriquée.

4. Démarrez une instance Container à l'aide de l'instance Server.

Pour que des fragments puissent être positionnées sur un serveur imbriqué, vous devez créer un conteneur sur le serveur. L'interface Server contient une méthode createContainer et accepte un argument DeploymentPolicy. L'exemple ci-après utilise l'instance de serveur que vous avez obtenue pour créer un conteneur à l'aide du fichier de règles de déploiement. Notez que les conteneurs requièrent un chargeur de classe pour lequel les fichiers binaires de l'application sont disponibles, à des fins de sérialisation. Vous pouvez rendre ces fichiers binaires disponibles en appelant la méthode createContainer avec comme chargeur de classe du contexte de l'unité d'exécution, celui que vous souhaitez utiliser.

```
DeploymentPolicy policy = DeploymentPolicyFactory.createDeploymentPolicy(new
   URL("file://urltodeployment.xml"),
 new URL("file://urltoobjectgrid.xml"));
Container container = server.createContainer(policy);
```

5. Supprimez et nettoyez un conteneur.

   Vous pouvez supprimer et nettoyer un serveur conteneur en exécutant la méthode teardown sur l'instance Container obtenue. L'exécution de la méthode teardown sur un conteneur nettoie ce dernier de manière appropriée et supprime le conteneur du serveur imbriqué.

   La procédure de nettoyage du conteneur inclut le déplacement et le démontage de tous les fragments positionnés dans ce conteneur. Chaque serveur peut contenir plusieurs conteneurs et fragments. Le nettoyage d'un conteneur n'affecte pas le cycle de vie de l'instance Server parent. L'exemple ci-après montre comment exécuter la méthode teardown sur un serveur. La méthode teardown est rendue accessible via l'interface ContainerMBean. En utilisant l'interface ContainerMBean, si vous n'avez plus accès à ce conteneur à l'aide d'un programme, vous pouvez toujours le supprimer et le nettoyer avec son bean géré. Une méthode terminate existe également dans l'interface Container ; ne l'utilisez pas, sauf si cela est indispensable. Cette méthode est plus puissante et ne coordonne pas un déplacement et un nettoyage des fragments appropriés.

   ```
   container.teardown();
   ```

6. Arrêtez le serveur imbriqué.

   Lorsque vous arrêtez un serveur imbriqué, vous arrêtez également les conteneurs et les fragments en cours d'exécution sur ce serveur. Lorsque vous arrêtez un serveur imbriqué, vous devez nettoyer toutes les connexions ouvertes et déplacer ou démonter tous les fragments. L'exemple ci-après illustre comment arrêter un serveur et utiliser la méthode waitFor sur l'instance Server pour s'assurer que cette dernière s'arrête complètement. Comme pour l'exemple de conteneur, la méthode stopServer est rendue accessible via l'interface ServerMBean. A l'aide de cette interface, vous pouvez arrêter un serveur avec le bean géré (MBean) correspondant.

   ```
   ServerFactory.stopServer();  // Utilise la fabrique pour arrêter le singleton du serveur
    // ou
   server.stopServer(); // Utilise directement l'instance Server
   server.waitFor(); // Est renvoyé une fois que le serveur a correctement terminé
   ses procédures d'arrêt
   ```

   Exemple de code complet :

   ```
    import java.net.MalformedURLException;
   import java.net.URL;

   import com.ibm.websphere.objectgrid.ObjectGridException;
   import com.ibm.websphere.objectgrid.deployment.DeploymentPolicy;
   import com.ibm.websphere.objectgrid.deployment.DeploymentPolicyFactory;
   import com.ibm.websphere.objectgrid.server.Container;
   import com.ibm.websphere.objectgrid.server.Server;
   import com.ibm.websphere.objectgrid.server.ServerFactory;
   import com.ibm.websphere.objectgrid.server.ServerProperties;

   public class ServerFactoryTest {
   ```

```
    public static void main(String[] args) {

        try {

            ServerProperties props = ServerFactory.getServerProperties();
            props.setCatalogServiceBootstrap("catalogservice-hostname:catalogservice-port");
            props.setServerName("ServerOne"); // name server
            props.setTraceSpecification("com.ibm.ws.objectgrid=all=enabled"); // TraceSpec

            /*
             * Dans la plupart des cas, le serveur ne sert que de serveur conteneur
             * et se connecte à un service de catalogue externe. Cette utilisation
             * favorise davantage la haute disponibilité. L'extrait de code commenté
             * ci-après permet à ce serveur de devenir un service de catalogue.
             *
             *
             * CatalogServerProperties catalogProps =
             * ServerFactory.getCatalogProperties();
             * catalogProps.setCatalogServer(true); // activez le service de catalogue
             * catalogProps.setQuorum(true); // activez le quorum
             */

            Server server = ServerFactory.getInstance();

            DeploymentPolicy policy = DeploymentPolicyFactory.createDeploymentPolicy
     (new URL("url to deployment xml"),  new URL("url to objectgrid xml file"));
            Container container = server.createContainer(policy);

            /*
             * Le fragment est maintenant positionné sur ce conteneur si les exigences
             * de déploiement sont satisfaites.
             * Cela englobe la création du serveur et du conteneur imbriqués.
             *
             * Les lignes ci-après illustrent simplement l'appel des méthodes de nettoyage
             */

            container.teardown();
            server.stopServer();
            int success = server.waitFor();

        } catch (ObjectGridException e) {
            // Container failed to initialize
        } catch (MalformedURLException e2) {
            // invalid url to xml file(s)
        }

    }

}
```

API de serveurs intégrés

WebSphere eXtreme Scale comprend des interfaces de programmes d'application (API) et des interfaces de programmation de système permettant d'intégrer des serveurs et des clients eXtreme Scale dans vos applications Java existantes. Nous allons décrire ces API de serveurs intégrés.

Instanciation du serveur eXtreme Scale

Plusieurs propriétés permettent de configurer l'instance du serveur eXtreme Scale, qu'il est possible d'extraire de la méthode ServerFactory.getServerProperties. L'objet ServerProperties étant un singleton, chaque appel à la méthode getServerProperties extrait la même instance.

Le code suivant permet de créer un serveur :

```
Server server = ServerFactory.getInstance();
```

Toutes les propriétés définies avant la première invocation de getInstance sont utilisées pour initialiser le serveur.

Définir les propriétés du serveur

Vous pouvez définir les propriétés du serveur jusqu'au premier appel à la méthode ServerFactory.getInstance. Ce premier appel instancie le serveur eXtreme Scale et lit toutes les propriétés configurées. Les propriétés définies après la création n'ont aucun effet. L'exemple qui suit montre comment définir les propriétés avant d'instancier une instance Server.

```
// L'on obtient les propriétés du serveur associées à ce processus.
ServerProperties serverProperties = ServerFactory.getServerProperties();

// L'on définit le nom du serveur pour ce processus.
serverProperties.setServerName("EmbeddedServerA");

// L'on définit le nom de la zone dans laquelle est contenu ce processus.
serverProperties.setZoneName("EmbeddedZone1");

// L'on définit les informations de point de contact requises pour
// l'amorçage du service de catalogue.
serverProperties.setCatalogServiceBootstrap("localhost:2809");

// L'on définit le nom de l'hôte d'écoute de l'ORB à utiliser pour la liaison.
serverProperties.setListenerHost("host.local.domain");

// L'on définit le port d'écoute de l'ORB à utiliser pour la liaison.
serverProperties.setListenerPort(9010);

// L'on désactive tous les beans gérés pour ce processus.
serverProperties.setMBeansEnabled(false);

Server server = ServerFactory.getInstance();
```

Incorporer le service de catalogue

Tout paramètre JVM marqué par la méthode CatalogServerProperties.setCatalogServer peut héberger le service de catalogue d'eXtreme Scale. Cette méthode demande à l'environnement d'exécution du serveur eXtreme Scale d'instancier le service de catalogue lors du démarrage du serveur. Le code qui suit montre comment instancier le serveur de catalogue eXtreme Scale :

```
CatalogServerProperties catalogServerProperties =
 ServerFactory.getCatalogProperties();
catalogServerProperties.setCatalogServer(true);

Server server = ServerFactory.getInstance();
```

Incorporer le conteneur eXtreme Scale

La méthode Server.createContainer permet à n'importe quelle machine virtuelle Java d'héberger plusieurs conteneurs eXtreme Scale. Le code qui suit montre comment instancier un conteneur eXtreme Scale :

```
Server server = ServerFactory.getInstance();
DeploymentPolicy policy = DeploymentPolicyFactory.createDeploymentPolicy(
    new File("META-INF/embeddedDeploymentPolicy.xml").toURI().toURL(),
    new File("META-INF/embeddedObjectGrid.xml").toURI().toURL());
Container container = server.createContainer(policy);
```

Processus serveur autonome

Vous pouvez démarrer ensemble tous les services, ce qui est utilise aussi bien en phase de développement qu'en production. En démarrant les services ensemble, le même processus se charge de toutes les tâches suivantes : démarrage du service de

catalogue, démarrage d'un ensemble de conteneurs, exécution de la logique de connexion client. Démarrer les services de cette manière résout les problèmes de programmation antérieurs au déploiement dans un environnement réparti. Le code qui suit montre comment instancier un serveur eXtreme Scale autonome :

```
CatalogServerProperties catalogServerProperties =
 ServerFactory.getCatalogProperties();
catalogServerProperties.setCatalogServer(true);

Server server = ServerFactory.getInstance();
DeploymentPolicy policy = DeploymentPolicyFactory.createDeploymentPolicy(
    new File("META-INF/embeddedDeploymentPolicy.xml").toURI().toURL(),
    new File("META-INF/embeddedObjectGrid.xml").toURI().toURL());
Container container = server.createContainer(policy);
```

Intégrer eXtreme Scale dans WebSphere Application Server

La configuration eXtreme Scale est définie automatiquement lorsque vous installez eXtreme Scale dans un environnement WebSphere Application Server. Vous n'êtes pas obligé de définir des propriétés avant d'accéder au serveur pour créer un conteneur. Le code qui suit montre comment instancier un serveur eXtreme Scale dans WebSphere Application Server :

```
Server server = ServerFactory.getInstance();
DeploymentPolicy policy = DeploymentPolicyFactory.createDeploymentPolicy(
    new File("META-INF/embeddedDeploymentPolicy.xml").toURI().toURL(),
    new File("META-INF/embeddedObjectGrid.xml").toURI().toURL);
Container container = server.createContainer(policy);
```

«Utilisation de l'API de serveur embarqué pour démarrer et arrêter les serveurs», à la page 410 contient un exemple expliquant pas à pas comment démarrer par programmation un service de catalogue et un conteneur intégrés.

Administration avec l'utilitaire `xscmd`

Avec **xscmd**, vous pouvez effectuer des tâches d'administration dans l'environnement, telles qu'établir des liens de réplication multimaître, remplacer un quorum et arrêter des groupes de serveurs avec la commande teardown.

Avant de commencer

- Les serveurs de catalogue et les serveurs de conteneur doivent être démarrés. Si les serveurs de catalogue se trouvent dans un domaine de service de catalogue, au moins deux serveurs de catalogue doivent être démarrés.
- Vérifiez que la variable d'environnement *JAVA_HOME* est définie pour utiliser l'environnement d'exécution installé avec le produit. Si vous utilisez la version d'évaluation du produit, vous devez définir la variable d'environnement *JAVA_HOME*.

Pourquoi et quand exécuter cette tâche

L'utilitaire **xscmd** remplace l'exemple d'utilitaire **xsadmin** comme outils de surveillance et d'administration complètement pris en charge. Vous pouvez exécuter des opérations similaires avec l'outil **xsadmin**, mais cet outil n'est pas pris en charge. L'exemple **xsadmin** fournit une méthode pour effectuer l'analyse syntaxique et la détection des données de déploiement actuelles et peut servir de modèle pour l'écriture d'utilitaires personnalisés. Si vous utilisiez l'outil **xsadmin** pour la surveillance et l'administration, mettez à jour vos scripts pour utiliser l'utilitaire **inattendue**. Pour plus d'informations sur le mappage des commandes

xsadmin à la nouvelle commande **xscmd**, voir «Migration de l'outil **xsadmin** vers l'outil **xscmd**», à la page 216.

Procédure

1. Ouvrez une fenêtre de ligne de commande. Sur la ligne de commande, définissez les variables d'environnement appropriées.
   a. Définissez la variable d'environnement *CLIENT_AUTH_LIB* :
      - Windows `set CLIENT_AUTH_LIB=<path_to_security_JAR_or_classes>`
      - UNIX `set CLIENT_AUTH_LIB=<path_to_security_JAR_or_classes>`
        `export CLIENT_AUTH_LIB`
2. Accédez au répertoire *rép_base_wxs*`/bin`.
   `cd rép_base_wxs/bin`
3. Affichez l'aide des différentes options **xscmd**.
   - Pour afficher l'aide générale, exécutez la commande suivante :
     - UNIX `./xscmd.sh -h`
     - Windows `xscmd.bat -h`
   - Pour afficher la liste de toutes les commandes, exécutez la commande suivante :
     - UNIX `./xscmd.sh -lc`
     - Windows `xscmd.bat -lc`
   - Pour afficher l'aide d'une commande, exécutez la commande suivante :
     - UNIX `./xscmd.sh -h command_name`
     - Windows `xscmd.bat -h command_name`
   - Pour afficher une liste des groupes de commandes, exécutez la commande suivante :
     - UNIX `./xscmd.sh -lcg`
     - Windows `xscmd.bat -lcg`
   - Pour afficher la liste des commandes dans un groupe de commandes, exécutez la commande suivante :
     - UNIX `./xscmd.sh -lc command_group_name`
     - Windows `xscmd.bat -lc command_group_name`
4. Exécutez les commandes de connexion à de serveurs de catalogue spécifiques. Par défaut, **xscmd** se connecte au serveur de catalogue sur l'hôte local en utilisant le nom d'hôte et le port `localhost:2809`. Vous pouvez également fournir la liste des noms d'hôte et des ports à la commande pour que vous puissiez vous connecter aux serveurs de catalogue sur d'autres hôtes. Dans la liste, l'un des utilitaires **xscmd** se connecte à un hôte aléatoire. La liste des hôtes que vous fournissez doit se trouver dans le même domaine de services de catalogue.
   - Fournissez la liste des serveurs de catalogue autonomes auxquels vous voulez vous connecter :
     - UNIX `./xscmd.sh -c <command_name> -cep hostname:port(,hostname:port)`
     - Windows `xscmd.bat -c <command_name> -cep hostname:port(,hostname:port)`

Dans les commandes précédentes, *command_name* est le nom de la commande que vous exécutez. La valeur *hostname:port* est le nom d'hôte du serveur de catalogue et le port d'écoute. La valeur de port d'écoute sur un serveur de catalogue autonome est définie lorsque vous exécutez la commande **startOgServer**.

- Fournissez la liste des serveurs de catalogue WebSphere Application Server auxquels vous voulez vous connecter. Vous ne pouvez pas vous connecter aux serveurs de catalogue qui s'exécutent sur WebSphere Application Server avec la valeur localhost par défaut :
  - UNIX `./xscmd.sh -c <command_name> -cep was_hostname:port(,hostname:port)`
  - Windows `xscmd.bat -c <command_name> -cep was_hostname:port(,hostname:port)`

  Dans les commandes précédentes, *command_name* est le nom de la commande que vous exécutez. La valeur *was_hostname* est le nom d'hôte du serveur de catalogue dans la cellule WebSphere Application Server. La valeur *port* est le port d'écoute. La valeur de port d'écoute dans WebSphere Application Server est héritée par la configuration de port BOOTSTRAP_ADDRESS. La valeur par défaut est 9809 si le serveur de catalogue s'exécute dans le gestionnaire de déploiement. Si vous exécutez le serveur de catalogue sur un serveur d'applications, vérifiez la configuration de port BOOTSTRAP_ADDRESS du serveur d'applications pour déterminer le numéro de port.

Démarrage des serveurs eXtreme Scale en utilisant l'infrastructure OSGi Eclipse Equinox

Les serveurs de conteneur WebSphere eXtreme Scale peuvent être démarrés dans une infrastructure OSGi Eclipse Equinox en utilisant plusieurs méthodes.

Avant de commencer

Pour pouvoir démarrer un conteneur eXtreme Scale, vous devez exécuter les tâches suivantes :

1. L'ensemble de serveur WebSphere eXtreme Scale doit être installé dans Eclipse Equinox.
2. L'application doit être placée dans un ensemble OSGi.
3. Les plug-ins WebSphere eXtreme Scale (s'il en existe) doivent être placés dans un ensemble OSGi. Ils peuvent se trouver dans le même ensemble que l'application ou dans des ensembles séparés.

Pourquoi et quand exécuter cette tâche

Cette tâche explique comment démarrer un serveur de conteneur eXtreme Scale dans une infrastructure OSGi Eclipse Equinox. Vous pouvez utiliser n'importe laquelle des méthodes suivantes pour démarrer les serveurs de conteneur en utilisant l'implémentation Eclipse Equinox :

- Service OSGi Blueprint

  Vous pouvez inclure toute la configuration et toutes les métadonnées dans un ensemble OSGi. Voir l'illustration suivante pour comprendre le processus Eclipse Equinox de cette méthode :

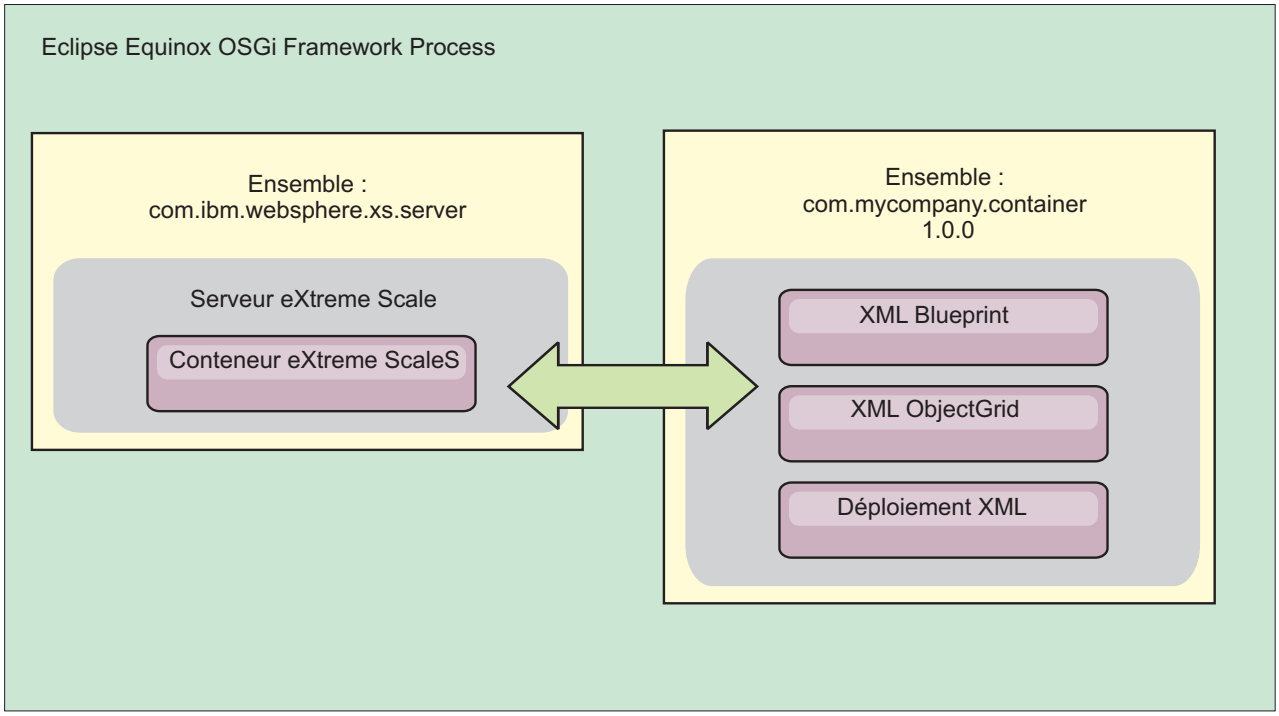


Figure 49. Processus Eclipse Equinox pour inclure toute la configuration et toutes les métadonnées dans un ensemble OSGi

- Service Admin de configuration OSGi

  Vous pouvez définir la configuration et les métadonnées en dehors d'un ensemble OSGi. Voir l'image suivante pour comprendre le processus Eclipse Equinox pour cette méthode :

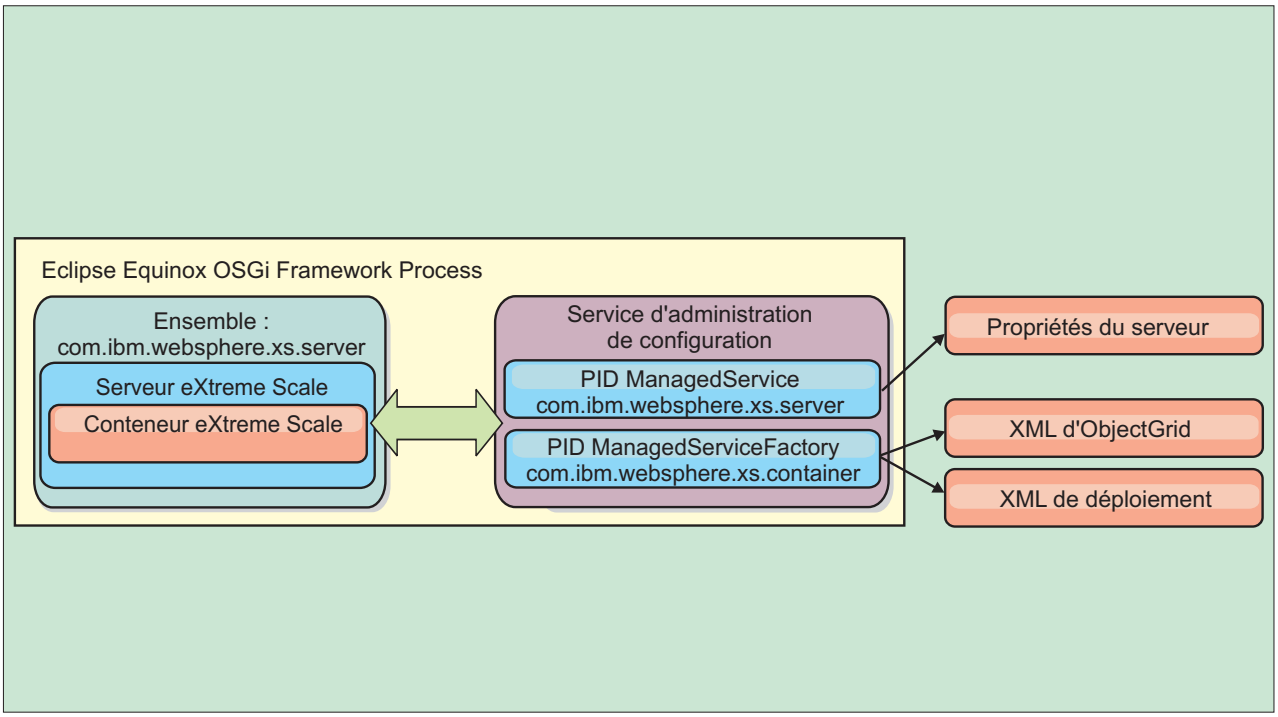


Figure 50. Processus Eclipse Equinox pour définir la configuration et les métadonnées en dehors d'un ensemble OSGi

- A l'aide d'un programme

  Prend en charge les solutions de configuration personnalisées.

Dans chaque cas, un singleton de serveur eXtreme Scale est configuré et un ou plusieurs conteneurs sont configurés.

L'ensemble de serveur eXtreme Scale, `objectgrid.jar`, contient toutes les bibliothèques nécessaires pour démarrer et exécuter un conteneur de grille eXtreme Scale dans une infrastructure OSGi. L'environnement d'exécution du serveur communique avec les plug-ins fournis par l'utilisateur et les objets de données en utilisant le gestionnaire de service OSGi.

Important : Après que l'ensemble de serveur eXtreme Scale a été démarré et le serveur eXtreme Scale initialisé, il ne peut pas être redémarré. Le processus Eclipse Equinox doit être redémarré pour redémarrer le serveur eXtreme Scale.

Vous pouvez utiliser le support eXtreme Scale pour l'espace de nom Spring pour configurer les serveurs de conteneur eXtreme Scale dans un fichier XML Blueprint. Lorsque les éléments XML de serveur et de conteneur sont ajoutés au fichier XML Blueprint, le gestionnaire d'espace de nom eXtreme Scale démarre automatiquement un serveur de conteneur en utilisant les paramètres définis dans le fichier XML Blueprint lors du démarrage de l'ensemble. Le gestionnaire arrête le conteneur lorsque l'ensemble s'arrête.

Pour configurer les serveurs de conteneur eXtreme Scale avec XML Blueprint, procédez comme suit :

Procédure

- Démarrez un serveur de conteneur eXtreme Scale en utilisant OSGi Blueprint.
  1. Créez un ensemble de conteneur.

2. Installez l'ensemble de conteneur dans l'infrastructure OSGi Eclipse Equinox. Voir «Installation et démarrage des plug-ins OSGi».
3. Démarrez l'ensemble de conteneur.

- Démarrez un serveur de conteneur eXtreme Scale en utilisant l'administrateur de configuration OSGi.
  1. Configurez le serveur et le conteneur en utilisant l'administrateur de configuration.
  2. Lorsque l'ensemble de serveur eXtreme Scale est démarré ou que les PID (persistent identifier) sont créé avec l'administrateur de configuration, le serveur et le conteneur démarrent automatiquement.
- Démarrez un serveur de conteneur eXtreme Scale en utilisant l'API ServerFactory. Voir la documentation d'API de serveur.
  1. Créez une classe d'activateur d'ensemble OSGi et utilisez l'API eXtreme Scale ServerFactory pour démarrer un serveur.

Installation et démarrage des plug-ins OSGi

Dans cette tâches, vous installez l'ensemble de plug-in dynamique dans l'infrastructure OSGi, puis vous démarrez le plug-in.

Avant de commencer

Cette rubrique suppose que vous avez exécuté les tâches suivantes :

- Vous avez installé l'ensemble serveur ou client eXtreme Scale dans l'infrastructure OSGi Eclipse Equinox. Voir «Installation des ensembles eXtreme Scale», à la page 205.
- Vous avez implémenté un ou plusieurs plug-ins dynamiques BackingMap ou ObjectGrid. Voir Génération de plug-ins dynamiques eXtreme Scale.
- Vous avez regroupé les plug-ins dynamiques comme services OSGi dans des ensembles OSGi.

Pourquoi et quand exécuter cette tâche

Cette tâche explique comment installer l'ensemble en utilisant la console Eclipse Equinox. L'ensemble peut être installé en utilisant plusieurs méthodes différentes, y compris en modifiant le fichier de configuration `config.ini`. Les produits qui intègrent Eclipse Equinox incluent des méthodes alternatives de gestion des ensembles. Pour plus d'informations sur l'ajout d'ensembles dans le fichier `config.ini` dans Eclipse Equinox, voir les options d'exécution Eclipse.

OSGi permet de démarrer les ensembles ayant des services dupliqués. WebSphere eXtreme Scale utilise le dernier classement de service. Lors du démarrage de plusieurs infrastructures OSGi dans une grille de données eXtreme Scale, vous devez veiller à démarrer les classements de service corrects sur chaque serveur afin que la grille ne soit pas démarrée en utilisant une combinaison de versions différentes.

Pour identifier les versions utilisées par la grille de données, utilisez l'utilitaire xscmd pour vérifier les classements en cours et disponibles. Pour plus d'informations sur les classements de service disponibles, voir «Mise à jour des services OSGi pour les plug-ins eXtreme Scale avec **xscmd**», à la page 425.

Procédure

Installez l'ensemble de plug-in dans l'infrastructure OSGi Eclipse Equinox en utilisant la console OSGi.

1. Démarrez l'infrastructure Eclipse Equinox avec la console activée, par exemple :
   ```
   <java_home>/bin/java -jar <equinox_root>/plugins/org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
   ```
2. Installez l'ensemble de plug-in dans la console Equinox.
   ```
   osgi> install file:///<path to bundle>
   ```
   Equinox affiche l'ID du nouvel ensemble installé :
   ```
   Bundle id is 17
   ```
3. Entrez la ligne suivante pour démarrer l'ensemble dans la console Equinox, où `<id>` est l'ID d'ensemble affecté lors de l'installation de l'ensemble :
   ```
   osgi>  install <id>
   ```
4. Extrayez l'état du service dans la console Equinox pour vérifier que l'ensemble a démarré :
   ```
   osgi> ss
   ```
   Lorsque l'ensemble a démarré correctement, il affiche l'état ACTIVE, par exemple :
   ```
   17      ACTIVE      com.mycompany.plugin.bundle_VRM
   ```

Installez l'ensemble de plug-in dans l'infrastructure OSGi Eclipse Equonix en utilisant le fichier config.ini file.

5. Copiez l'ensemble de plug-in dans le répertoire Eclipse Equinox plug-ins, par exemple :
   ```
   <equinox_root>/plugins
   ```
6. Modifiez le fichier de configuration Eclipse Equinox `config.ini` et ajoutez l'ensemble à la propriété osgi.bundles, par exemple :
   ```
   osgi.bundles=\
   org.eclipse.osgi.services_3.2.100.v20100503.jar@1:start, \
   org.eclipse.osgi.util_3.2.100.v20100503.jar@1:start, \
   org.eclipse.equinox.cm_1.0.200.v20100520.jar@1:start, \
   com.mycompany.plugin.bundle_VRM.jar@1:start
   ```
   Important : Vérifiez qu'il existe une ligne blanche après le dernier nom d'ensemble. Chaque ensemble est séparé par une virgule.
7. Démarrez l'infrastructure Eclipse Equinox avec la console activée, par exemple :
   ```
   <java_home>/bin/java -jar <equinox_root>/plugins/org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
   ```
8. Extrayez l'état de service dans la console Equinox pour vérifier que l'ensemble est démarré. Par exemple :
   ```
   osgi> ss
   ```
   Une fois l'ensemble démarré, il affiche l'état ACTIVE. Par exemple :
   ```
   17      ACTIVE      com.mycompany.plugin.bundle_VRM
   ```

Résultats

L'ensemble de plug-in est maintenant installé et démarré. Le conteneur ou le client peut être maintenant démarré eXtreme Scale. Pour plus d'informations sur le développement des plug-ins eXtreme Scale, voir la rubrique API système et plug-ins.

Administration des services OSGi en utilisant l'utilitaire xscmd

Vous pouvez utiliser l'utilitaire **xscmd** pour exécuter des tâches d'administration, telles qu'afficher les serveurs et leurs classements utilisés par chaque conteneur, et mettre à niveau l'environnement d'exécution pour utiliser les nouvelles versions des ensembles.

Pourquoi et quand exécuter cette tâche

Avec l'infrastructure Eclipse Equinox OSGi, vous pouvez installer plusieurs versions d'un même ensemble et vous pouvez mettre à jour ces ensemble lors de l'exécution. WebSphere eXtreme Scale est un environnement distribué qui exécute les serveurs de conteneur dans une multitude d'instances de l'infrastructure OSGi.

Les administrateurs doivent copier, installer et démarrer manuellement les ensembles dans l'infrastructure OSGi. eXtreme Scale contient un personnaliseur ServiceTrackerCustomizer OSGi pour suivre les services identifiés comme plug-ins eXtreme Scale dans le fichier XML descripteur. Utilisez l'utilitaire **xscmd** pour valider la version utilisée du plug-in, les versions pouvant être utilisées et exécuter des mises à niveau d'ensemble.

eXtreme Scale utilise le numéro de classement de service pour identifier la version de chaque service. Lorsque au moins deux services sont chargés avec la même référence, eXtreme Scale utilise automatiquement le service ayant le classement le plus élevé.

Procédure

- Exécutez la commande **osgiCurrent** et vérifiez que chaque serveur eXtreme Scale utilise le classement de service de plug-in correct.

  Comme eXtreme Scale choisit automatiquement la référence de service ayant le classement le plus élevé, il se peut que la grille de données démarre avec plusieurs classements d'un service de plug-in.

  Si la commande détecte une discordance de classements ou qu'elle ne trouve pas un service, un niveau d'erreur différent de zéro est défini. Si la commande aboutit, le niveau d'erreur 0 est défini.

  L'exemple suivant montre la sortie de la commande **osgiCurrent** lorsque deux plus-ins sont installés dans une grille sur quatre serveurs. Le plug-in loaderPlugin utilise le classement 1 et le plug-in txCallbackPlugin, le classement 2.

```
OSGi Service Name Current Ranking ObjectGrid Name MapSet Name Server Name
----------------- --------------- --------------- ----------- -----------
loaderPlugin      1               MyGrid          MapSetA     server1
loaderPlugin      1               MyGrid          MapSetA     server2
loaderPlugin      1               MyGrid          MapSetA     server3
loaderPlugin      1               MyGrid          MapSetA     server4
txCallbackPlugin  2               MyGrid          MapSetA     server1
txCallbackPlugin  2               MyGrid          MapSetA     server2
txCallbackPlugin  2               MyGrid          MapSetA     server3
txCallbackPlugin  2               MyGrid          MapSetA     server4
```

  L'exemple suivant montre la sortie de la commande **osgiCurrent** lorsque le serveur 2 a été démarré avec un nouveau classement du plug-in loaderPlugin :

```
OSGi Service Name Current Ranking ObjectGrid Name MapSet Name Server Name
----------------- --------------- --------------- ----------- -----------
loaderPlugin      1               MyGrid          MapSetA     server1
loaderPlugin      2               MyGrid          MapSetA     server2
loaderPlugin      1               MyGrid          MapSetA     server3
loaderPlugin      1               MyGrid          MapSetA     server4
```

```
txCallbackPlugin  2              MyGrid          MapSetA     server1
txCallbackPlugin  2              MyGrid          MapSetA     server2
txCallbackPlugin  2              MyGrid          MapSetA     server3
txCallbackPlugin  2              MyGrid          MapSetA     server4
```

- Exécutez la commande **osgiAll** pour vérifier que les services de plug-in ont été correctement démarrés sur chaque serveur de conteneur eXtreme Scale.

  Lorsque des ensembles contenant des services référencés par une configuration ObjectGrid démarrent, l'environnement d'exécution eXtreme Scale suit le plug-in, mais il ne l'utilise pas immédiatement. La commande **osgiAll** montre les plug-ins disponibles pour chaque serveur.

  Lorsqu'elle est exécutée sans paramètres, tous les services de toutes les grilles et de tous les serveurs sont indiqués. Des filtres supplémentaires, notamment le filtre **-serviceName** <service_name>, peuvent être définis pour limiter la sortie à un seul service ou sous-ensemble de la grille de données.

  L'exemple suivant montre la sortie de la commande **osgiAll** lorsque deux plug-ins sont démarrés sur deux serveurs. Les classements 1 et 2 du plug-in loaderPlugin sont démarrés et le classement 1 du plug-in txCallbackPlugin est démarré. Le résumé à la fin de la sortie indique que les deux serveurs voient les mêmes classements de service :

```
Server: server1
   OSGi Service Name   Available Rankings
   -----------------   ------------------
   loaderPlugin        1, 2
   txCallbackPlugin    1

Server: server2
   OSGi Service Name   Available Rankings
   -----------------   ------------------
   loaderPlugin        1, 2
   txCallbackPlugin    1

Summary - All servers have the same service rankings.
```

  L'exemple suivant montre la sortie de la commande **osgiAll** lorsque l'ensemble qui contient le plug-in loaderPlugin avec le classement 1 est arrêté sur le serveur 1. Le résumé à la fin de la sortie indique que le serveur n'a pas le plug-in loaderPlugin avec le classement 1 :

```
Server: server1
   OSGi Service Name   Available Rankings
   -----------------   ------------------
   loaderPlugin        2
   txCallbackPlugin    1

Server: server2
   OSGi Service Name   Available Rankings
   -----------------   ------------------
   loaderPlugin        1, 2
   txCallbackPlugin    1

Summary - The following servers are missing service rankings:
   Server  OSGi Service Name Missing Rankings
   ------  ----------------- ----------------
   server1 loaderPlugin      1
```

  L'exemple suivant montre la sortie si le nom de service est défini avec l'argument **-sn** et que le service n'existe pas.

```
Server: server2
   OSGi Service Name Available Rankings
   ---------------- ------------------
   invalidPlugin     No service found

Server: server1
```

```
OSGi Service Name Available Rankings
---------------- ------------------
invalidPlugin     No service found

Summary - All servers have the same service rankings.
```

- Exécutez la commande **osgiCheck** pour vérifier les groupes de services de plug-in et de classements s'ils sont disponibles.

  La commande **osgiCheck** accepte un ou plusieurs groupes de classements de service de la manière suivante `-serviceRankings <service name>;<ranking>[,<serviceName>;<ranking>]`

  Lorsque les classements sont tous disponibles, la méthode retourne un niveau d'erreur 0. Si un ou plusieurs classements sont indisponibles, un niveau d'erreur différent de zéro et la table de tous les serveurs qui ne contiennent pas les classements de service définis sont spécifiés. Des filtres supplémentaires peuvent être utilisés pour limiter la vérification des services à un sous-ensemble des serveurs disponibles dans le domaine eXtreme Scale.

  Par exemple, si le classement ou le service est absent, le message suivant s'affiche :

```
Server  OSGi Service Unavailable Rankings
------  ------------ --------------------
server1 loaderPlugin 3
server2 loaderPlugin 3
```

- Exécutez la commande **osgiUpdate** pour mettre à jour le classement d'un ou de plusieurs plug-ins pour tous les serveurs dans un seul ObjectGrid et MapSet dans une seule opération.

  La commande accepte un ou plusieurs groupes de classements de service de la manière suivante : `-serviceRankings <service name>;<ranking>[,<serviceName>;<ranking>] -g <grid name> -ms <mapset name>`

  Avec cette commande, vous pouvez exécuter les opérations suivantes :

  – Vérifier que les services spécifiés sont disponibles pour la mise à niveau sur chacun des serveurs.
  – Mettre la grille hors ligne en utilisant l'interface StateManager. Pour plus d'informations, voir «Gestion de la disponibilité ObjectGrid», à la page 429. Ce processus met au repos la grille et attend la fin des transactions en cours en interdisant le démarrage de nouvelles transactions. Ce processus indique également aux programmes d'écoute ObjectGridLifecycleListener et BackingMapLifecycleListener d'arrêter toute activité transactionnelle. Voir Plug-in de programme d'écoute d'événement pour plus d'informations sur les plug-ins de programme d'écoute.
  – Mettre à jour chaque conteneur eXtreme Scale exécuté dans une infrastructure OSGi pour utiliser les nouvelles versions de service.
  – Mettre la grille en ligne pour reprendre l'exécution des transactions.

  Le processus de mise à jour est idempotent de sortie que si un client n'exécute pas une tâche, l'opération est annulée. Si un client ne peut pas exécuter l'annulation ou qu'il est interrompu pendant la mise à jour, la même commande peut être réexécutée et elle reprend à l'étape appropriée.

  Si le client ne peut pas continuer et que le processus est redémarré depuis un autre client, utilisez l'option `-force` pour permettre au client d'exécuter la mise à jour. La commande **osgiUpdate** empêche plusieurs client de mettre à jour simultanément un même groupe de mappes. Pour plus d'informations sur la commande **osgiUpdate**, voir «Mise à jour des services OSGi pour les plug-ins eXtreme Scale avec **xscmd**», à la page 425.

Mise à jour des services OSGi pour les plug-ins eXtreme Scale avec `xscmd`

WebSphere eXtreme Scale prend en charge la mise à niveau des ensembles de plug-in de serveur de conteneur lorsque la grille est active. Ainsi, les administrateurs peuvent mettre à jour les applications et effectuer des ajouts sans avoir à démarrer les processus de la grille.

Avant de commencer

Procédez comme suit avant de mettre à jour les ensembles eXtreme Scale OSGi vers une nouvelle version :

1. Démarrez les serveurs eXtreme Scale dans une infrastructure OSGi compatible.
2. Divisez tous les plug-ins eXtreme Scale en ensembles ; ils doivent utiliser les classements de service pour identifier chaque version des plug-ins.
3. Définissez les objets cache comme types primitifs Java, tels que `byte[]`, Integer ou String ou bien ils doivent être stockés en utilisant un plug-in MapSerializerPlugin. Les objets données sont stockés dans l'ensemble eXtreme Scale et ne sont pas mis à niveau. Seuls les plug-ins qui interagissent avec les données sont mis à jour.
4. Créez des données d'objet cache compatibles avec la version. Les nouveaux plug-ins doivent pouvoir interagir avec les données créées par les anciens plug-ins.
5. Créez des plug-ins pour écouter les événements ObjectGridLifecycle et BackingMapLifecycle et pour régénérer les références aux autres plug-ins ou métadonnées dans ces plug-ins. Ainsi, les plug-ins référencés sont régénérés lorsque le plug-in principal est mis à jour.
6. Le processus de mise à jour OSGi eXtreme Scale affecte uniquement les serveurs. Vous devez mettre à jour de manière indépendante les clients qui utilisent les plug-ins.

Pourquoi et quand exécuter cette tâche

Sans l'activation d'OSGi, si un administrateur doit mettre à jour les plug-ins d'application ou les objets cache, chaque noeud de grille doit être mis à jour un par un, ce qui affecte le réseau, la mémoire et l'utilisation du processeur. Cette opération est nécessaire, car les plug-ins et les objets Java cache sont directement stockés dans la grille. Lorsque les classes sont mises à jour sans redémarrer les processus, les plug-ins de grille génèrent des conflits, car chaque classe un chargeur ClassLoader différent.

Le produit eXtreme Scale contient l'utilitaire xscmd et des beans gérés qui permettent aux administrateurs d'afficher tous les ensembles de plug-in installés dans l'infrastructure OSGi d'hébergement de chaque conteneur de la grille et de choisir la révision à utiliser. Lorsque vous utilisez l'utilitaire xscmd pour mettre à jour les plug-ins vers un nouveau classement, la grille est mise au repos et toutes les transactions sont arrêtées, les plug-ins sont mis à jour et la grille est réactivée. En cas d'erreur lors de la mise à jour, le processus est annulé et l'ancien classement est restauré.

Procédure

1. Créez une version de l'ensemble en augmentant le numéro de version dans le manifeste de l'ensemble et le classement de chaque service de plug-in eXtreme

Scale. Si la version de l'ensemble d'origine est `Bundle-Version: 1.0.0`, la version suivante peut être `Bundle-Version: 1.1.0`.

Si le classement de service d'origine est `ranking="1"`, le classement suivant peut être `ranking="2"`.

Important : Les classements de service OSGi doivent être des entiers.

2. Copiez le nouvel ensemble vers chaque noeud de l'infrastructure OSGi qui héberge un serveur de conteneur eXtreme Scale.
3. Installez le nouvel ensemble dans l'infrastructure OSGi. L'ensemble est affecté d'un identificateur, par exemple :

   ```
   osgi> install <URL to bundle>
   ```

4. Démarrez le nouvel ensemble en utilisant l'identificateur affecté, par exemple :

   ```
   osgi> start <id>
   ```

   Une fois le nouvel ensemble démarré, le programme de suivi de service OSGi eXtreme Scale détecte l'ensemble et le rend disponible pour la mise à jour.
5. Utilisez la commande **xscmd -c osgiAll** pour vérifier que chaque serveur de conteneur voit le nouvel ensemble. La commande **osgiAll** interroge tous les conteneurs dans la grille pour tous les services qui sont référencés dans le fichier XML descripteur ObjectGrid et affiche tous les classements disponibles, par exemple :

   ```
   xscmd -c osgiAll

   Server: server1
      OSGi Service Name          Available Rankings
      -----------------          ------------------
      myLoaderServiceFactory     1, 2
      mySerializerServiceFactory 1, 2

   Server: server2
      OSGi Service Name          Available Rankings
      -----------------          ------------------
      myLoaderServiceFactory     1, 2
      mySerializerServiceFactory 1, 2

   Summary - All servers have the same service rankings.
   ```

6. Utilisez la commande **xscmd -c osgiCheck** pour vérifier qu'un ou plusieurs classements de service sont des cibles de mise à jour valides. Par exemple :

   ```
   xscmd -c osgiCheck -sr
   mySerializerServiceFactory;2,myLoaderServiceFactory;2

   CWXSI0040I: The command osgiCheck has completed successfully.
   ```

7. Si la commande **osgiCheck** ne trouve pas des erreurs résultantes, suspendez l'équilibreur du service de placement pour éviter les mouvements de fragments en cas d'erreur lors de la mise à jour. Pour suspendre le placement, utilisez la commande **xscmd -c suspendBalancing** pour chaque grille d'objets et chaque groupe de mappes affectés par la mise à jour, par exemple :

   ```
   xscmd -c suspendBalancing -g MyGrid -ms MyMapSet
   ```

8. Lorsque l'équilibrage a été suspendu pour chaque grille d'objets et groupe de mappes, utilisez la commande **xscmd -c osgiCheck** de nouveau pour vérifier qu'un ou plusieurs classements de service sont des cibles de mise à jour valides. Par exemple :

   ```
   xscmd -c osgiCheck -sr
   mySerializerServiceFactory;2,myLoaderServiceFactory;2

   CWXSI0040I: The command osgiCheck has completed successfully.
   ```

9. Lorsque l'équilibrage a été suspendu pour chaque grille d'objets et groupe de mappes, utilisez la commande **osgiUpdate** pour mettre à jour le service sur tous les serveurs pour une grille d'objets et un groupe de mappes. Par exemple :

   ```
   xscmd -c osgiUpdate  -sr
   mySerializerServiceFactory;2,myLoaderServiceFactory;2 -g MyGrid -ms MyMapSet
   ```

10. Vérifiez que la mise à niveau a abouti. Par exemple :

    ```
    La mise à jour abouti pour les classements de services suivants :
    Service                     Ranking
    -------                     -------
    mySerializerServiceFactory  2
    myLoaderServiceFactory      2
    ```

11. Après avoir vérifié que le classement a été mis à jour, activez de nouveau l'équilibrage en utilisant la commande **xscmd -c resumeBalancing**. Par exemple, :

    ```
    xscmd -c resumeBalancing -g MyGrid -ms MyMapSet
    ```

12. Arrêtez et désinstallez l'ancien ensemble dans chaque infrastructure OSGi qui héberge le conteneur eXtreme Scale. Par exemple, entrez le code suivant dans la console Eclipse Equinox :

    ```
    osgi> stop <id>
    osgi> uninstall <id>
    ```

Résultats

L'ensemble eXtreme Scale a été mis à jour vers une nouvelle version.

Contrôle du placement

Vous pouvez utiliser différentes options pour contrôler quand les fragments sont placés sur les différents serveurs de la configuration. Lors du démarrage, vous pouvez décider de retarder le placement des fragments. Lorsque vous exécutez tous les serveurs de conteneur, il peut être nécessaire de suspendre ou de changer le placement pendant dans la gestion des serveurs.

Procédure

Contrôle du placement lors du démarrage
Vous pouvez contrôler quand les fragments commencent à être placés lors du démarrage de l'environnement. Il existe un contrôle par défaut. Si vous n'exécutez aucune action pour contrôler le placement des fragments, ce dernier commence immédiatement. Dans ce cas, les fragments peuvent ne pas être placés uniformément lorsque les serveurs de conteneur suivants démarrent et que d'autres opérations de placement sont exécutées pour équilibrer la répartition.

- Suspendez temporairement l'équilibrage des fragments pour que les fragments ne soient pas placés immédiatement lorsque les serveurs de conteneur démarrent.

  Avant de démarrer les serveurs de conteneur, utilisez la commande **xscmd -c suspendBalancing** pour arrêter l'équilibrage des fragments pour une grille de données et un groupe de mappes donnés. Après que les serveurs de conteneur ont démarré, vous pouvez utiliser la commande **xscmd -c resumeBalancing** pour commencer à placer les fragments sur le serveurs de conteneur.

- 7.1.1+ Définissez la propriété **placementDeferralInterval**.

La propriété **placementDeferralInterval** réduit le nombre de cycles de placement de fragment sur les serveurs de conteneur. Le placement des fragments se déclenche à la fréquence définie.

Définissez la propriété **placementDeferralInterval** dans le fichier des propriétés du serveur de catalogue. Si vous utilisez l'API de serveur intégré, utilisez la méthode setPlacementDeferralInterval sur l'interface CatalogServerProperties. Cette propriété définit le délai en millisecondes qui précède le placement des fragments sur les serveurs de conteneur. La valeur par défaut de la propriété est 15 secondes. Avec cette valeur, lorsqu'un serveur de conteneur démarre, le placement ne démarre pas tant que le délai défini dans la propriété n'est pas écoulé. Si plusieurs serveurs de conteneur démarrent consécutivement, le chronomètre de report est réinitialisé si un nouveau serveur de conteneur démarre dans le délai défini. Si, par exemple, un deuxième conteneur démarre 10 secondes après le premier, le placement a lieu 15 secondes après le démarrage du deuxième serveur de conteneur. Toutefois, si un troisième serveur de conteneur démarre 20 secondes après le deuxième, le placement a déjà commencé sur les deux premiers serveurs de conteneur.

Lorsqu'un serveur de conteneur devient indisponible, le placement a lieu dès que le serveur de catalogue a connaissance de l'événement pour que la récupération ait lieu aussi rapidement que possible.

Suivez les conseils ci-dessous pour déterminer si la valeur de report du placement correspond au délai correct :

- Lorsque vous démarrez les serveurs simultanément, consultez les messages CWOBJ1001 dans le fichier `SystemOut.log` de chaque serveur de conteneur. L'horodatage de ces messages dans chaque fichier journal de serveur de conteneur indique l'heure de début du serveur de conteneur. Vous pouvez envisager d'ajuster la propriété **placementDeferralInterval** pour inclure plus de démarrages de serveur de conteneur. Par exemple, si le premier serveur de conteneur démarre 90 secondes avant le dernier serveur de conteneur, vous pouvez affecter la valeur 90 secondes à la propriété.
- Notez le délai d'apparition des messages CWOBJ1511 après les messages CWOBJ1001. Ce délai peut indiquer si le report a abouti.
- Si vous utilisez un environnement de développement, tenez compte du délai lorsque vous testez l'application.

- Définissez l'attribut **numInitialContainers**.

  Si vous avez déjà utilisé l'attribut **numInitialContainers**, vous pouvez continuer de l'utiliser. Toutefois, il est préférable d'utiliser les commandes **xscmd -c suspendBalancing** et **xscmd -c resumeBalancing** suivies de **placementDeferralInterval** que l'attribut **numInitialContainers** pour contrôler le placement. L'attribut **numInitialContainers** indique le nombre de serveurs de conteneur nécessaires avant le placement initial des fragments dans cet élément mapSet. L'attribut **numInitialContainers** se trouve dans le fichier XML descripteur de stratégie de déploiement. Si **numInitialContainers** et **placementDeferralInterval** sont définis, aucun placement n'a lieu jusqu'à ce que la valeur **numInitialContainers** soit atteinte, quelle que soit la valeur de la propriété **placementDeferralInterval**.

Contrôle du placement après le démarrage initial

- Forcez le placement.

  Utilisez la commande **xscmd -c triggerPlacement -g my_OG -ms my_Map_Set**, où *my_OG* et *my_Map_Set* sont affectés de valeur pour la grille de données et le groupe de mappes, pour forcer le placement à un moment où il n'aurait pas lieu.

Par exemple, vous pouvez exécuter cette commande lorsque le délai défini par la propriété **placementDeferralInterval** n'a pas encore été transmise ou lorsque l'équilibrage est suspendu.

- Réaffectez un fragment primaire.

  Utilisez la commande **xscmd -c swapShardWithPrimary** pour affecter un fragment de réplique comme fragment primaire. Le fragment primaire antérieur devient une réplique.

- Rééquilibrez les fragments primaire et de réplique.

  Utilisez la commande **xscmd -c balanceShardTypes** pour ajuster le taux des fragments primaire et de réplique uniformément entre les serveurs de conteneur de la configuration. Le taux est cohérent dans un fragment sur chaque serveur de conteneur.

- Suspendez ou relancez le placement.

  Utilisez la commande **xscmd -c suspendBalancing** ou **xscmd -c resumeBalancing** pour arrêter et démarrer l'équilibrage des fragments d'une grille de données ou d'un groupe de mappes. Lorsque l'équilibrage est suspendu, les actions de placement suivantes peuvent toujours s'exécuter :

  – La promotion de fragment peut avoir lieu lorsque les serveurs de conteneur sont défaillants.

  – La permutation de rôle de fragment avec la commande **xscmd -c swapShardWithPrimary**.

  – L'équilibrage déclenché par le placement de fragments avec la commande **xscmd -c triggerPlacement -g myOG -ms myMapSet**.

Que faire ensuite

Vous pouvez surveiller le placement de votre environnement à l'aide de la commande **xscmd -c placementServiceStatus**.

Gestion de la disponibilité ObjectGrid

L'état de disponibilité d'une instance ObjectGrid détermine les requêtes pouvant être traitées à tout moment. Vous pouvez utiliser l'interface de StateManager pour définir et extraire l'état d'une instance ObjectGrid.

Pourquoi et quand exécuter cette tâche

Il existe quatre états de disponibilité pour une instance ObjectGrid.

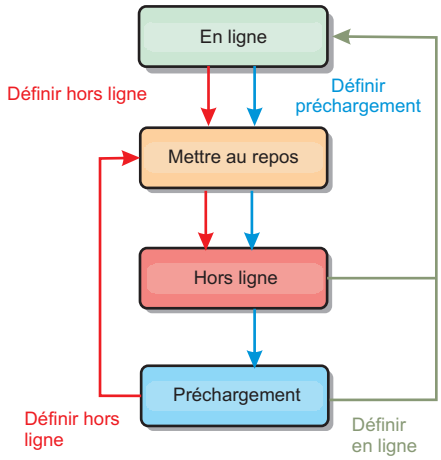


Figure 51. Etats de disponibilité d'une instance ObjectGrid

ONLINE
L'état ONLINE est l'état de disponibilité par défaut d'un ObjectGrid. Un ObjectGrid en ligne est capable de traiter n'importe quelle requête d'un client eXtreme Scale typique. Toutefois, les requêtes d'un client de préchargement sont rejetées lorsque l'ObjectGrid est en ligne.

QUIESCE
L'état QUIESCE est transitionnel. Un ObjectGrid qui a l'état QUIESCE passe rapidement à l'état OFFLINE. Lorsqu'un ObjectGrid a l'état QUIESCE, il peut traiter les transactions en attente. Toutefois, les nouvelles transactions sont rejetées. Un ObjectGrid peut rester au repos jusqu'à 30 secondes, Ensuite, l'état de disponibilité devient OFFLINE.

OFFLINE
L'état OFFLINE provoque le rejet de toutes les transactions envoyées à ObjectGrid.

PRELOAD
L'état PRELOAD (préchargement) peut servir à charger des données dans un ObjectGrid à partir d'un client de préchargement. Lorsque l'ObjectGrid est à l'état de préchargement, seul un client de préchargement peut valider des transactions par rapport à cet ObjectGrid. Toutes les autres transactions sont rejetées.

Une demande est rejetée si un ObjectGrid n'a pas l'état de disponibilité approprié pour la demande. Dans ce cas, une exception AvailabilityException est générée.

Procédure

1. Définissez l'état initial d'un ObjectGrid avec le fichier XML de configuration de l'ObjectGrid.

   Vous pouvez utiliser l'attribut **initialState** sur ObjectGrid pour indiquer son état au démarrage. Normalement, lorsqu'un ObjectGrid termine son initialisation, il devient disponible pour le routage. L'état peut ensuite être changé de manière à empêcher l'acheminement du trafic vers l'ObjectGrid. Si l'ObjectGrid doit être initialisé, mais pas immédiatement disponible, vous pouvez utiliser l'attribut **initialState**.

   L'attribut initialState est défini dans le fichier XML de configuration de l'ObjectGrid. L'état par défaut est ONLINE. Les valeurs admises sont les suivantes :

- ONLINE (par défaut)
- PRELOAD
- OFFLINE

Voir Fichier XML du descripteur d'ObjectGrid pour plus d'informations sur l'attribut **initialState**.

Si l'attribut initialState est défini dans un ObjectGrid, l'état doit être explicitement ramené à Online, car sinon l'ObjectGrid reste indisponible. Une exception AvailabilityException se produit si l'ObjectGrid n'a pas l'état.

Voir la AvailabilityState documentation d'API pour plus d'informations.

Utilisation de l'attribut initialState pour le préchargement

Si l'ObjectGrid est préchargé avec des données, un laps de temps est susceptible de s'écouler entre le moment où il est disponible et le moment où il passe à l'état de préchargement permettant de bloquer le trafic client. Pour éviter ce laps de temps, l'état initial d'un ObjectGrid peut être défini comme PRELOAD. L'ObjectGrid effectue toujours l'initialisation requise, mais bloque le trafic jusqu'au changement d'état et permet au préchargement d'avoir lieu.

Les états PRELOAD et OFFLINE bloquent le trafic, mais seul l'état PRELOAD permet de lancer un préchargement.

Basculement et équilibrage

Si une grille de données de réplique est promue pour être une grille de données primaire, la réplique n'utilise pas le paramètre **initialState**. Si la grille de données primaire est déplacée pour cause de rééquilibrage, le paramètre **initialState** n'est pas utilisé, car les données sont copiées vers le nouvel emplacement primaire avant la fin du transfert Si la réplication n'est pas configurée, le fragment primaire passe à l'état **initialState** en cas de basculement et un nouveau fragment primaire doit être placé.

2. Changez l'état de disponibilité avec l'interface StateManager.

   Utilisez l'interface de StateManager pour définir l'état de disponibilité d'un ObjectGrid. Pour définir l'état de disponibilité d'un ObjectGrid exécuté sur les serveurs, transmettez un client ObjectGrid correspondant à l'interface de StateManager. Le code suivant démontre comment changer l'état de disponibilité d'un ObjectGrid.

   ```
   Client ClientClusterContext = ogManager.connect("localhost:2809", null, null);
   ObjectGrid myObjectGrid = ogManager.getObjectGrid(client, "myObjectGrid");
   StateManager stateManager = StateManagerFactory.getStateManager();
   stateManager.setObjectGridState(AvailabilityState.OFFLINE, myObjectGrid);
   ```

   Chaque fragment de l'ObjectGrid passe à l'état à appliquer lorsque la méthode setObjectGridState est appelée sur l'interface de StateManager. Lorsque la méthode est renvoyée, tous les fragments de l'ObjectGrid doivent être définis sur l'état adéquat.

   Utilisez un plug-in ObjectGridEventListener pour changer l'état de disponibilité d'un ObjectGrid côté serveur. Changez l'état de disponibilité d'un ObjectGrid côté serveur seulement lorsque ce dernier présente une partition unique. Si l'ObjectGrid présente plusieurs partitions, la méthode shardActivated est appelée sur chaque partition principale, ce qui entraîne des appels superflus pour le changement d'état de l'ObjectGrid

   ```
   public class OGListener implements ObjectGridEventListener,
      ObjectGridEventGroup.ShardEvents {
       public void shardActivated(ObjectGrid grid) {
               StateManager stateManager = StateManagerFactory.getStateManager();
               stateManager.setObjectGridState(AvailabilityState.PRELOAD, grid);
       }
   }
   ```

L'état QUIESCE étant transitionnel, vous ne pouvez pas utiliser l'interface de StateManager pour définir l'état d'un ObjectGrid sur QUIESCE. L'ObjectGrid passe par cet état avant d'être défini sur l'état OFFLINE.

3. Extrayez l'état de disponibilité.

   Utilisez la méthode getObjectGridState de l'interface de StateManager pour récupérer l'état de disponibilité d'un ObjectGrid.

   ```
   StateManager stateManager = StateManagerFactory.getStateManager();
   AvailabilityState state = stateManager.getObjectGridState(inventoryGrid);
   ```

   La méthode getObjectGridState choisit une partition principale de l'ObjectGrid au hasard et renvoie son état de disponibilité. Tous les fragments ObjectGrid doivent présenter le même état ou être en transition vers le même état. C'est pour cela que cette méthode propose un résultat acceptable pour l'état de disponibilité actuel de l'ObjectGrid.

Gestion des incidents du centre de données

Lorsque le centre de données entre un scénario d'échec, envisagez de remplacer de quorum de sorte que les événements du serveur de conteneur ne soient pas ignorés. Vous pouvez utiliser l'utilitaire **xscmd** pour obtenir des informations sur les tâches de quorum et exécuter ces tâches, telles que l'état du quorum et le remplacement du quorum.

Avant de commencer

- Configurez le mécanisme de quorum qui doit être identique dans tous les serveurs de catalogue. Pour plus d'informations, voir Configuration du mécanisme de quorum.
- Le quorum est le nombre minimum de serveurs de catalogue nécessaires à l'exécution des opérations de positionnement pour la grille de données et représente l'ensemble des serveurs de catalogue, sauf si vous définissez un nombre inférieur. WebSphere eXtreme Scale s'attend à perdre le quorum pour les raisons suivantes :
  - défaillance d'un membre machine virtuelle Java du service de catalogue
  - microcoupure réseau
  - perte de centre de données

  Le message suivant indique que le quorum a été perdu. Recherchez la présence éventuelle de ce message dans les journaux de vos services de catalogue.

  ```
  CWOBJ1254W: Le service de catalogue attend un quorum.
  ```

Pourquoi et quand exécuter cette tâche

Remplacement du quorum dans un scénario de défaillance du centre de données uniquement. Lorsque vous remplacez le quorum, n'importe quelle instance de serveur de catalogue restante peut être utilisée. Tous les survivants sont notifiés lorsque l'un d'entre eux reçoit l'injonction de redéfinir le quorum.

Procédure

- Interrogation de l'état du quorum avec l'utilitaire **xscmd**.

  ```
  xscmd -c showQuorumStatus -cep cathost:2809
  ```

  Utilisez cette option pour afficher l'état du quorum d'une instance de service de catalogue. L'une des sorties suivantes s'affiche :

– `Le quorum est désactivé` : les serveurs de catalogue s'exécutent en mode Quorum désactivé. Ce mode est un mode de développement ou un mode de centre de données unique. Ne l'utilisez pas pour plusieurs configurations de centre de données.
– `Le quorum est activé et le serveur de catalogue a le quorum` : le quorum est activé et le système fonctionne normalement.
– `Le quorum est activé et le serveur de catalogue attend un quorum` : le quorum est activé et le quorum a été perdu.
– `Le quorum est activé et le quorum est remplacé` : le quorum est activé et le quorum n'a pas été remplacé.
– `L'état du quorum est proscrit` : lorsqu'une microcoupure se produit, le service de catalogue est scindé en deux partitions A et B. Le serveur de catalogue A a redéfini le quorum. La partition réseau se résout et le serveur dans la partition B est proscrit, nécessitant un redémarrage des machines virtuelles Java. Cela se produit également si la machine virtuelle Java du catalogue redémarre pendant la microcoupure et que cette dernière se termine.

• Remplacez le quorum avec l'utilitaire **xscmd**.

```
xscmd -c overrideQuorum -cep cathost:2809
```

L'exécution de cette commande oblige le serveur de catalogue restant à rétablir un quorum.

• Diagnostiquez un quorum avec l'utilitaire **xscmd**.
– **Affichage de la liste des groupes centraux :**

Utilisez l'option **-c listCoreGroups** pour afficher la liste de tous les groupes principaux du serveur de catalogue.

```
xscmd -c listCoreGroups -cep cathost:2809
```

– **Désassemblage des serveurs :**

Utilisez l'option **–c teardown** pour supprimer manuellement un serveur de la grille de données. Le retrait d'un serveur de la grille est généralement inutile. Les serveurs sont automatiquement supprimés lorsqu'ils sont déclarés comme étant défaillants, mais la commande est fournie pour être utilisée sous le contrôle du support IBM. Voir «Arrêt propre des serveurs avec l'utilitaire **xscmd**», à la page 409 pour plus d'informations sur l'utilisation de cette commande.

```
xscmd –c teardown server1,server2,server3 -cep cathost:2809 –g Grid
```

– **Affichage de la table de routage :**

Utilisez l'option **-c routetable** pour afficher la table de routage en cours en simulant une nouvelle connexion client à la grille de données. Elle valide également la table de routage en confirmant que tous les serveurs conteneurs reconnaissent bien leur rôle dans la table (par exemple, quel type de fragment pour quelle partition).

```
xscmd -c routetable -cep cathost:2809 –g myGrid
```

– **Vérification des tailles de mappes :**

Utilisez l'option **-c showMapSizes** pour vérifier que la distribution des clés est uniforme sur les fragments dans la clé. Si certains serveurs de conteneur ont plusieurs clés que d'autres, il est probable que la distribution de la fonction de hachage sur les objets key est incorrecte.

```
xscmd -c showMapSizes -cep cathost:2809 -g myGrid -ms myMapSet
```

– **Définition des chaînes de trace :**

Utilisez l'option **-c setTraceSpec** pour définir les paramètres de trace de toutes les machines virtuelles Java qui correspondent au filtre spécifié pour la commande **xscmd**. Ce paramètre modifie les paramètres de trace uniquement, jusqu'à ce qu'une autre commande soit utilisée ou que les machines virtuelles modifiées soient défaillantes ou s'arrêtent.

```
xscmd -c setTraceSpec -spec ObjectGrid*=event=enabled -cep cathost:1099
–g myGrid –hf host1
```

Cette chaîne permet de tracer toutes les machines virtuelles Java sur le serveur avec le nom d'hôte indiqué, `host1`, en l'occurrence.

– **Affichage des fragments non affectés :**

Utilisez l'option **-c showPlacement -sf U** pour afficher la liste des fragments qui ne peuvent pas être placés sur la grille de données. Les fragments ne peuvent pas être placés lorsque le service de placement a une contrainte qui empêche le placement. Par exemple, si vous démarrez des machines virtuelles sur un serveur physique unique en mode de production, seuls seront placés les fragments primaires. Les répliques ne sont pas affectées jusqu'à ce que les machines JVM démarrent sur un second serveur physique. Le service de placement place les répliques uniquement sur les machines JVM avec des adresses IP différentes de celles qui hébergent les fragments primaires. L'absence de machines virtuelles Java dans une zone peut également provoquer la non-attribution de fragments.

```
xscmd -c showPlacement -sf U  -cep cathost:2809 –g myGrid
```

Administration avec les beans gérés (MBeans)

Vous pouvez utiliser plusieurs types de beans gérés JMX (Java Management Extensions) différents pour administrer et surveiller les déploiements. Chaque bean géré fait référence à une entité spécifique, une mappe, une grille de données, un serveur ou un service.

Interfaces MBean JMX et WebSphere eXtreme Scale

Chaque bean géré contient des méthodes get qui représentent des valeurs d'attribut. Ces méthodes get ne peuvent pas être appelées directement à partir de votre programme. La spécification JMX traite les attributs différemment des opérations. Vous pouvez afficher les attributs à l'aide de la console JMX d'un fournisseur et effectuer des opérations dans votre programme ou à l'aide de la console JMX d'un fournisseur.

Package com.ibm.websphere.objectgrid.management

Voir la documentation d'API pour une présentation et les spécifications détaillées pour la programmation de tous les beans gérés utilisables : Package com.ibm.websphere.objectgrid.management .

Accès aux beans gérés (MBeans) à l'aide de l'outil wsadmin

Vous pouvez utiliser l'utilitaire wsadmin fourni dans WebSphere Application Server pour accéder aux informations des beans gérés (MBean).

Procédure

Exécutez l'outil wsadmin depuis le répertoire `bin` dans votre installation WebSphere Application Server. L'exemple suivant restaure une vue de la position actuelle du fragment dans un logiciel eXtreme Scale dynamique. Vous pouvez

exécuter l'outil wsadmin depuis n'importe quelle installation où eXtreme Scale est en cours d'exécution. Vous n'avez pas besoin d'exécuter l'outil wsadmin sur le service de catalogue.

```
$ wsadmin.sh -lang jython
wsadmin>placementService = AdminControl.queryNames
 ("com.ibm.websphere.objectgrid:*,type=PlacementService")
wsadmin>print AdminControl.invoke(placementService,
 "listObjectGridPlacement","library ms1")

<objectGrid name="library" mapSetName="ms1">
  <container name="container-0" zoneName="DefaultDomain"
  hostName="host1.company.org" serverName="server1">
     <shard type="Primary" partitionName="0"/>
     <shard type="SynchronousReplica" partitionName="1"/>
  </container>
  <container name="container-1" zoneName="DefaultDomain"
  hostName="host2.company.org" serverName="server2">
     <shard type="SynchronousReplica" partitionName="0"/>
     <shard type="Primary" partitionName="1"/>
  </container>
  <container name="UNASSIGNED" zoneName="_ibm_SYSTEM"
  hostName="UNASSIGNED" serverName="UNNAMED">
    <shard type="SynchronousReplica" partitionName="0"/>
    <shard type="AsynchronousReplica" partitionName="0"/>
  </container>
</objectGrid>
```

Accès aux beans gérés (MBeans) à l'aide d'un programme

Vous pouvez vous connecter aux beans gérés avec des applications Java. Ces applications utilisent les interfaces dans le package com.ibm.websphere.objectgrid.management.

Pourquoi et quand exécuter cette tâche

Les méthodes d'accès à l'aide d'un programme aux beans gérés varient en fonction du type de serveur auquel vous vous connectez.

- Connexion à un serveur MBean de service de catalogue
- Connexion à un serveur MBean de conteneur
- Connexion à un serveur MBean de service de catalogue hébergé dans WebSphere Application Server
- Connexion à un serveur Mbean de service de catalogue avec la sécurité activée

Procédure

- **Connectez-vous à un serveur MBean de service de catalogue autonome :**

  L'exemple de programme suivant se connecte à un serveur MBean de service de catalogue autonome et renvoie une chaîne formatée XML qui répertorie chaque serveur conteneur avec ses fragments alloués pour ObjectGrid et un MapSet donné.

```
package com.ibm.websphere.sample.xs.admin;

import java.util.Set;

import javax.management.MBeanServerConnection;
import javax.management.ObjectName;
import javax.management.remote.JMXConnector;
import javax.management.remote.JMXConnectorFactory;
import javax.management.remote.JMXServiceURL;

/**
 * Collecte les informations de placement à partir du serveur de catalogue pour un ObjectGrid donné.
 */
public final class CollectPlacementPlan {
    private static String hostName = "localhost";

    private static int port = 1099;

    private static String objectGridName = "library";

    private static String mapSetName = "ms1";

    /**
     * Se connecte au service de catalogue ObjectGrid pour extraire les informations de placement et
    * les affiche.
     *
     * @param args
     * @throws Exception
     *             If there is a problem connecting to the catalog service MBean server.
     */
    public static void main(String[] args) throws Exception {
        String serviceURL = "service:jmx:rmi:///jndi/rmi://" + hostName + ":" + port +
                    "/objectgrid/MBeanServer";
        JMXServiceURL jmxUrl = new JMXServiceURL(serviceURL);
        JMXConnector jmxCon = JMXConnectorFactory.connect(jmxUrl);

        try {
            MBeanServerConnection catalogServerConnection = jmxCon.getMBeanServerConnection();

            Set placementSet = catalogServerConnection.queryNames(new ObjectName
           ("com.ibm.websphere.objectgrid"
             + ":*,type=PlacementService"), null);
            ObjectName placementService = (ObjectName) placementSet.iterator().next();
            Object placementXML = catalogServerConnection.invoke(placementService,
                "listObjectGridPlacement", new Object[] {
                objectGridName, mapSetName }, new String[] { String.class.getName(),
                String.class.getName() });
            System.out.println(placementXML);
        } catch (Exception e) {
            if(jmxCon != null) {
                jmxCon.close();
            }
        }
    }

}
```

Figure 52. CollectPlacementPlan.java

Quelques remarques concernant l'exemple de programme :

– La valeur **JMXServiceURL** pour le service de catalogue a toujours le format suivant : `service:jmx:rmi:///jndi/rmi://<host>:<port>/objectgrid/MBeanServer`, où `<host>` est l'hôte sur lequel le service de catalogue est exécuté et `<port>` est le port du service JMX fourni avec l'option **-JMXServicePort** lors du démarrage du service de catalogue. Si aucun port n'est défini, la valeur par défaut est `1099`.

– Pour activer les statistiques ObjectGrid ou de mappe, vous devez définir la propriété suivantes dans le fichier des propriétés du serveur lorsque vous démarrez un conteneur ObjectGrid : `statsSpec=all=enabled`
– Pour désactiver les beans gérés exécutés dans les serveurs de conteneur, définissez la propriété suivante dans le fichier des propriétés du serveur : `enableMBeans=false`.

Exemple de sortie. Cette sortie indique que deux serveurs de conteneur sont actifs. Le serveur de conteneur `Container-0` héberge quatre fragments primaires. Le serveur de conteneur `Container-1` héberge une réplique synchrone pour chaque fragment primaire sur le serveur de conteneur `Container-0`. Dans cette configuration, deux répliques synchrones et une réplique asynchrone sont configurées. Par conséquent, le serveur de conteneur `Unassigned` dispose des fragments restants. Si plus de deux serveurs sont démarrés, le serveur de conteneur `Unassigned` n'est pas affiché.

```
<objectGrid name="library" mapSetName="ms1">
  <container name="Container-1" zoneName="DefaultZone"
    hostName="myhost.mycompany.com" serverName="ogserver">
    <shard type="SynchronousReplica" partitionName="0"/>
    <shard type="SynchronousReplica" partitionName="1"/>
    <shard type="SynchronousReplica" partitionName="2"/>
    <shard type="SynchronousReplica" partitionName="3"/>
  </container>
  <container name="Container-0" zoneName="DefaultZone"
    hostName="myhost.mycompany.com" serverName="ogserver">
    <shard type="Primary" partitionName="0"/>
    <shard type="Primary" partitionName="1"/>
    <shard type="Primary" partitionName="2"/>
    <shard type="Primary" partitionName="3"/>
  </container>
  <container name="library:ms1:_UnassignedContainer_" zoneName="_ibm_SYSTEM"
    hostName="UNASSIGNED" serverName="UNNAMED">
    <shard type="SynchronousReplica" partitionName="0"/>
    <shard type="SynchronousReplica" partitionName="1"/>
    <shard type="SynchronousReplica" partitionName="2"/>
    <shard type="SynchronousReplica" partitionName="3"/>
    <shard type="AsynchronousReplica" partitionName="0"/>
    <shard type="AsynchronousReplica" partitionName="1"/>
    <shard type="AsynchronousReplica" partitionName="2"/>
    <shard type="AsynchronousReplica" partitionName="3"/>
  </container>
</objectGrid>
```

- **Connectez-vous à un serveur MBean de conteneur :**

  Les serveurs de conteneur hébergent des beans gérés pour obtenir des informations sur les mappes et les instances ObjectGrid individuelles exécutées dans le serveur de conteneur. L'exemple de programme suivant affiche l'état de chaque serveur de conteneur hébergé par le serveur de catalogue avec l'adresse JMX `localhost:1099`:

```
package com.ibm.websphere.sample.xs.admin;

import java.util.List;
import java.util.Set;

import javax.management.MBeanServerConnection;
import javax.management.ObjectInstance;
import javax.management.ObjectName;
import javax.management.remote.JMXConnector;
import javax.management.remote.JMXConnectorFactory;
import javax.management.remote.JMXServiceURL;

/**
 * Collecte l'état du placement directement depuis les conteneurs disponibles.
 */
public final class CollectContainerStatus {
    private static String hostName = "localhost";

    private static int port = 1099;

    /**
     * @param args
     */
    public static void main(String[] args) throws Exception {
        String serviceURL = "service:jmx:rmi:///jndi/rmi://" + hostName + ":" + port + "/objectgrid/MBeanServer";
        JMXServiceURL jmxUrl = new JMXServiceURL(serviceURL);
        JMXConnector jmxCon = JMXConnectorFactory.connect(jmxUrl);

        try {
            MBeanServerConnection catalogServerConnection = jmxCon.getMBeanServerConnection();

            Set placementSet = catalogServerConnection.queryNames(new ObjectName("com.ibm.websphere.objectgrid"
                    + ":*,type=PlacementService"), null);

            ObjectName placementService = (ObjectName) placementSet.iterator().next();
            List<String> containerJMXAddresses = (List<String>) catalogServerConnection.invoke(placementService,
                    "retrieveAllServersJMXAddresses", new Object[0], new String[0]);
            for (String address : containerJMXAddresses) {
                JMXServiceURL containerJMXURL = new JMXServiceURL(address);
                JMXConnector containerConnector = JMXConnectorFactory.connect(containerJMXURL);
                MBeanServerConnection containerConnection = containerConnector.getMBeanServerConnection();
                Set<ObjectInstance> containers = containerConnection.queryMBeans(
                        new ObjectName("*:*,type=ObjectGridContainer"), null);
                for (ObjectInstance container : containers) {
                    System.out.println(containerConnection.getAttribute(container.getObjectName(), "Status"));
                }
            }
        } finally {
            if(jmxCon != null) {
                jmxCon.close();
            }
        }
    }
}
```

Figure 53. CollectContainerStatus.java

L'exemple de programme affiche l'état du serveur de chaque conteneur. Ci-après, un exemple de sortie :

```
<container name="Container-0" zoneName="DefaultZone" hostName="descartes.rchland.ibm.com"
  serverName="ogserver">
  <shard type="Primary" partitionName="1"/>
  <shard type="Primary" partitionName="0"/>
  <shard type="Primary" partitionName="3"/>
  <shard type="Primary" partitionName="2"/>
</container>
```

- **Connectez-vous à un serveur MBean de serveur de catalogue hébergé dans WebSphere Application Server :**

  La méthode d'accès à l'aide d'un programme aux beans gérés dans MBeans WebSphere Application Server diffère légèrement de la méthode d'accès aux beans gérés dans une configuration autonome.

1. Créez et compilez un programme Java pour vous connecter au serveur MBean. Exemple de programme :

```
package com.ibm.websphere.sample.xs.admin;

import java.util.Set;

import javax.management.MBeanServerConnection;
import javax.management.ObjectName;
import javax.management.remote.JMXConnector;
import javax.management.remote.JMXConnectorFactory;
import javax.management.remote.JMXServiceURL;

/**
 * Collecte les informations de placement à partir du serveur de catalogue exécuté dans un
 * gestionnaire de déploiement pour un ObjectGrid donné.
 */
public final class CollectPlacementPlanWAS {
    private static String hostName = "localhost";

    private static int port = 9809;

    private static String objectGridName = "library";

    private static String mapSetName = "ms1";

    /**
     * Connects to the catalog service to retrieve placement information and prints it out.
     *
     * @param args
     * @throws Exception
     *             If there is a problem connecting to the catalog service MBean server.
     */
    public static void main(String[] args) throws Exception {

        // connect to bootstrap port of the deployment manager
        String serviceURL = "service:jmx:iiop://" + hostName + ":" + port + "/jndi/JMXConnector";
        JMXServiceURL jmxUrl = new JMXServiceURL(serviceURL);
        JMXConnector jmxCon = JMXConnectorFactory.connect(jmxUrl);

        try {
            MBeanServerConnection catalogServerConnection = jmxCon.getMBeanServerConnection();

            Set placementSet = catalogServerConnection.queryNames(new ObjectName("com.ibm.websphere.objectgrid"
             + ":*,type=PlacementService"), null);

            ObjectName placementService = (ObjectName) placementSet.iterator().next();
            Object placementXML = catalogServerConnection.invoke(placementService,
                        "listObjectGridPlacement", new Object[] {
                    objectGridName, mapSetName }, new String[] { String.class.getName(),
                    String.class.getName() });
            System.out.println(placementXML);
        } finally {
            if(jmxCon != null) {
                jmxCon.close();
            }
        }
    }

}
```

Figure 54. CollectPlacementPlan.java

2. Exécutez la commande suivante.

```
"$JAVA_HOME/bin/java" "$WAS_LOGGING" -Djava.security.auth.login.config="$app_server_root/properties/wsjaas_client.conf" \
 -Djava.ext.dirs="$JAVA_HOME/jre/lib/ext:$WAS_EXT_DIRS:$WAS_HOME/plugins:$WAS_HOME/lib/WMQ/java/lib" \
 -Djava.naming.provider.url=<an_IIOP_URL_or_a_corbaloc_URL_to_your_application_server_machine_name> \
 -Djava.naming.factory.initial=com.ibm.websphere.naming.WsnInitialContextFactory \
 -Dserver.root="$WAS_HOME" "$CLIENTSAS" "$CLIENTSSL" $USER_INSTALL_PROP \
 -classpath "$WAS_CLASSPATH":<list_of_your_application_jars_and_classes> \
 <fully_qualified_class_name_to_run> <your_application_parameters>
```

Cette commande suppose que le script *racine_was*/bin/setupCmdLine.sh a été exécuté pour définir correctement les variables.

`corbaloc:iiop:1.0@<host>:<port>/NameService` est un exemple de format de valeur de propriété java.naming.provider.url.

- **Connectez-vous à un serveur MBean de service de catalogue avec la sécurité activée :**

  Pour plus d'informations sur la connexion au bean géré du service de catalogue avec la sécurité activée, voir «Sécurité JMX (Java Management Extensions)», à la page 517.

Que faire ensuite

Pour d'autres exemples sur la manière d'afficher les statistiques et effectuer des opérations d'administration avec des beans gérés, voir l'exemple d'application **xsadmin**. Vous pouvez analyser le code source de l'exemple d'application xsadmin dans le fichier *rép_base_wxs*/samples/xsadmin.jar d'une installation autonome ou dans le fichier *rép_base_wxs*/xsadmin.jar dans une installation WebSphere Application Server. Voir Exemple : utilitaire **xsadmin** pour plus d'informations sur les opérations que vous pouvez exécuter avec l'exemple d'application **xsAdmin**.

Des informations supplémentaires sont également disponibles sur les beans gérés dans le package com.ibm.websphere.objectgrid.management.

Chapitre 8. Contrôle

Vous pouvez utiliser la console de surveillance, les API, les beans gérés, les journaux et les utilitaires inclus pour surveiller les performances de votre environnement d'application.

Présentation des statistiques

Les statistiques dans WebSphere eXtreme Scale sont basés sur une arborescence interne de statistiques. L'API StatsAccessor, les modules PMI (Performance Monitoring Infrastructure) et l'API MBean sont générés à partir de l'arborescence interne.

L'illustration suivante montre la configuration générale des statistiques pour WebSphere eXtreme Scale.

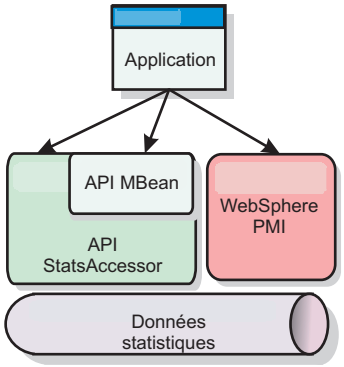


Figure 55. Présentation des statistiques

Toutes ces API permettent de visualiser l'arborescence des statistiques, mais chacune d'entre elles possède une fonction spécifique :

- **API Statistics**: L'API Statistics permet aux développeurs d'accéder directement aux statistiques pour des solutions flexibles et personnalisables d'intégration de statistiques, tels que des beans gérés personnalisés ou la consignation.
- **API MBean :** cette API est un mécanisme de surveillance basé sur une spécification. Elle utilise l'API Statistics et s'exécute de manière locale sur la machine virtuelle Java du serveur. Les structures de l'API et des beans gérés sont conçues pour s'intégrer aisément à des utilitaires tiers. Utilisez l'API MBean lorsque vous exécutez une grille d'objets répartie.
- **WebSphere Application Server Modules PMI :** utilisez ces modules si vous exécutez WebSphere eXtreme Scale dans WebSphere Application Server. Ces modules permettent de visualiser l'arborescence interne des statistiques.

API Statistics

A l'instar d'une mappe d'arborescence, il existe un chemin et une clé correspondants qui permettent d'extraire un module spécifique ou, dans ce cas, le niveau de granularité ou d'agrégation. Par exemple, supposons que l'arborescence contienne toujours un nœud racine arbitraire et que les statistiques soient regroupées pour une mappe appelée "payroll" appartenant à une instance

d'ObjectGrid appelée "accounting". Par exemple, pour accéder au module en fonction du niveau d'agrégation ou de granularité d'une mappe, vous pouvez insérer un paramètre String[] des chemins. Dans ce cas, vous obtenez String[] {root, "accounting", "payroll"}, chaque paramètre String représentant le chemin du nœud. Cette structure a pour avantage de permettre à l'utilisateur de spécifier le tableau dans un nœud quelconque du chemin et d'obtenir le niveau d'agrégation du nœud en question. L'insertion du paramètre String[] {root, "accounting"} vous permet d'obtenir les statistiques de mappe, sauf pour la grille entière de "accounting". L'utilisateur peut ainsi spécifier les types de statistiques à surveiller, ainsi que le niveau d'agrégation nécessaire pour l'application.

WebSphere Application ServerModules PMI

WebSphere eXtreme Scale inclut des modules de statistiques à utiliser avec l'infrastructure PMI WebSphere Application Server. Lorsqu'une instance de WebSphere eXtreme Scale est ajoutée à un profil WebSphere Application Server, les scripts d'ajout intègrent automatiquement les modules WebSphere eXtreme Scale dans les fichiers de configuration WebSphere Application Server. PMI vous permet d'activer et de désactiver les modules de statistiques, d'assembler automatiquement les statistiques selon différents niveaux de granularité et même de représenter les données sous forme de graphiques à l'aide du logiciel pré-intégré Tivoli Performance Viewer. Pour plus d'informations, voir «Surveillance à l'aide de la fonction PMI de WebSphere Application Server», à la page 463.

Intégration de produits tiers avec les beans gérés (MBean)

Les API eXtreme Scale et les beans gérés sont conçus pour faciliter l'intégration d'applications de surveillance tierces. JConsole et MC4J sont des exemples de consoles Java Management Extensions (JMX) légères qui permettent d'analyser les informations relatives à une topologie eXtreme Scale. Vous avez également la possibilité d'utiliser les API de programmation pour écrire des implémentations d'adaptateur afin de créer des instantanés ou d'effectuer un suivi des performances d'eXtreme Scale. WebSphere eXtreme Scale inclut un exemple d'application de surveillance qui permet d'effectuer la surveillance dès l'installation et qui peut servir de modèle pour créer des utilitaires de surveillance plus avancés.

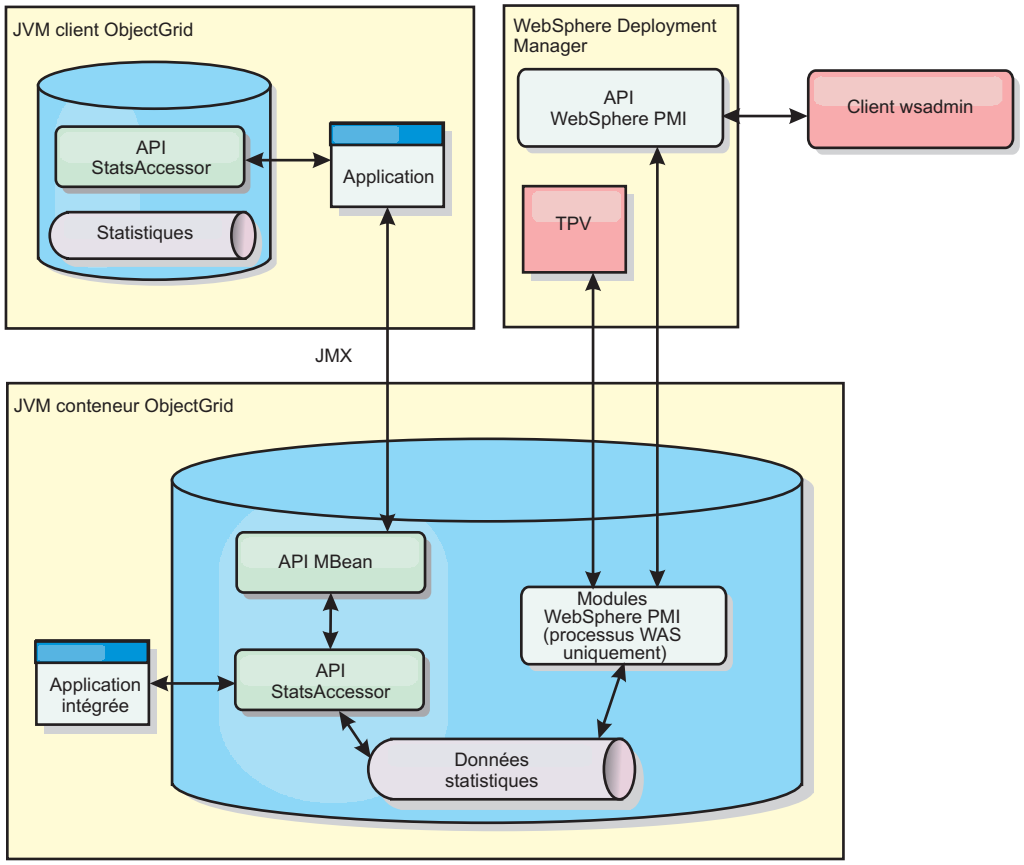


Figure 56. Présentation de l'API de bean géré

Pour plus d'informations, voir Exemple : utilitaire **xsadmin**. Pour plus d'informations sur l'intégration d'applications tierces spécifiques, voir les rubriques suivantes :

- Surveillance d'eXtreme Scale à l'aide d'un agent de surveillance IBM Tivoli
- «Surveillance d'eXtreme Scale à l'aide de Hyperic HQ», à la page 486
- «Surveillance des applications eXtreme Scale à l'aide de CA Wily Introscope», à la page 483

Surveillance à l'aide de la console Web

Avec la console Web, vous pouvez générer des graphiques des statistiques actuelles et historiques. Cette console fournit un certain nombre de graphiques préconfigurés pour des présentations générales et elle comporte une page de rapports personnalisés que vous pouvez utiliser pour élaborer des graphiques à partir des statistiques disponibles. Les fonctionnalités graphiques de la console de surveillance de WebSphere eXtreme Scale permettent de visualiser les performances globales des grilles des données présentes dans votre environnement.

Démarrage et consignation sur la console Web

Démarrez le serveur de la console en exécutant la commande **startConsoleServer** et en vous connectant au serveur en utilisant l'ID utilisateur et le mot de passe par défaut.

Avant de commencer

- **Configuration requise**
  - Utilisez un système d'exploitation AIX, Linux ou Windows pour exécuter la console Web.
  - Installez un serveur autonome WebSphere eXtreme Scale sur le système qui héberge le serveur de la console. Pour plus d'informations, voir «Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client autonomes», à la page 194.
  - Le système du serveur de la console doit pouvoir se connecter à votre service de catalogue. Le service de catalogue doit lui aussi être capable de se connecter au serveur de console Web.
- **Navigateurs Web requis**

  Utilisez l'un des navigateurs suivants avec la console Web :
  - Mozilla Firefox, version 3.5.x et versions ultérieures
  - Mozilla Firefox, version 3.6.x et versions ultérieures
  - Microsoft Internet Explorer version 7 ou 8

Procédure

1. Facultatif : Si vous voulez exécuter votre serveur de console sur un port autre que le port par défaut, éditez le fichier *racine_install_wxs*`/ObjectGrid/console/config/zero.config`. Le port par défaut pour le serveur de la console est `7080` pour HTTP et `7443` pour HTTPS. Vous pouvez éditer les propriétés suivantes pour modifier les valeurs par défaut :

   ```
   /config/http/port = 7080
   /config/https/port = 7443
   ```

   Si vous éditez ces valeurs une fois que le serveur de la console est démarré, redémarrez le serveur pour utiliser les nouveaux numéros de port.
2. Démarrez le serveur de la console. Le script **`startConsoleServer.bat|sh`** de démarrage du serveur de la console se trouve dans le répertoire *racine_install_wxs*`/ObjectGrid/bin` de votre installation.
3. Connectez-vous à la console.
   a. Dans votre navigateur Web, accédez à `https://your.console.host:7443`, en remplaçant `your.console.host` par le nom de l'hôte du serveur sur lequel vous avez installé la console.
   b. Connectez-vous à la console.
      - **ID utilisateur :** `admin`
      - **Mot de passe :** `admin`

      La page d'accueil de la console s'affiche.
4. Modifiez la configuration de la console. Cliquez sur **Paramètres** > **Configuration** pour afficher la configuration de la console. La configuration de la console comprend ce type d'informations :
   - la chaîne de trace pour le client WebSphere eXtreme Scale, comme `*=all=disabled`
   - le nom et le mot de passe de l'administrateur
   - son adresse e-mail

Que faire ensuite

- Connectez vos serveurs de catalogue à la console Web pour démarrer le suivi des statistiques. Pour plus d'informations, voir «Connexion de la console Web aux serveurs de catalogue».
- Si vous avez besoin d'arrêter le serveur de la console Web, exécutez le script **stopConsoleServer.bat|sh**. Ce script se trouve dans le répertoire *racine_install_wxs*/ObjectGrid/bin de votre installation.

Connexion de la console Web aux serveurs de catalogue

Pour démarrer les statistiques d'affichage dans la console Web, vous devez d'abord vous connecter aux serveurs de catalogue que vous voulez surveiller. Des étapes supplémentaires sont requises si la sécurité est activée sur les serveurs de catalogue.

Avant de commencer

- Le serveur de console Web doit être actif. Pour plus d'informations, voir «Démarrage et consignation sur la console Web», à la page 443.
- Vous devez disposer d'au moins un serveur de catalogue en cours d'exécution auquel vous voulez vous connecter. Pour plus d'informations, voir «Démarrage d'un service de catalogue autonome», à la page 395.

Procédure

1. Si SSL (Secure Sockets Layer) est activé sur vos serveurs de catalogue, vous devez configurer un fichier de clés, un fichier de clés certifiées et un fichier de propriétés client. Vous activez SSL pour un serveur de catalogue en affectant à l'attribut TransportType la valeur SSL-Required dans Fichier de propriétés du serveur.
   a. Configurez un fichier de clés et un fichier de clés certifiées, puis échangez ou les certificats publics ou importez-les. Par exemple, vous pouvez copier le fichier de clés et le fichier de clés certifiées vers un emplacement sur le serveur qui exécute la console Web.
   b. Editez le fichier de propriétés du client sur le serveur de la console Web pour inclure les propriétés de configuration SSL. Par exemple, vous pouvez éditez le fichier *racine_install_wxs*/ObjectGridProperties/sampleclient.properties. Les propriétés suivantes sont requises pour les connexions SSL sortantes à partir de la console Web :

```
#-------------------------------------------------------------------------------
# SSL Configuration
#
# - contextProvider     (IBMJSSE2, IBMJSSE, IBMJSSEFIPS, etc.)
# - protocol            (SSL, SSLv2, SSLv3, TLS, TLSv1, etc.)
# - keyStoreType        (JKS, JCEK, PKCS12, etc.)
# - trustStoreType      (JKS, JCEK, PKCS12, etc.)
# - keyStore            (fully qualified path to key store file)
# - trustStore          (fully qualified path to trust store file)
# - alias               (string specifying ssl certificate alias to use from keyStore)
# - keyStorePassword    (string specifying password to the key store - encoded or not)
# - trustStorePassword  (string specifying password to the trust store - encoded or not)
#
# Uncomment these properties to set the SSL configuration.
#-------------------------------------------------------------------------------
#alias=clientprivate
#contextProvider=IBMJSSE
#protocol=SSL
#keyStoreType=JKS
#keyStore=etc/test/security/client.private
#keyStorePassword={xor}PDM2OjErLyg\=
#trustStoreType=JKS
#trustStore=etc/test/security/server.public
#trustStorePassword={xor}Lyo9MzY8
```

Important : Windows Si vous utilisez Windows, vous devez définir littéralement la barre oblique inverse (\) dans le chemin. Par exemple, si vous souhaitez utiliser le chemin d'accès `C:\opt\ibm`, entrez `C:\\opt\\ibm` dans le fichier des propriétés.

2. Créez et maintenez des connexions aux serveurs de catalogue que vous voulez surveiller. Répétez les étapes suivantes pour ajouter chaque serveur de catalogue à la configuration.
   a. Cliquez sur **Paramètres** > **Serveurs de catalogue eXtreme Scale**.
   b. Ajoutez un nouveau serveur de catalogue.

      1) Cliquez sur l'icône Ajouter ( ) pour enregistrer un serveur de catalogue existant.
      2) Fournissez des informations, telles que le nom d'hôte et le port d'écoute. Voir «Planification des ports réseau», à la page 64 pour plus d'informations sur la configuration des ports et les valeurs par défaut.
      3) Cliquez sur **OK**.
      4) Vérifiez que le serveur de catalogue a bien été ajouté à l'arborescence de navigation.
3. Regroupez les serveurs de catalogue que vous avez créés dans un domaine de services de catalogue. Vous devez créer un domaine de services de catalogue lorsque la sécurité est activée dans vos serveurs de catalogue, car les paramètres de sécurité sont configurés dans le domaine de services de catalogue.
   a. Cliquez sur la page **Paramètres** > **Domaines eXtreme Scale**.
   b. Ajoutez un nouveau service de catalogue.

      1) Cliquez sur l'icône Ajouter ( ) pour enregistrer un service de catalogue existant. Entrez le nom du domaine de services de catalogue.
      2) Une fois que vous avez créé le domaine de services de catalogue, vous pouvez modifier les propriétés. Les propriétés du domaine de services de catalogue sont les suivantes :

         Nom Indique le nom d'hôte du domaine, attribué par l'administrateur.

         Serveurs de catalogue
         Liste un ou plusieurs catalogues qui appartiennent au domaine sélectionné. Vous pouvez ajouter les serveurs de catalogue que vous avez créés dans l'étape précédente.

         Classe de génération
         Indique le nom de la classe qui implémente l'interface CredentialGenerator. Cette classe utilisée pour obtenir les données d'identification des clients. Si vous définissez une valeur dans cette zone, la valeur remplace la propriété **`crendentialGeneratorClass`** dans le fichier `client.properties`.

         Propriétés du générateur
         Spécifie les propriétés de la classe d'implémentation CredentialGenerator. Les propriétés correspondent à l'objet avec la méthode setProperties(String). La valeur credentialGeneratorprops n'est utilisée que si la valeur de la propriété credentialGeneratorClass n'est pas null. Si vous

définissez une valeur dans cette zone, la valeur remplace la propriété **credentialGeneratorProps** dans le fichier client.properties.

Chemin des propriétés du client eXtreme Scale
Indique le chemin d'accès au fichier des propriétés du client que vous avez modifié pour inclure les propriétés SSL dans une étape précédente. Par exemple, vous pouvez indiquer le fichier c:\ObjectGridProperties\sampleclient.properties. Si vous voulez empêcher la console d'utiliser les connexions SSL, vous pouvez supprimer la valeur dans cette zone. Après avoir défini le chemin, la console utilise une connexion non sécurisée.

3) Cliquez sur **OK**.
4) Vérifiez que le domaine a bien été ajouté à l'arborescence de navigation.

Pour afficher les informations concernant un domaine de services de catalogue existant, cliquez sur le nom du domaine de services de catalogue dans l'arborescence de navigation sur la page **Paramètres** > **Domaines eXtreme Scale**.

4. Visualisez le statut de la connexion La zone **Domaine en cours** indique le nom du domaine de services de catalogue qui est actuellement utilisé pour afficher des informations dans la console Web. L'état de la connexion s'affiche en regard du nom du domaine de services de catalogue.

Affichage des statistiques avec la console Web

Vous pouvez surveiller les statistiques et d'autres informations de performances avec la console Web.

Avant de commencer

Pour pouvoir afficher les statistiques avec la console Web, vous devez procéder comme suit :

1. Démarrez le serveur de console Web. Pour plus d'informations, voir «Démarrage et consignation sur la console Web», à la page 443.
2. Connectez vos serveurs de catalogue au serveur de la console Web. Pour plus d'informations, voir «Connexion de la console Web aux serveurs de catalogue», à la page 445.
3. Exécutez les grilles et les applications de données actives dans les serveurs qui sont gérés par votre domaine de services de catalogue.

Pourquoi et quand exécuter cette tâche

Une fois les grilles de données créées et les applications configurées pour utiliser ces grilles, laissez aux statistiques le temps d'être générées. Par exemple, avec une grille de données de caches dynamiques, les statistiques ne sont pas disponibles jusqu'à ce qu'un WebSphere Application Server qui exécute un cache dynamique se connecte au cache dynamique. En général, il suffit d'attendre environ une minute après l'apport d'une modification de configuration importante pour observer le changement au niveau des statistiques.

Conseil : Pour afficher des informations spécifiques sur un point de données ou un graphique, placez le pointeur de la souris sur l'élément.

Procédure

- Pour afficher les statistiques en cours du serveur, cliquez sur **Surveiller** > **Vue d'ensemble du serveur**.
- Pour afficher les performances de toutes vos grilles de données, cliquez sur **Surveiller** > **Vue d'ensemble des domaines de grilles de données**.
- Pour afficher des grilles de données individuelles, cliquez sur **Surveiller** > **Présentation de la grille de données** > *data_grid_name*. Cette page propose un récapitulatif incluant le nombre d'entrées en cache, la durée moyenne des transactions et le débit moyen.
- Pour afficher d'autres détails sur une grille de données spécifique, cliquez sur **Surveiller** > **Détails d'une grille de données**. Une arborescence affiche toutes les grilles de données de votre configuration. Vous explorer cette arborescence en aval et accéder à une grille de données spécifique afin d'afficher les mappes appartenant à cette grille de données. Vous pouvez cliquer sur le nom d'une grille de données ou d'une mappe pour obtenir plus d'informations.
- Pour choisir les statistiques à placer dans votre rapport personnalisé, cliquez sur **Contrôler** > **Rapports personnalisés**.

  Cette vue permet d'élaborer des graphiques détaillés à partir des diverses statistiques. L'arborescence permet d'explorer les grilles de données et les serveurs disponibles, ainsi que leurs statistiques. Un menu s'affiche lorsqu'on clique ou que l'on appuie sur Entrée sur un noeud qui référence des données pouvant être représentées dans un graphique. Vous pouvez créer un nouveau graphique contenant les statistiques ou, si elles sont compatibles, ajouter ces statistiques à celles d'un graphique existant. Pour plus d'informations, voir «Surveillance à l'aide de rapports personnalisés», à la page 454.

Statistiques de la console Web

En fonction de la vue que vous utilisez dans la console Web, vous pouvez afficher différentes statistiques relatives à votre configuration. Ces statistiques incluent la mémoire utilisée, les grilles de données les plus utilisées et le nombre d'entrées en mémoire cache.

- «Présentation du domaine de grille de données»
- «Présentation de la grille de données», à la page 449
- «Informations une grille de données», à la page 449
- «Présentation du serveur», à la page 450
- «Rapports personnalisés : statistiques du domaine de services de catalogue», à la page 450
  - «Rapports personnalisés : statistiques du serveur de conteneur», à la page 451
  - «Rapports personnalisés : statistiques de la grille de données», à la page 452
  - «Rapports personnalisés : statistiques de mappe», à la page 452

Présentation du domaine de grille de données

Les statistiques de présentation du domaine de la grille de données figurent dans la page **Surveiller** > **Présentation du domaine de la grille de données**. Cliquez sur l'un des onglets suivants pour plus d'informations sur le domaine de la grille de données :

Onglet Capacité utilisée

Le graphique de **distribution de la capacité utilisée de la grille de données en cours** contient une image du **pool total** et les **plus importants consommateurs de capacité utilisés**. Ne sont affichées que les 25 grilles de

données les plus consommatrices. Le graphique de **capacité utilisé dans le temps** indique le nombre d'octets consommés par la grille affichée.

Onglet Débit moyen
Le graphique des **cinq grilles de données les plus actives par temps moyen de transaction en millisecondes** contient la liste des cinq premiers caches de données organisés en fonction du temps de transaction moyen. Le graphique **Débit moyen au fil du temps** affiche les débits moyens, maximum et minimum au cours de la dernière heure, du dernier jour et de la dernière semaine.

Onglet Délai de transaction moyen
Le graphique des **cinq grilles de données les plus lentes** contient des données sur les grilles de données les plus lentes. Le graphique du **temps de transaction moyen dans le temps** indique les temps de transaction moyen, maximum et minimum au cours de la dernière heure, du dernier jour et de la dernière semaine.

Présentation de la grille de données

Pour afficher des statistiques sur une grille de données, cliquez sur **Surveiller** > **Présentation de la grille de données** > *data_grid_name*.

Récapitulatif en cours sur les 30 dernières secondes
Affiche le nombre actuel d'entrées de cache, le temps de transaction moyen, le débit moyen, et le nombre de réussites en mémoire de la grille de données sélectionnées.

Onglet Capacité utilisée
Le graphique **Récapitulatif en cours au cours des 30 seconds dernières minutes** affiche le nombre d'entrées de cache et de la capacité utilisée en octets au cours d'un intervalle de temps spécifié.

Onglet Utilisation du cache
L'onglet **Utilisation de cache** permet de visualiser le nombre de demandes ayant abouti dans le cache, et affiche les tentatives de cache, de réussites en mémoire cache, et le taux de réussite en mémoire cache au cours d'un intervalle de temps spécifié.

Onglet Débit moyen
Le graphique **Débit moyen vs. Durée moyenne de transaction** affiche le temps de transaction et le débit au cours d'une période donnée.

Informations une grille de données

Les statistiques de grille de données figurent dans la page **Surveiller** > **Détails de la grille de données**. Vous pouvez consulter les données pour une grille sélectionnée et les mappes qui se trouvent dans cette grille.

Récapitulatif en cours sur les 30 seconds dernières secondes
Affiche la capacité utilisée actuelle, le nombre d'entrées de cache, le temps de transaction moyen et le débit moyen de la grille de données sélectionnée.

Répartition des capacités actuelles utilisées par les mappes de grilles d'objets eXtreme Scale
Permet de visualiser un pool total, ce qui inclut les capacités par zone ainsi que les capacités totales dans chaque zone. Ne sont affichées que les 25

plus importantes mappes ObjectGrid. Vous pouvez également afficher les principaux consommateurs de capacité utilisés en fonction de chaque mappe.

Répartition de la capacité utilisée pour la zone actuelle
Affiche un pool total qui inclut le pool total et les principaux consommateurs de capacité utilisés dans la zone de la grille de données sélectionnée. Vous pouvez également afficher les principaux consommateurs de capacité utilisés en fonction de chaque zone.

Statistiques de mappe :

Récapitulatif en cours sur les 30 seconds dernières secondes
Affiche la capacité utilisée actuelle, le nombre d'entrées de cache, le temps de transaction moyen et le débit moyen de la grille de données sélectionnée.

Répartition de la capacité utilisée pour la partition actuelle
Vue d'une partition qui contient le pool total et les principaux consommateurs de capacité utilisés. Ne sont affichées que les 25 partitions les plus consommatrices. Vous pouvez également afficher les principaux consommateurs de capacité utilisés en fonction de chaque partition.

Présentation du serveur

Les statistiques du serveur figurent dans la page **Surveiller** > **Présentation du serveur**

Répartition actuelle de la mémoire utilisée du serveur
Ce graphique est composé de deux vues. **Pool total** affiche la quantité actuelle de mémoire utilisée (réelle) dans l'environnement d'exécution du serveur. **Plus grands consommateurs de mémoire utilisés** indique la mémoire utilisée par serveur, mais seuls les 25 premiers serveurs qui utilisent le plus de mémoire sont indiqués.

Total de la mémoire utilisée dans le temps
Affiche l'utilisation de la mémoire réelle dans l'environnement d'exécution du serveur.

Mémoire utilisée au fil du temps
Affiche la quantité de mémoire utilisée dans l'environnement d'exécution du serveur.

Rapports personnalisés : statistiques du domaine de services de catalogue

Vous pouvez afficher des statistiques de domaine de services de catalogue en créant un rapport personnalisé. Cliquez sur **Surveiller** > **Rapports personnalisés**.

Délai de transaction moyen (ms)
Affiche la durée moyenne que met une transaction à s'effectuer dans ce domaine.

Débit de transaction moyen (trans/s)
Affiche le nombre moyen de transactions par seconde dans ce domaine.

Délai maximum de transaction (ms)
Affiche le temps *maximum* qu'a mis une transaction pour s'exécuter dans ce domaine.

Délai minimum de transaction (ms)
Affiche le temps *minimum* qu'a mis une transaction pour s'exécuter dans ce domaine.

Délai de transaction total (ms)
Affiche le temps total passé à des transactions dans ce domaine depuis l'initialisation de ce dernier.

Rapports personnalisés : statistiques du serveur de conteneur

Vous pouvez afficher des statistiques de domaine de serveur de catalogue en créant un rapport personnalisé. Cliquez sur **Surveiller** > **Rapports personnalisés**.

Délai de transaction moyen (ms)
Affiche pour ce serveur de catalogue la durée moyenne que met une transaction à s'effectuer.

Débit de transaction moyen (trans/s)
Affiche le nombre moyen de transactions par seconde pour ce serveur de catalogue.

Délai maximum de transaction (ms)
Affiche pour ce serveur de catalogue le temps *maximum* qu'a mis une transaction pour s'exécuter.

Délai minimum de transaction (ms)
Affiche pour ce serveur de catalogue le temps *minimum* qu'a mis une transaction pour s'exécuter.

Délai de transaction total (ms)
Affiche pour ce serveur de catalogue le temps total passé à des transactions depuis l'initialisation du serveur.

Nombre total d'entrées en cache
Affiche le nombre actuel d'objets mis en cache appartenant à des grilles supervisées par ce serveur de catalogue.

Taux de réussites (pourcentage)
Affiche le taux de réussites pour la grille de données sélectionnée. Un taux élevé est souhaitable. Ce taux indique le degré d'efficacité de la grille pour éviter d'accéder au stockage de persistance.

Octets utilisés
Affiche la consommation de la mémoire par cette mappe. Les statistiques d'octets utilisés sont exactes uniquement lorsque vous utilisez des objets simples ou le mode de copie COPY_TO_BYTES.

Nombre minimal d'octets utilisés
Affiche le point bas de la consommation de mémoire par ce service de catalogue et ses mappes. Les statistiques d'octets utilisés sont exactes uniquement lorsque vous utilisez des objets simples ou le mode de copie COPY_TO_BYTES.

Nombre maximal d'octets utilisés
Affiche le point haut de la consommation de mémoire par ce service de catalogue et ses mappes. Les statistiques d'octets utilisés sont exactes uniquement lorsque vous utilisez des objets simples ou le mode de copie COPY_TO_BYTES.

Nombre total de réussites
Affiche le nombre total de fois où les données demandées ont été trouvées dans la mappe, dispensant de devoir accéder au stockage de persistance.

Nombre total de demandes get
Affiche le nombre total de fois où la mappe a dû accéder au stockage de persistance pour obtenir des données.

Segments de mémoire disponibles (Mo)
Affiche la quantité effective de segments mémoire disponibles pour la machine virtuelle Java en cours d'utilisation par le serveur de catalogue.

Total des segments de mémoire
Affiche la quantité effective de segments mémoire disponibles pour la machine virtuelle Java en cours d'utilisation par ce serveur de catalogue.

Nombre de processeurs disponibles
Affiche le nombre de processeurs qui sont disponibles pour ce service de catalogue et ses mappes. Pour une stabilité optimale, vous devez exécuter les serveurs à 60 % du chargement de processeur et les segments de mémoire des machines virtuelles Java à 60 % du chargement des segments de mémoire. Les pics peuvent alors pousser l'utilisation du processeur à 80-90 %, mais ce ne doit pas être le niveau habituel d'exécution de vos serveurs.

Taille maximale des segments de mémoire (Mo)
Affiche la quantité maximale de segments mémoire disponibles pour la machine virtuelle Java en cours d'utilisation par ce serveur de catalogue.

Mémoire utilisée
Affiche la mémoire utilisée dans la machine virtuelle Java en cours d'utilisation par ce serveur de catalogue.

Rapports personnalisés : statistiques de la grille de données

Vous pouvez afficher des statistiques de grille de données en créant un rapport personnalisé. Cliquez sur **Surveiller** > **Rapports personnalisés**.

Délai de transaction moyen (ms)
Affiche la durée moyenne que mettent pour s'effectuer des transactions impliquant cette grille.

Débit de transaction moyen (trans/s)
Affiche le nombre moyen de transactions effectuées par seconde par cette grille.

Délai maximum de transaction (ms)
Affiche le temps *maximum* qu'a mis une transaction effectuée par cette grille.

Délai minimum de transaction (ms)
Affiche le temps *minimum* qu'a mis une transaction effectuée par cette grille.

Délai de transaction total (ms)
Affiche le temps total de traitement des transactions pour cette grille.

Rapports personnalisés : statistiques de mappe

Vous pouvez afficher des statistiques en créant un rapport personnalisé. Cliquez sur **Surveiller** > **Rapports personnalisés**.

Nombre total d'entrées en cache
Affiche le nombre d'objets de la mappe actuellement mis en cache.

Taux de réussites (pourcentage)
Affiche le taux de réussites pour la mappe sélectionnée. Un taux élevé est souhaitable. Ce taux indique le degré d'efficacité de la mappe pour éviter d'accéder au stockage de persistance.

Octets utilisés
Affiche la consommation de la mémoire par cette mappe. Les statistiques d'octets utilisés sont exactes uniquement lorsque vous utilisez des objets simples ou le mode de copie COPY_TO_BYTES.

Nombre minimal d'octets utilisés
Affiche la consommation minimum en octets pour cette mappe. Les statistiques d'octets utilisés sont exactes uniquement lorsque vous utilisez des objets simples ou le mode de copie COPY_TO_BYTES.

Nombre maximal d'octets utilisés
Affiche la consommation maximum en octets pour cette mappe. Les statistiques d'octets utilisés sont exactes uniquement lorsque vous utilisez des objets simples ou le mode de copie COPY_TO_BYTES.

Nombre total de réussites
Affiche le nombre total de fois où les données demandées ont été trouvées dans la mappe, dispensant de devoir accéder au stockage de persistance.

Nombre total de demandes get
Affiche le nombre total de fois où la mappe a dû accéder au stockage de persistance pour obtenir des données.

Segments de mémoire disponibles (Mo)
Affiche la quantité effective de segments mémoire disponibles pour cette mappe dans la machine virtuelle Java en cours d'utilisation par le serveur de catalogue.

Total des segments de mémoire (Mo)
Affiche la quantité totale de segments mémoire disponibles pour cette mappe dans la machine virtuelle Java en cours d'utilisation par le serveur de catalogue. Pour une stabilité optimale, vous devez exécuter les serveurs à 60 % du chargement de processeur et les segments de mémoire des machines virtuelles Java à 60 % du chargement des segments de mémoire. Les pics peuvent alors pousser l'utilisation du processeur à 80-90 %, mais ce ne doit pas être le niveau habituel d'exécution de vos serveurs.

Nombre de processeurs disponibles
Affiche le nombre de processeurs disponibles pour cette mappe. Pour une stabilité optimale, vous devez exécuter les serveurs à 60 % du chargement de processeur et les segments de mémoire des machines virtuelles Java à 60 % du chargement des segments de mémoire. Les pics peuvent alors pousser l'utilisation du processeur à 80-90 %, mais ce ne doit pas être le niveau habituel d'exécution de vos serveurs.

Taille maximale des segments de mémoire (Mo)
Affiche la quantité maximum de segments mémoire disponibles pour cette mappe dans la machine virtuelle Java en cours d'utilisation par le serveur de catalogue.

Mémoire utilisée (Mo)
Affiche la quantité de mémoire utilisée dans cette mappe.

Surveillance à l'aide de rapports personnalisés

Vous pouvez générer des rapports personnalisés pour enregistrer les divers graphiques qui contiennent des statistiques sur les domaine de services de catalogue, les grilles de données et les serveurs de conteneur dans votre environnement. Vous pouvez enregistrer les rapports personnalisés et les charger pour les consulter ultérieurement.

Avant de commencer

Pour pouvoir afficher les statistiques avec la console Web, vous devez procéder comme suit :

1. Démarrez le serveur de console Web. Pour plus d'informations, voir «Démarrage et consignation sur la console Web», à la page 443.
2. Connectez vos serveurs de catalogue au serveur de la console Web. Pour plus d'informations, voir «Connexion de la console Web aux serveurs de catalogue», à la page 445.
3. Exécutez les grilles et les applications de données actives dans les serveurs qui sont gérés par votre domaine de services de catalogue

Procédure

- Créez un rapport personnalisé.
  1. Cliquez sur **Contrôler** > **Rapports personnalisés**. Les domaines eXtreme Scale que vous avez définis sont répertoriés dans un format d'arborescence. Vous pouvez développer chacun de ces domaines pour afficher les statistiques disponibles que vous pouvez ajouter au rapport personnalisé.
  2. Ajoutez des graphiques avec les statistiques à suivre. Les statistiques disponibles sont signalées par l'icône de graphique (). Cliquez sur l'une des statistiques à suivre. Choisissez **Ajouter au nouveau graphique** ou **Ajouter au graphique existant**. En fonction de votre sélection, la statistique sélectionnée s'affiche dans un nouvel onglet de graphique ou dans le graphique sélectionné. Vous pouvez ajouter une métrique à un graphique existant uniquement si les métriques déjà sur le graphique et la nouvelle métrique utilisent la même unité.
- Enregistrez un rapport personnalisé. La sauvegarde du rapport personnalisé enregistre les statistiques dans tous les onglets que vous avez créés. Pour sauvegarder le rapport, cliquez sur **Sauvegarder**.
- Chargez un rapport personnalisé. Cliquez sur **Charger** et choisissez le rapport personnalisé enregistré à afficher.

Surveillance à l'aide de fichiers CSV

Vous pouvez activer la surveillance des données à écrire dans des fichiers CSV. Ces fichiers CSV peuvent contenir des informations sur la machine JVM, la mappe, ou instance ObjectGrid.

Pourquoi et quand exécuter cette tâche

En activant la surveillance des données à écrire dans des fichiers CSV, vous pouvez télécharger et analyser les données historiques d'un serveur de conteneur ou de catalogue. Les données sont collectées lorsque vous démarrez le serveur avec les propriétés serveur qui activent les fichiers CSV. Vous pouvez ensuite télécharger les fichiers CSV à tout moment et utiliser les fichiers comme vous le désirez.

Procédure

1. Mettez à jour le fichier des propriétés du serveur avec les propriétés suivantes qui sont liées à l'activation des fichiers CSV.

```
parameter=default value
jvmStatsLoggingEnabled=true
maxJVMStatsFiles=5
maxJVMStatsFileSize=100
jvmStatsFileName=jvmstats
jvmStatsWriteRate=10

mapStatsLoggingEnabled=true
maxMapStatsFiles=5
maxMapStatsFileSize=100
mapStatsFileName=mapstats
mapStatsWriteRate=10

ogStatsLoggingEnabled=true
maxOGStatsFiles=5
maxOGStatsFileSize=100
ogStatsFileName=ogstats
ogStatsWriteRate=10
```

   Pour plus d'informations sur ces propriétés, voir Fichier de propriétés du serveur.
2. Redémarrez le serveur pour sélectionner les modifications dans le fichier des propriétés du serveur.
3. Téléchargez le fichier CSV. Le fichier CSV est écrit dans le répertoire *server_name*/logs.
4. Importez le fichier CSV dans le programme que vous utilisez pour traiter les données, par exemple, une feuille de calcul.

Que faire ensuite

Pour plus d'informations sur les données qui figurent dans les fichiers CSV, voir «Définition des statistiques des fichiers CSV».

Définition des statistiques des fichiers CSV

Les fichiers CSV que vous pouvez télécharger pour un serveur comprennent des statistiques que vous pouvez utiliser pour créer des diagrammes d'historique ou d'autres informations.

Journal des statistiques JVM (Java virtual machine)

TimeStamp
Indique la date et l'heure de l'image instantanée des statistiques, prise pour la machine JVM.

ServerName
Indique le nom du serveur de la machine JVM.

Nom d'hôte
Indique le nom de la machine JVM.

DomainName
Indique le domaine du service de catalogue auquel la machine JVM appartient.

FreeMemory
Indique le nombre d'octets disponibles pour la machine JVM.

MaxMemory
Indique le nombre maximal d'octets qui peut être attribué pour la machine JVM.

TotalMemory
Affiche l'utilisation de la mémoire réelle dans l'environnement d'exécution du serveur.

AvailProcs
Affiche le nombre de processeurs qui sont disponibles pour ce service de catalogue et ses mappes. Pour une stabilité optimale, vous devez exécuter les serveurs à 60 % du chargement de processeur et les segments de mémoire des machines virtuelles Java à 60 % du chargement des segments de mémoire. Les pics peuvent alors pousser l'utilisation du processeur à 80-90 %, mais ce ne doit pas être le niveau habituel d'exécution de vos serveurs.

Journal des statistiques de mappe

TimeStamp
Indique la date et l'heure de l'image instantanée des statistiques, prise pour la mappe.

MapName
Indique le nom de la mappe.

OgName
Indique le nom de la grille de données à laquelle appartient la mappe.

PartitionId
Indique l'ID de la partition.

MapSetName
Indique le groupe de mappes auquel appartient la mappe.

HitRate
Affiche le taux de réussites pour la mappe sélectionnée. Un taux élevé est souhaitable. Le taux de réussite indique la manière dont la grille de données contribue à éviter d'accéder au stockage de persistance.

Number
Indique le nombre d'échantillons de données collectés depuis le démarrage du serveur. Par exemple, la valeur 100 indique que l'entrée est le 100ème échantillon collecté depuis le démarrage du serveur.

TotalGetCount
Affiche le nombre total de fois où la mappe a dû accéder au stockage de persistance pour obtenir des données.

TotalHitCount
Affiche le nombre total de fois où les données demandées ont été trouvées dans la mappe, dispensant de devoir accéder au stockage de persistance.

StartTime
Indique l'heure à laquelle l'appel a commencé à partir de la dernière réinitialisation des compteurs. Les réinitialisations se produisent lorsque le serveur démarre ou redémarre.

LastCount
Indique la durée écoulée depuis le dernier échantillon de données.

LastTotalGetCount
Indique le nombre total actuel d'opérations d'extraction à partir de la mémoire cache moins le nombre d'opérations d'extraction dans la période précédente.

LastTotalHitCount
Indique le nombre total actuel d'opérations d'extraction à partir de la mémoire cache moins le nombre d'opérations d'extraction dans la période précédente.

UsedBytes
Affiche la consommation de la mémoire par cette mappe. Les statistiques d'octets utilisés sont exactes uniquement lorsque vous utilisez des objets simples ou le mode de copie COPY_TO_BYTES.

MinUsedBytes
Affiche le point bas de la consommation de mémoire par ce service de catalogue et ses mappes. Les statistiques d'octets utilisés sont exactes uniquement lorsque vous utilisez des objets simples ou le mode de copie COPY_TO_BYTES.

MaxUsedBytes
Affiche le point haut de la consommation de mémoire par ce service de catalogue et ses mappes. Les statistiques d'octets utilisés sont exactes uniquement lorsque vous utilisez des objets simples ou le mode de copie COPY_TO_BYTES.

LastUsedBytes
Indique la valeur UsedBytes en cours moins la valeur UsedBytes à partir de la période de collecte des statistiques précédentes.

SampleLen
Indique la durée, en millisecondes, de la période d'échantillonnage des données.

Journal de statistiques ObjectGrid

Nombre
Indique le nombre d'échantillons de données collectés depuis le démarrage du serveur. Par exemple, la valeur 100 indique que l'entrée est le 100ème échantillon collecté depuis le démarrage du serveur.

TimeStamp
Indique la date et l'heure de l'image instantanée des statistiques, prise pour la grille de données.

OgName
Indique le nom de la grille de données.

PartitionId
Indique l'ID de la partition.

Hostname
Indique le nom d'hôte.

DomainName
Indique le domaine du service de catalogue auquel la grille de données appartient.

MaxTime
Affiche pour ce serveur de catalogue le temps *maximum* qu'a mis une transaction pour s'exécuter.

MinTime
Affiche pour ce serveur de catalogue le temps *minimum* qu'a mis une transaction pour s'exécuter.

MeanTime
Indique le temps moyen passé sur une transaction.

TotalTime
Affiche pour ce serveur de catalogue le temps total passé à des transactions depuis l'initialisation du serveur.

AvgTransTime
Affiche pour ce serveur de catalogue la durée moyenne que met une transaction à s'effectuer.

AvgThroughPut
Affiche le nombre moyen de transactions par seconde pour ce serveur de catalogue.

SumOfSquares
Spécifie la somme des carrés pour le temps de transaction. Cette valeur mesure l'écart par rapport à la moyenne à un moment donné.

SampleLen
Indique la durée, en millisecondes, de la période d'échantillonnage des données.

LastCount
Indique la durée écoulée depuis le dernier échantillon de données.

LastTotalTime
Indique le temps total actuel moins le temps total précédent de l'échantillonnage de données.

StartTime
Indique l'heure à laquelle les statistiques ont commencé à être collectées depuis la dernière réinitialisation des données. Les données sont réinitialisées lorsque le serveur redémarre.

Surveillance à l'aide de l'API Statistics

L'API Statistics est l'interface directe avec l'arborescence interne des statistiques. Les statistiques sont désactivées par défaut, mais peuvent être activées en définissant une interface StatsSpec. Une interface StatsSpec définit la manière dont WebSphere eXtreme Scale doit surveiller les statistiques.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser l'API locale StatsAccessor pour interroger les données et accéder aux statistiques d'une instance ObjectGrid qui se trouve sur la même machine virtuelle Java (JVM) que le code en cours d'exécution. Pour plus d'informations sur les interfaces spécifiques, voir la documentation de l'API. Utilisez les étapes ci-après pour activer la surveillance de l'arborescence des statistiques interne.

Procédure

1. Extrayez l'objet StatsAccessor. L'interface StatsAccessor suit le modèle des singletons. Par conséquent, en dehors des problèmes liés au chargeur de classe, il doit exister une instance StatsAccessor pour chaque JVM. Cette classe sert d'interface principale pour toutes les opérations sur les statistiques locales. Le

code ci-après illustre l'extraction de la classe de l'accesseur. Appelez cette opération avant tout autre appel ObjectGrid.

```
public class LocalClient
{

   public static void main(String[] args) {

      // extrayez un descripteur de StatsAccessor
      StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

   }

}
```

2. Définissez l'interface StatsSpec de grille de données. Définissez cette JVM de sorte qu'elle ne collecte toutes les statistiques qu'au niveau d'ObjectGrid. Vous devez vérifier qu'une application active toutes les statistiques qui peuvent être requises avant de commencer des transactions. L'exemple ci-après définit l'interface StatsSpec à l'aide d'un champ de constante statique et d'une chaîne de spécification. L'utilisation d'un champ de constante statique est plus simple car le champ a déjà défini la spécification. Toutefois, en utilisant une chaîne de spécification, vous pouvez autoriser toutes les combinaisons de statistiques requises.

```
public static void main(String[] args) {

      // extrayez un descripteur de StatsAccessor
      StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

      // Définissez la spéc via la zone statique
      StatsSpec spec = new StatsSpec(StatsSpec.OG_ALL);
      accessor.setStatsSpec(spec);

      // Définissez la spécification via la chaîne de spécification
      StatsSpec spec = new StatsSpec("og.all=enabled");
      accessor.setStatsSpec(spec);

}
```

3. Envoyez des transactions à la grille pour force la collecte des données en vue de la surveillance. Pour collecter des données utiles pour les statistiques, vous devez envoyer des transactions à la grille de données. L'extrait de code suivant insère un enregistrement dans MapA, qui se trouve dans ObjectGridA. Les statistiques se trouvant au niveau d'ObjectGrid, toute mappe dans l'ObjectGrid renvoie les mêmes résultats.

```
public static void main(String[] args) {

      // extrayez un descripteur de StatsAccessor
      StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

      // Définissez la spéc via la zone statique
      StatsSpec spec = new StatsSpec(StatsSpec.OG_ALL);
      accessor.setStatsSpec(spec);

      ObjectGridManager manager =
    ObjectGridmanagerFactory.getObjectGridManager();
      ObjectGrid grid = manager.getObjectGrid("ObjectGridA");
      Session session = grid.getSession();
      Map map = session.getMap("MapA");

      // Effectuez une insertion
      session.begin();
      map.insert("SomeKey", "SomeValue");
      session.commit();
}
```

4. Interrogez un objet StatsFact à l'aide de l'API StatsAccessor. Tous les chemins d'accès aux statistiques sont associés à une interface StatsFact. L'interface StatsFact est une marque de réservation générique permettant d'organiser et d'inclure un objet StatsModule. Pour que vous puissiez accéder au véritable module de statistiques, l'objet StatsFact doit être extrait.

```
public static void main(String[] args)
{

      // extrayez un descripteur de StatsAccessor
      StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

      // Définissez la spéc via la zone statique
      StatsSpec spec = new StatsSpec(StatsSpec.OG_ALL);
      accessor.setStatsSpec(spec);

      ObjectGridManager manager =
    ObjectGridManagerFactory.getObjectGridManager();
      ObjectGrid grid = manager.getObjectGrid("ObjectGridA");
      Session session = grid.getSession();
      Map map = session.getMap("MapA");

      // Effectuez une insertion
      session.begin();
      map.insert("SomeKey", "SomeValue");
      session.commit();

      // Extrayez StatsFact

      StatsFact fact = accessor.getStatsFact(new String[] {"EmployeeGrid"},
    StatsModule.MODULE_TYPE_OBJECT_GRID);

}
```

5. Interagissez avec l'objet StatsModule. L'objet StatsModule est contenu dans l'interface StatsFact. Vous pouvez obtenir une référence au module à l'aide de l'interface StatsFact. L'interface StatsFact étant une interface générique, vous devez transtyper le module renvoyé dans le type StatsModule attendu. Cette tâche collectant des statistiques eXtreme Scale, l'objet StatsModule renvoyé est transtypé dans un type OGStatsModule. Une fois que le module est transtypé, vous avez accès à toutes les statistiques disponibles.

```
public static void main(String[] args) {

      // extrayez un descripteur de StatsAccessor
      StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

      // Définissez la spéc via la zone statique
      StatsSpec spec = new StatsSpec(StatsSpec.OG_ALL);
      accessor.setStatsSpec(spec);

      ObjectGridManager manager =
    ObjectGridmanagerFactory.getObjectGridManager();
      ObjectGrid grid = manager.getObjectGrid("ObjectGridA");
      Session session = grid.getSession();
      Map map = session.getMap("MapA");

      // Effectuez une insertion
      session.begin();
      map.insert("SomeKey", "SomeValue");
      session.commit();

      // Extrayez StatsFact
      StatsFact fact = accessor.getStatsFact(new String[] {"EmployeeGrid"},
    StatsModule.MODULE_TYPE_OBJECT_GRID);

      // Extrayez le module et l'heure
```

```
        OGStatsModule module = (OGStatsModule)fact.getStatsModule();
        ActiveTimeStatistic timeStat =
      module.getTransactionTime("Default", true);
        double time = timeStat.getMeanTime();

  }
```

Modules des statistiques

WebSphere eXtreme Scale utilise un modèle de statistiques interne pour suivre et filtrer les données. Toutes les vues de données se basent sur cette structure sous-jacente pour assembler des instantanés des statistiques. Vous pouvez extraire des informations des modules de statistiques à l'aide de plusieurs méthodes.

Présentation

Dans WebSphere eXtreme Scale, les statistiques sont suivies et stockées dans des composants StatsModules. Le modèle de statistiques contient plusieurs types de modules :

OGStatsModule
: Fournit des statistiques sur une instance ObjectGrid, notamment le temps de réponse des transactions.

MapStatsModule
: Fournit des statistiques sur une mappe unique, notamment le nombre d'entrées et le taux de réussite.

QueryStatsModule
: Fournit des statistiques sur les requêtes, notamment la création de plan et les temps d'exécution.

AgentStatsModule
: Fournit des statistiques sur les agents d'API DataGrid, notamment les temps de sérialisation et d'exécution.

HashIndexStatsModule
: Fournit des statistiques sur la requête HashIndex et les temps d'exécution de maintenance.

SessionStatsModule
: Fournit des statistiques sur le plug-in du gestionnaire de sessions HTTP.

Pour de plus amples informations sur les modules de statistiques, voir l'le package com.ibm.websphere.objectgrid.stats dans la documentation de l'API.

Statistiques dans un environnement local

Le modèle est structuré comme un arbre n-aire (une arborescence dont tous les nœuds sont au même degré) contenant tous les types de modules de statistiques répertoriés dans la liste précédente. Du fait de cette structure, tous les nœuds de l'arborescence sont représentés par l'interface StatsFact. L'interface StatsFact peut représenter un seul module ou un groupe de modules à des fins d'agrégation. Par exemple, si plusieurs nœuds terminaux de l'arborescence représentent des objets MapStatsModule spécifiques, le nœud StatsFact parent de ces nœuds contient les statistiques agrégés pour tous les modules enfants. Une fois l'objet StatsFact extrait, vous pouvez extraire le module de statistiques correspondant à l'aide de l'interface.

A l'instar d'une mappe d'arborescence, vous pouvez utiliser un chemin ou une clé correspondante pour extraire un objet StatsFact spécifique. Le chemin est une

valeur String[] qui contient tous les nœuds du chemin de l'objet demandé. Par exemple, supposons que vous avez créé un objet ObjectGrid appelé ObjectGridA, qui contient deux mappes : MapA et MapB. Le chemin du module de statistiques de MapA se présente comme suit : `[ObjectGridA, MapA]`. Le chemin des statistiques agrégées des deux mappes se présente comme suit : `[ObjectGridA]`.

Statistiques dans un environnement réparti

Dans un environnement réparti, les modules de statistiques sont extraits à l'aide d'un chemin différent. Un serveur pouvant contenir plusieurs partitions, l'arborescence de statistiques doit suivre la partition à laquelle chaque module appartient. Le chemin de recherche d'un objet StatsFact spécifique est donc différent. A l'aide de l'exemple précédent, en indiquant que les mappes se trouvent dans la partition 1, utilisez le chemin `[1, ObjectGridA, MapA]` afin d'extraire cet objet StatsFact pour MapA.

Surveillance avec l'utilitaire `xscmd`

L'utilitaire **xscmd** remplace l'exemple d'utilitaire **xsadmin** comme outil de surveillance et d'administration complètement pris en charge. Avec l'utilitaire **xscmd**, vous pouvez afficher des informations textuelles relatives à votre topologie WebSphere eXtreme Scale.

Avant de commencer

- Pour que l'utilitaire **xscmd** affiche des résultats, vous devez avoir créé votre topologie de grille. Les serveurs de catalogue et les serveurs de conteneur doivent être démarrés. Pour plus d'informations, voir «Démarrage et arrêt des serveurs sécurisés», à la page 395.
- Voir «Administration avec l'utilitaire **xscmd**», à la page 415 pour plus d'informations sur le démarrage de l'utilitaire **xscmd**.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser l'utilitaire **xscmd** pour afficher la structure et l'état actuels de la grille de données (par exemple, le contenu de la grille. Dans cet exemple, la structure de grille de données dans cette tâche est une grille de données simple *ObjectGridA* avec la mappe *MapA* qui appartient au groupe de mappes *MapSetA*. Cet exemple montre comment afficher tous les conteneurs dans une grille de données et afficher des mesures filtrées concernant la taille de la mappe *MapA*. Pour afficher toutes les options de la commande, exécutez l'utilitaire **xscmd** sans arguments ou avec l'option **-help**.

Procédure

Surveillez l'environnement avec l'utilitaire **xscmd**.

- Pour activer les statistiques pour tous les serveurs, exécutez la commande suivante :
  - UNIX `./xscmd.sh -c setStatsSpec -spec ALL=enabled -g ObjectGridA`
  - Windows `xscmd.bat -c setStatsSpec -spec ALL=enabled -g ObjectGridA`
- Pour afficher tous les serveurs de conteneur en ligne pour une grille de données, exécutez la commande suivante :
  - UNIX `./xscmd.sh -c showPlacement -g ObjectGridA -ms MapSetA`
  - Windows `xscmd.bat -c showPlacement -g ObjectGridA -ms MapSetA`

Toutes les informations sur les conteneurs s'affichent.

Avertissement : Pour obtenir ces informations lorsque le protocole TLS/SSL (Transport Layer Security/Secure Sockets Layer) est activé, vous devez démarrer les serveurs de catalogue et de conteneur avec le port de service JMX défini. Pour le définir, vous pouvez utiliser l'option **-JMXServicePort** sur le script **startOgServer** ou appeler la méthode setJMXServicePort sur l'interface ServerProperties.

- Pour afficher des informations sur les mappes de la grille de données ObjectGridA, exécutez la commande suivante :
  - UNIX `./xscmd.sh -c showMapSizes -g ObjectGridA -ms MapSetA`
  - Windows `xscmd.bat -c showMapSizes -g ObjectGridA -ms MapSetA`
- Pour vous connecter au service de catalogue et afficher des informations sur la mappe MapA pour l'ensemble du domaine de services de catalogue, exécutez la commande suivante :
  - UNIX `./xscmd.sh -c showMapSizes -g ObjectGridA -ms MapSetA -m MapA -cep CatalogMachine:6645`
  - Windows `xscmd.bat -c showMapSizes -g ObjectGridA -ms MapSetA -m MapA -cep CatalogMachine:6645`

  L'utilitaire **xscmd** se connecte au serveur MBean qui s'exécute dans un serveur de catalogue. En vous connectant à un seul serveur de catalogue, vous pouvez extraire des informations sur l'ensemble du domaine de services de catalogue. Un serveur de catalogue peut s'exécuter comme processus autonome, processus WebSphere Application Server ou être intégré dans un processus d'application personnalisé. Utilisez l'option **-cep** pour définir le nom d'hôte et le port du service de catalogue. Si vous incluez une liste de serveurs de catalogue pour l'option **-cep**, les serveurs de catalogue doivent se trouver dans le même domaine de services de catalogue. Vous pouvez extraire des statistiques pour un seul domaine de services de catalogue à la fois.
- Pour afficher le placement configuré et d'exécution de votre configuration, exécutez la commande suivante :
  - `xscmd -c placementServiceStatus`
  - `xscmd -c placementServiceStatus -g ObjectGridA -ms MapSetA`
  - `xscmd -c placementServiceStatus -ms MapSetA`
  - `xscmd -c placementServiceStatus -g ObjectGridA`

  Vous pouvez définir la portée de la commande pour afficher les informations de placement de l'intégralité de la configuration, une grille de données unique, un groupe de mappes unique ou une combinaison de grille de données et de groupe de mappes

Surveillance à l'aide de la fonction PMI de WebSphere Application Server

WebSphere eXtreme Scale prend en charge PMI (Performance Monitoring Infrastructure) lorsqu'il est exécuté dans un serveur d'applications WebSphere Application Server ou WebSphere Extended Deployment. PMI collecte des données de performances relatives aux applications exécutables et offre des interfaces permettant aux applications externes de surveiller les données de performances. Vous pouvez utiliser la console d'administration ou l'outil wsadmin pour accéder aux données de surveillance.

Avant de commencer

Vous pouvez utiliser PMI pour surveiller votre environnement lorsque vous utilisez WebSphere eXtreme Scale avec WebSphere Application Server.

Pourquoi et quand exécuter cette tâche

WebSphere eXtreme Scale utilise la fonction PMI personnalisée de WebSphere Application Server pour ajouter sa propre instrumentation de PMI. Avec cette approche, vous pouvez activer et désactiver la fonction PMI de WebSphere eXtreme Scale à l'aide de la console d'administration ou des interfaces JMX (Java Management Extensions) de l'outil wsadmin. En outre, vous pouvez accéder aux statistiques de WebSphere eXtreme Scale à l'aide des interfaces PMI et JMX standard utilisées par les outils de surveillance et notamment Tivoli Performance Viewer.

Procédure

1. Activez la fonction PMI de eXtreme Scale. Vous devez activer PMI pour afficher les statistiques PMI. Pour plus d'informations, voir «Activation de PMI».
2. Extrayez les statistiques PMI de eXtreme Scale. Affichez les performances de vos applications eXtreme Scale à l'aide de Tivoli Performance Viewer. Pour plus d'informations, voir «Récupération des statistiques PMI», à la page 466.

Que faire ensuite

Pour plus d'informations sur l'outil wsadmin, voir «Accès aux beans gérés (MBeans) à l'aide de l'outil wsadmin», à la page 434.

Activation de PMI

Vous pouvez utiliser l'infrastructure PMI (Performance Monitoring Infrastructure) de WebSphere Application Server pour activer ou désactiver les statistiques à tout niveau. Par exemple, vous pouvez choisir d'activer les statistiques du nombre d'occurrences d'une mappe donnée, mais non le nombre de statistiques en entrée ou les statistiques de durée de mise à jour par lots du chargeur. Vous pouvez activer PMI dans la console d'administration ou à l'aide de scripts.

Avant de commencer

Votre serveur d'applications doit être démarré et une application compatible eXtreme Scale doit y être installée. Pour activer PMI à l'aide de scripts, vous devez pouvoir vous connecter et utiliser l'outil wsadmin. Pour plus d'informations sur l'outil wsadmin, reportez-vous à la rubrique Outil wsadmin, dans le Centre de documentation de WebSphere Application Server.

Pourquoi et quand exécuter cette tâche

Utilisez l'infrastructure PMI de WebSphere Application Server pour fournir un mécanisme granulaire à l'aide duquel vous pouvez activer ou désactiver les statistiques à tout niveau. Par exemple, vous pouvez choisir d'activer les statistiques du nombre d'occurrences d'une mappe donnée, mais non le nombre d'entrées ou les statistiques de durée de mise à jour par lots du chargeur. Cette section montre comment utiliser la console d'administration et les scripts wsadmin pour activer l'infrastructure PMI d'ObjectGrid.

Procédure

- **Activez PMI dans la console d'administration.**
  1. Dans la console d'administration, cliquez sur **Contrôle et réglage** > **Performance Monitoring Infrastructure** > *nom_serveur*.
  2. Vérifiez que la case Activer l'infrastructure PMI (Performance Monitoring Infrastructure) est cochée. Ce paramètre est activé par défaut. S'il ne l'est pas, cochez la case et redémarrez le serveur.
  3. Cliquez sur **Personnalisé**. Dans l'arborescence de configuration, sélectionnez l'ObjectGrid et le module Mappes d'ObjectGrid. Activez les statistiques de chaque module.

  La catégorie des types de transaction des statistiques ObjectGrid est créée lors de la phase d'exécution. Vous ne pouvez voir que les sous-catégories des statistiques ObjectGrid et des statistiques de mappe dans la page **Exécution**.
- **Activez PMI à l'aide de scripts.**
  1. Ouvrez une invite de ligne de commande. Accédez au répertoire *racine_was*/bin. Entrez **wsadmin** pour démarrer l'outil de ligne de commande wsadmin.
  2. Modifiez la configuration de l'environnement d'exécution de l'infrastructure PMI d'eXtreme Scale. Vérifiez que PMI est activé pour le serveur à l'aide des commandes suivantes :

     ```
     wsadmin>set s1 [$AdminConfig getid /Cell:CELL_NAME/Node:NODE_NAME/
         Server:APPLICATION_SERVER_NAME/]
     wsadmin>set pmi [$AdminConfig list PMIService $s1]
     wsadmin>$AdminConfig show $pmi.
     ```

     Si PMI n'est pas activé, exécutez les commandes suivantes pour activer PMI :

     ```
     wsadmin>$AdminConfig modify $pmi {{enable true}}
     wsadmin>$AdminConfig save
     ```

     Si vous avez besoin d'activer PMI, redémarrez le serveur.
  3. Définissez des variables pour modifier l'ensemble de statistiques en ensemble personnalisé à l'aide des commandes suivantes :

     ```
     wsadmin>set perfName [$AdminControl completeObjectName type=Perf,
     process=APPLICATION_SERVER_NAME,*]
     wsadmin>set perfOName [$AdminControl makeObjectName $perfName]
     wsadmin>set params [java::new {java.lang.Object[]} 1]
     wsadmin>$params set 0 [java::new java.lang.String custom]
     wsadmin>set sigs [java::new {java.lang.String[]} 1]
     wsadmin>$sigs set 0 java.lang.String
     ```
  4. Spécifiez un ensemble de statistiques personnalisé à l'aide de la commande suivante :

     ```
     wsadmin>$AdminControl invoke_jmx $perfOName setStatisticSet $params $sigs
     ```
  5. Définissez des variables pour activer les statistiques de l'infrastructure PMI d'objectGridModule à l'aide des commandes suivantes :

     ```
     wsadmin>set params [java::new {java.lang.Object[]} 2]
     wsadmin>$params set 0 [java::new java.lang.String objectGridModule=1]
     wsadmin>$params set 1 [java::new java.lang.Boolean false]
     wsadmin>set sigs [java::new {java.lang.String[]} 2]
     wsadmin>$sigs set 0 java.lang.String
     wsadmin>$sigs set 1 java.lang.Boolean
     ```
  6. Définissez la chaîne des statistiques à l'aide de la commande suivante :

```
wsadmin>set params2 [java::new {java.lang.Object[]} 2]
wsadmin>$params2 set 0 [java::new java.lang.String mapModule=*]
wsadmin>$params2 set 1 [java::new java.lang.Boolean false]
wsadmin>set sigs2 [java::new {java.lang.String[]} 2]
wsadmin>$sigs2 set 0 java.lang.String
wsadmin>$sigs2 set 1 java.lang.Boolean
```

7. Définissez la chaîne des statistiques à l'aide de la commande suivante :

```
wsadmin>$AdminControl invoke_jmx $perfOName setCustomSetString $params2 $sigs2
```

Ces étapes activent l'infrastructure PMI de l'environnement d'exécution d'eXtreme Scale, mais ne modifient pas la configuration de l'infrastructure PMI. Si vous redémarrez l'application, les paramètres PMI sont perdus, exceptée l'activation principale de PMI.

Exemple

Vous pouvez effectuer les étapes suivantes pour activer les statistiques PMI de l'exemple d'application :

1. Lancez l'application à l'aide de l'adresse Web `http://hôte:port/ObjectGridSample`, hôte et port correspondant au nom d'hôte et au numéro de port HTTP du serveur où l'exemple est installé.
2. Dans l'exemple d'application, cliquez sur ObjectGridCreationServlet, puis sur les boutons d'action 1, 2, 3, 4 et 5 pour générer des actions sur l'ObjectGrid et les mappes. Ne fermez pas tout de suite cette page de servlet.
3. Dans la console d'administration, cliquez sur **Contrôle et réglage** > **Performance Monitoring Infrastructure** > *nom_serveur*. Cliquez sur l'onglet **Exécution**.
4. Cliquez sur le bouton d'option **Personnalisé**.
5. Développez le module Mappes d'ObjectGrid dans l'arborescence d'exécution, puis cliquez sur le lien clusterObjectGrid. Le groupe Mappes d'ObjectGrid contient une instance ObjectGrid appelée clusterObjectGrid et le groupe clusterObjectGrid contient quatre mappes : counters, employees, offices, et sites. Dans l'instance ObjectGrids se trouve une instance clusterObjectGrid et sous cette instance, le type de transaction DEFAULT.
6. Vous pouvez activer les statistiques de votre choix. Par exemple, vous pouvez activer le nombre d'entrées de mappe pour la mappe des employés et le temps de réponse des transactions pour le type de transaction DEFAULT.

Que faire ensuite

Une fois que PMI est activé, vous pouvez afficher les statistiques PMI à l'aide de la console d'administration ou de scripts.

Récupération des statistiques PMI

En récupérant les statistiques PMI, vous pouvez voir les performances de vos applications eXtreme Scale.

Avant de commencer

- Activez la fonction de suivi des statistiques PMI pour votre environnement. Pour plus d'informations, voir «Activation de PMI», à la page 464.
- Les chemins dans cette tâche partent du principe que vous récupérez les statistiques pour l'exemple d'application, mais vous pouvez utiliser ces statistiques pour toute autre application avec des étapes similaires.

- Si vous utilisez la console d'administration, vous devez être capable de vous y connecter. Si vous utilisez un script, vous devez être capable de vous connecter à wsadmin.

Pourquoi et quand exécuter cette tâche

Vous pouvez récupérer les statistiques PMI pour les afficher dans Tivoli Performance Viewer en suivant les étapes dans la console d'administration ou par script.

- Etapes de la console d'administration
- Etapes du script

Pour plus d'informations concernant les statistiques qui peuvent être récupérées, voir «Modules PMI», à la page 468.

Procédure

- Récupérez les statistiques PMI dans la console d'administration.
  1. Dans la console d'administration, cliquez sur **Contrôle et réglage** > **Performance viewer** > **Activité actuelle**
  2. Sélectionnez le serveur que vous voulez contrôler à l'aide de Tivoli Performance Viewer, puis activez le contrôle.
  3. Cliquez sur le serveur pour afficher la page Performance viewer.
  4. Développez l'arborescence de configuration. Cliquez sur **ObjectGrid Maps** > **clusterObjectGrid**, sélectionnez **employés**. Développez **ObjectGrids** > **clusterObjectGrid** et sélectionnez **DEFAULT**.
  5. Dans le modèle d'application ObjectGrid, accédez au servlet ObjectGridCreationServlet, cliquez sur le bouton 1 et remplissez les mappes. Vous pouvez afficher les statistiques dans l'afficheur.
- Récupérez les statistiques PMI avec un script.
  1. Dans une invite de ligne de commande, accédez au répertoire *racine_was*/bin. Entrez wsadmin pour lancer l'outil wsadmin.
  2. Définissez les variables pour l'environnement à l'aide des commandes suivantes :

     ```
     wsadmin>set perfName [$AdminControl completeObjectName type=Perf,*]
     wsadmin>set perfOName [$AdminControl makeObjectName $perfName]
     wsadmin>set mySrvName [$AdminControl completeObjectName type=Server,
      name=APPLICATION_SERVER_NAME,*]
     ```

  3. Définissez les variables pour obtenir les statistiques de mapModule à l'aide des commandes suivantes :

     ```
     wsadmin>set params [java::new {java.lang.Object[]} 3]
     wsadmin>$params set 0 [$AdminControl makeObjectName $mySrvName]
     wsadmin>$params set 1 [java::new java.lang.String mapModule]
     wsadmin>$params set 2 [java::new java.lang.Boolean true]
     wsadmin>set sigs [java::new {java.lang.String[]} 3]
     wsadmin>$sigs set 0 javax.management.ObjectName
     wsadmin>$sigs set 1 java.lang.String
     wsadmin>$sigs set 2 java.lang.Boolean
     ```

  4. Obtenez les statistiques de mapModule à l'aide de la commande suivante :

     ```
     wsadmin>$AdminControl invoke_jmx $perfOName getStatsString $params $sigs
     ```

  5. Définissez les variables pour obtenir les statistiques d'objectGridModule à l'aide des commandes suivantes :

     ```
     wsadmin>set params2 [java::new {java.lang.Object[]} 3]
     wsadmin>$params2 set 0 [$AdminControl makeObjectName $mySrvName]
     wsadmin>$params2 set 1 [java::new java.lang.String objectGridModule]
     ```

```
wsadmin>$params2 set 2 [java::new java.lang.Boolean true]
wsadmin>set sigs2 [java::new {java.lang.String[]} 3]
wsadmin>$sigs2 set 0 javax.management.ObjectName
wsadmin>$sigs2 set 1 java.lang.String
wsadmin>$sigs2 set 2 java.lang.Boolean
```

6. Obtenez les statistiques d'objectGridModule à l'aide de la commande suivante :

```
wsadmin>$AdminControl invoke_jmx $perfOName getStatsString $params2 $sigs2
```

Résultats

Vous pouvez afficher les statistiques dans Tivoli Performance Viewer.

Modules PMI

Vous pouvez surveiller les performances de vos applications avec les modules PMI (Performance Monitoring Infrastructure).

objectGridModule

Le module objectGridModule contient une statistique de durée : le temps de réponse des transactions. Une transaction est définie comme la durée entre l'appel de méthode Session.begin et l'appel de méthode Session.commit. Cette durée est suivie comme temps de réponse des transactions. L'élément racine de la structure objectGridModule, "root", sert de point d'entrée aux statistiques de WebSphere eXtreme Scale. Cet élément racine contient des ObjectGrids comme éléments enfant et ces derniers possèdent des types de transaction comme éléments enfant. Les statistiques de temps de réponse sont associées à chaque type de transaction.

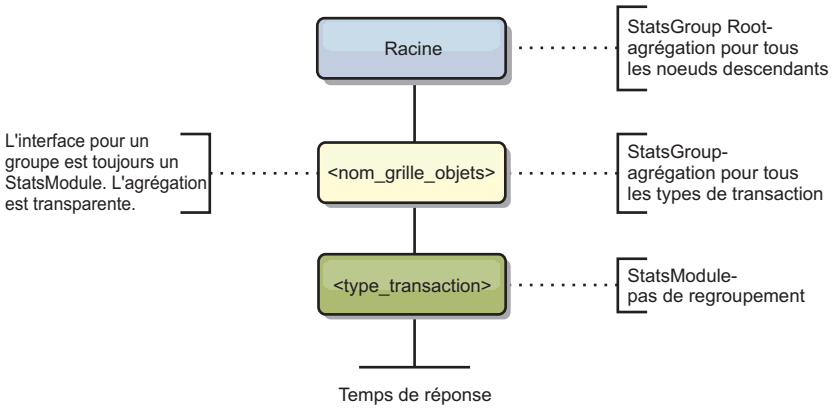


Figure 57. Structure de module ObjectGridModule

Le diagramme ci-après illustre un exemple de structure ObjectGridModule. Dans cet exemple, il existe deux instances ObjectGrid sur le système : ObjectGrid A et ObjectGrid B. L'instance ObjectGrid A possède deux types de transaction : A et default. L'instance ObjectGrid B ne possède que le type de transaction default.

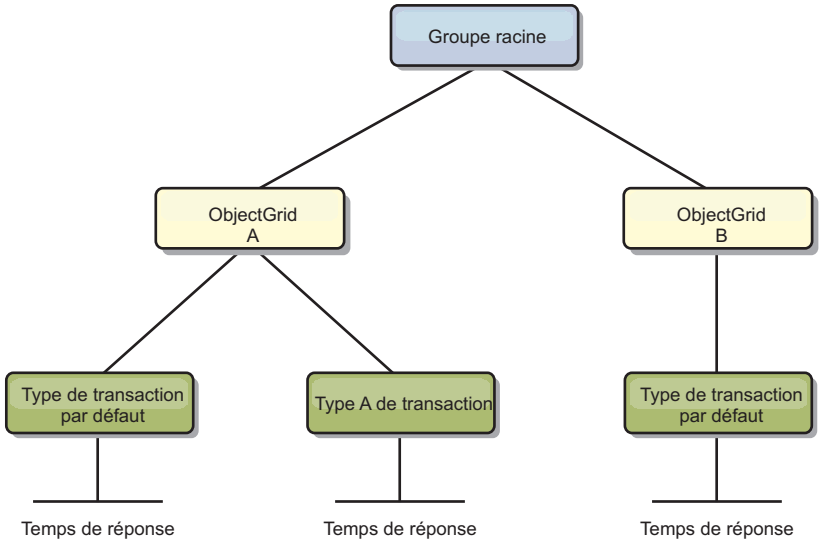


Figure 58. Exemple de structure de module ObjectGridModule

Les types de transaction sont définis par les développeurs d'applications car ils savent quels types de transaction sont utilisés par leurs applications. Le type de transaction est défini à l'aide de la méthode Session.setTransactionType(String) suivante :

```
/**
* Définit le type de transaction des transactions futures.
*
* Une fois que cette méthode a été appelée, toutes les transactions futures
* sont de même type jusqu'à ce qu'un autre type de transaction ait été
* défini. Si aucun type de transaction n'est défini, le type de transaction
* TRANSACTION_TYPE_DEFAULT par défaut est utilisé.
*
* Les types de transaction sont principalement utilisés à des fins de suivi
* des données statistiques.
* Les utilisateurs peuvent prédéfinir les types des transactions qui
* sont exécutées
* dans une application. L'idée consiste à regrouper les transactions de
* mêmes caractéristiques
* dans une même catégorie (type), afin qu'une statistique de temps de réponse
* des transactions puisse être utilisée pour rechercher chaque type de transaction.
*
* Ce suivi est utile si votre application possède différents types de
* transaction.
* Parmi eux, certains types de transaction, comme les transactions de mise à
* jour, possèdent un délai de traitement supérieur à celui d'autres
* transactions, telles que les transactions en lecture seule. Si le type de
* transaction est utilisé, les différentes transactions sont recherchées
* par des statistiques différentes, afin que ces dernières puissent
* être plus utiles.
* @param tranType Type de transaction des transactions futures.
*/
void setTransactionType(String tranType);
```

L'exemple suivant spécifie `updatePrice` comme type de transaction :

```
// Spécifiez le type de transaction updatePrice
// La durée entre session.begin() et session.commit() fait l'objet d'un suivi
// dans les statistiques de durée de "updatePrice".
session.setTransactionType("updatePrice");
session.begin();
map.update(stockId, new Integer(100));
session.commit();
```

La première ligne indique que le type de transaction suivant est updatePrice. Il existe une statistiques updatePrice sous l'instance ObjectGrid qui correspond à la session de l'exemple. A l'aide d'interfaces JMX (Java Management Extensions), vous pouvez obtenir le temps de réponse des transactions updatePrice. Vous pouvez également extraire les statistiques agrégées de tous les types de transaction sur l'instance ObjectGrid spécifiée.

mapModule

La structure mapModule contient trois statistiques sur les mappes eXtreme Scale :

- **Nombre d'occurrences de mappe** - *BoundedRangeStatistic* : Recherche le nombre d'occurrences d'une mappe. Le nombre d'occurrences est une valeur flottante comprise entre 0 et 100 compris, qui représente le pourcentage d'occurrences de mappe en relation avec les opérations d'extraction de mappe.
- **Nombre d'entrées** - *CountStatistic* : Recherche le nombre d'entrées dans la mappe.
- **Temps de réponse de la mise à jour par lots du chargeur** - *TimeStatistic* : Recherche le temps de réponse utilisé pour l'opération de mise à jour par lots du chargeur.

L'élément racine de la structure mapModule, "root", sert de point d'entrée aux statistiques des mappes ObjectGrid. Cet élément racine contient des ObjectGrids comme éléments enfant et ces derniers possèdent des mappes comme éléments enfant. Trois statistiques sont répertoriées pour chaque instance de mappe. La structure mapModule est illustrée dans le diagramme suivant :

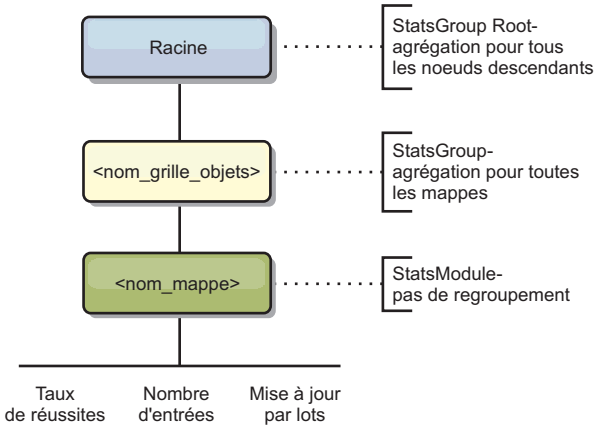
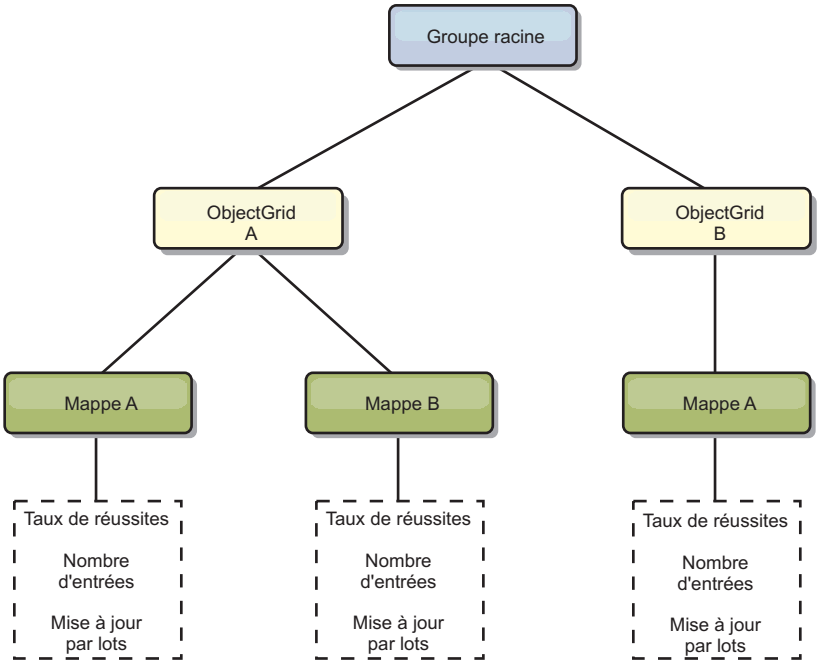


Figure 59. structure mapModule

Le diagramme suivant illustre un exemple de structure mapModule :

Figure 60. Exemple de structure de module mapModule



hashIndexModule

La structure hashIndexModule contient les statistiques suivantes sur les index de niveau mappe :

- **Nombre de recherches** - *CountStatistic* : Nombre d'appels de l'opération de recherche d'index.
- **Nombre de collisions** - *CountStatistic* : Nombre de collisions de l'opération de recherche.
- **Nombre d'échecs** - *CountStatistic* : Nombre d'échecs pour l'opération de recherche.
- **Nombre de résultats** - *CountStatistic* : Nombre de clés renvoyées par l'opération de recherche.
- **Nombre de mises à jour par lots** - *CountStatistic* : Nombre de mises à jour par lots sur cet index. Si la mappe correspondante es modifié qu'une quelconque manière, la méthode doBatchUpdate() de l'index est appelée. Cette statistiques indique la fréquence à laquelle votre index est modifié ou mis à jour.
- **Durée de recherche** - *TimeStatistic* : Temps que prend l'opération de recherche pour s'exécuter

L'élément racine de la structure hashIndexModule, "root", sert de point d'entrée aux statistiques de HashIndex. Cet élément racine contient des ObjectGrids comme éléments enfant, les ObjectGrids contiennent des mappes comme éléments enfant et enfin, ces mappes contiennent des instances HashIndex comme éléments enfant et les noeuds terminaux de l'arborescence. Trois statistiques sont répertoriées pour chaque instance HashIndex. La structure hashIndexModule est illustrée dans le diagramme suivant :

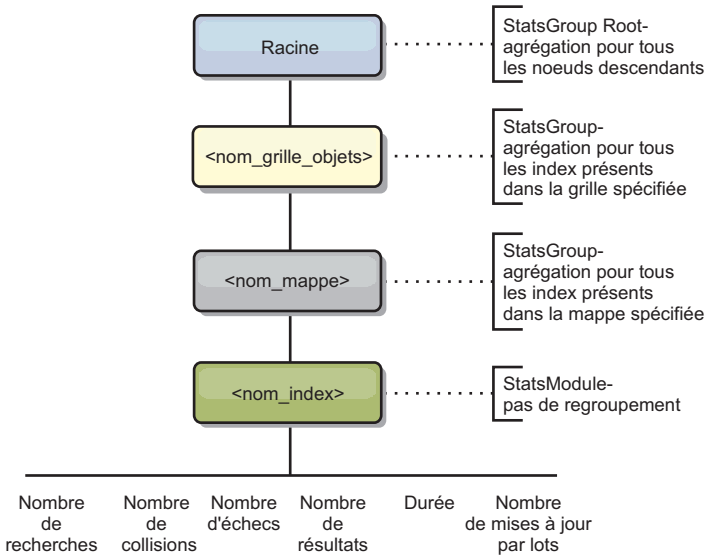


Figure 61. structure de module hashIndexModule

Le diagramme suivant illustre un exemple de structure hashIndexModule :

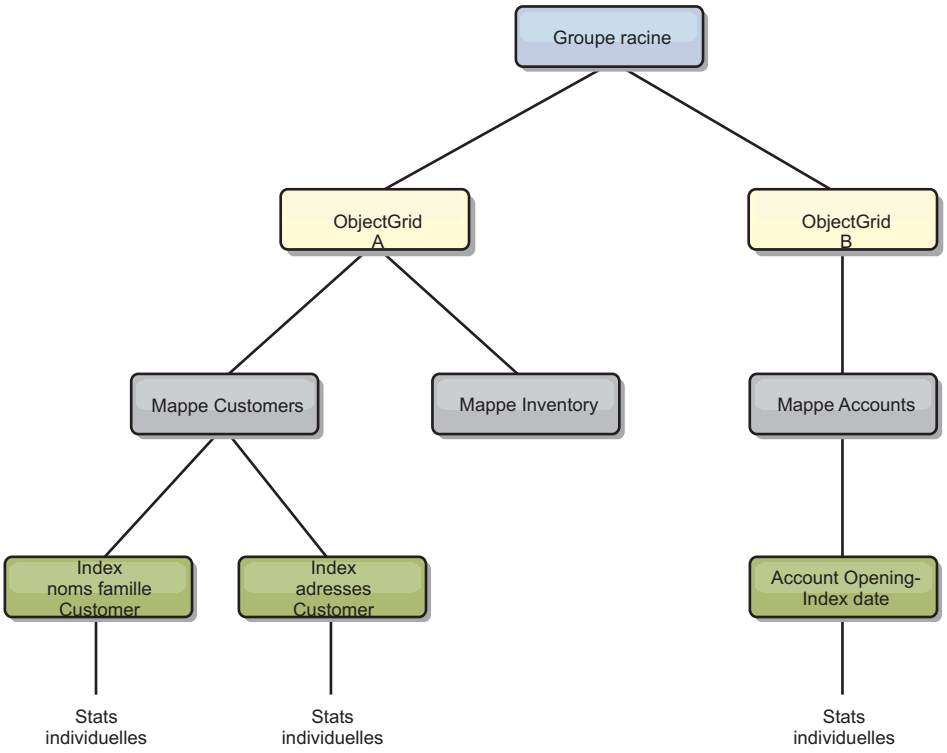


Figure 62. Exemple de structure de module hashIndexModule

agentManagerModule

La structure agentManagerModule contient les statistiques sur les agents de niveau mappe :

- **Temps de réduction** : *TimeStatistic* - Durée nécessaire pour que l'agent termine l'opération de réduction.

- **Durée totale** : *TimeStatistic* - Durée totale nécessaire à l'agent pour effectuer toutes les opérations.
- **Temps de sérialisation de l'agent** : *TimeStatistic* - Durée nécessaire pour sérialiser l'agent.
- **Temps d'inflation de l'agent** : *TimeStatistic* - Durée nécessaire pour l'inflation de l'agent sur le serveur.
- **Temps de sérialisation des résultats** : *TimeStatistic* - Durée nécessaire pour sérialiser les résultats de l'agent.
- **Temps d'inflation des résultats** : *TimeStatistic* - Durée nécessaire pour l'inflation des résultats de l'agent.
- **Nombre d'échecs** : *CountStatistic* - Nombre de fois que l'agent a échoué.
- **Nombre d'appels** : *CountStatistic* - Nombre d'appels d'AgentManager.
- **Nombre de partitions** : *CountStatistic* - Nombre de partitions vers lesquelles l'agent est envoyé.

L'élément racine de la structure agentManagerModule, "root", sert de point d'entrée aux statistiques d'AgentManager. Cet élément racine contient des ObjectGrids comme éléments enfant, les ObjectGrids contiennent des mappes comme éléments enfant et enfin, ces mappes contiennent des instances AgentManager comme éléments enfant et les noeuds terminaux de l'arborescence. Chaque instance AgentManager a des statistiques.

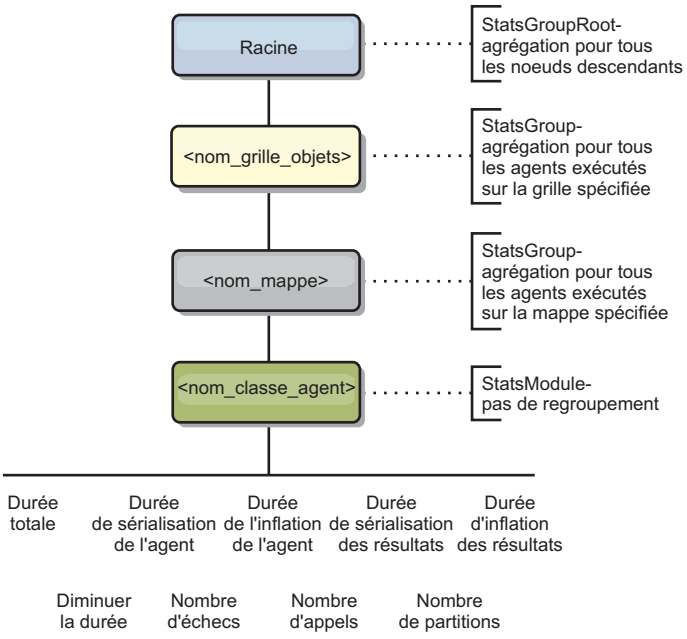


Figure 63. Structure agentManagerModule

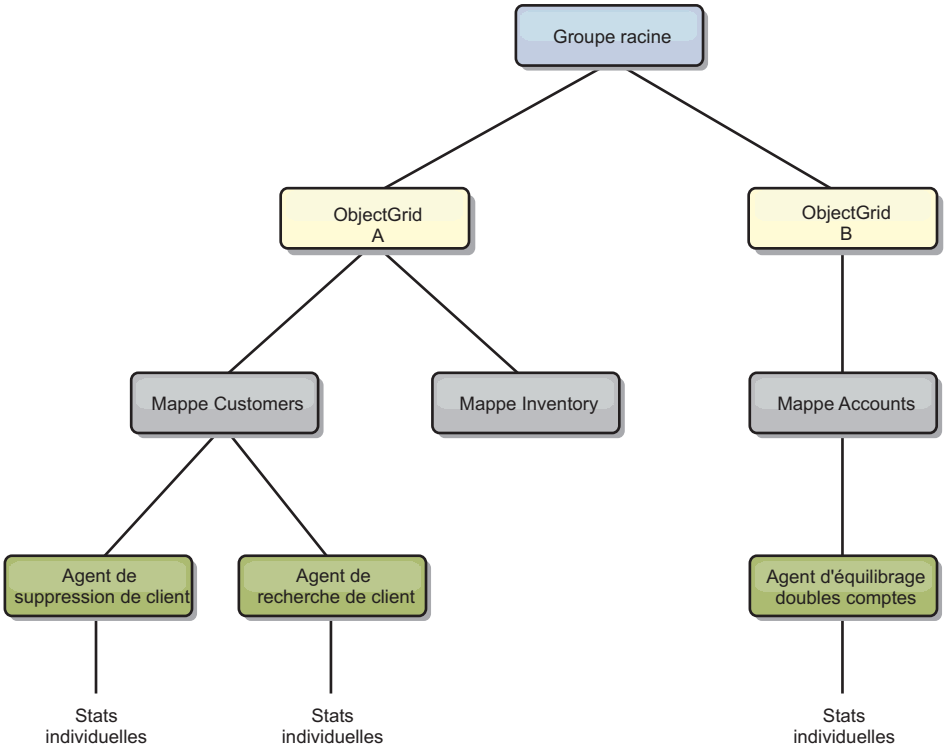


Figure 64. Exemple de structure agentManagerModule

queryModule

La structure queryModule contient les statistiques sur les requêtes eXtreme Scale :

- **Temps de création du plan** : *TimeStatistic* - Durée nécessaire pour créer le plan de requête.
- **Temps d'exécution** : *TimeStatistic* - Durée nécessaire pour exécuter la requête.
- **Nombre d'exécutions** : *CountStatistic* - Nombre de fois que la requête a été exécutée.
- **Nombre de résultats** : *CountStatistic* - Nombre de résultats pour chaque ensemble de résultats de chaque exécution de requête.
- **Nombre d'échecs** : *CountStatistic* - Nombre de fois que la requête a échoué.

L'élément racine de la structure queryModule, "root", sert de point d'entrée aux statistiques des requêtes. Cet élément racine contient des ObjectGrids comme éléments enfant et ces derniers possèdent des objets de requête comme éléments enfant et les noeuds terminaux de l'arborescence. Trois statistiques sont répertoriées pour chaque instance de requête.

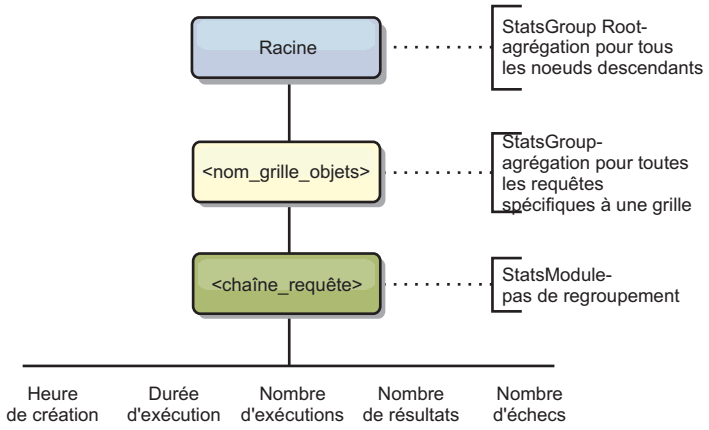


Figure 65. structure queryModule

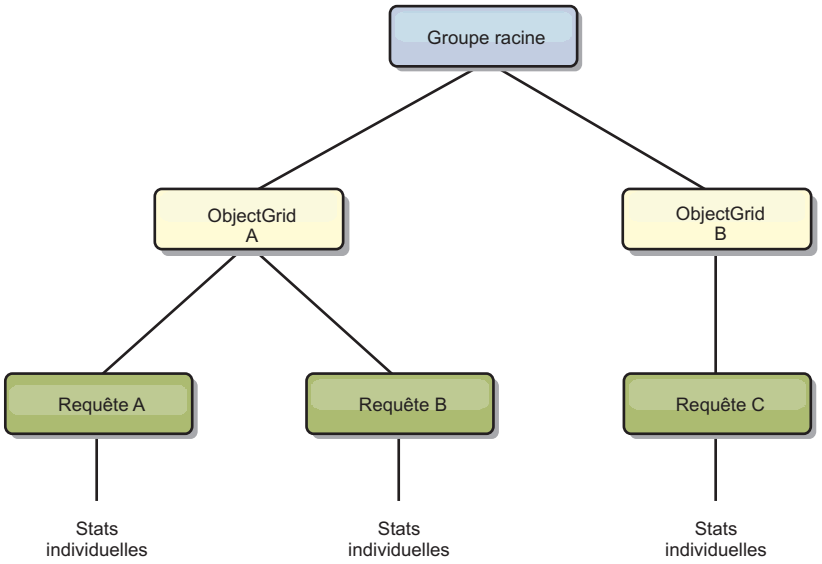


Figure 66. Exemple de structure queryModule QueryStats.jpg

Accès aux beans gérés (MBeans) à l'aide de l'outil wsadmin

Vous pouvez utiliser l'utilitaire wsadmin fourni dans WebSphere Application Server pour accéder aux informations des beans gérés (MBean).

Procédure

Exécutez l'outil wsadmin depuis le répertoire `bin` dans votre installation WebSphere Application Server. L'exemple suivant restaure une vue de la position actuelle du fragment dans un logiciel eXtreme Scale dynamique. Vous pouvez exécuter l'outil wsadmin depuis n'importe quelle installation où eXtreme Scale est en cours d'exécution. Vous n'avez pas besoin d'exécuter l'outil wsadmin sur le service de catalogue.

```
$ wsadmin.sh -lang jython
wsadmin>placementService = AdminControl.queryNames
 ("com.ibm.websphere.objectgrid:*,type=PlacementService")
wsadmin>print AdminControl.invoke(placementService,
 "listObjectGridPlacement","library ms1")
```

```
<objectGrid name="library" mapSetName="ms1">
  <container name="container-0" zoneName="DefaultDomain"
  hostName="host1.company.org" serverName="server1">
     <shard type="Primary" partitionName="0"/>
     <shard type="SynchronousReplica" partitionName="1"/>
  </container>
  <container name="container-1" zoneName="DefaultDomain"
  hostName="host2.company.org" serverName="server2">
     <shard type="SynchronousReplica" partitionName="0"/>
     <shard type="Primary" partitionName="1"/>
  </container>
  <container name="UNASSIGNED" zoneName="_ibm_SYSTEM"
  hostName="UNASSIGNED" serverName="UNNAMED">
    <shard type="SynchronousReplica" partitionName="0"/>
    <shard type="AsynchronousReplica" partitionName="0"/>
  </container>
</objectGrid>
```

Surveillance à l'aide de beans gérés (MBeans)

Vous pouvez utiliser des beans gérés (MBeans) pour effectuer le suivi des statistiques dans votre environnement.

Avant de commencer

Pour que les attributs puissent être enregistrés, vous devez activer les statistiques. Vous pouvez activer les statistiques de l'une des manières suivantes :

- **A l'aide du fichier de propriétés du serveur :**

  Vous pouvez activer les statistiques dans le fichier de propriétés du serveur avec l'entrée clé-valeur `statsSpec=<SpécStats>`. Voici quelques exemples de paramètres possibles :

  – pour activer toutes les statistiques, utilisez `statsSpec=all=enabled`

  – pour n'activer que les statistiques d'ObjectGrid, utilisez `statsSpec=og.all=enabled` Pour une description de toutes les spécifications de statistiques possibles, voir l'API StatsSpec dans la documentation de l'API.

  Pour plus d'informations sur le fichier de propriétés du serveur, voir Fichier de propriétés du serveur.

- **Avec un bean géré :**

  Vous pouvez activer les statistiques à l'aide de l'attribut StatsSpec dans le bean géré ObjectGrid. Pour plus d'informations, voir l'API StatsSpec dans la documentation d'API.

- **Par programmation :**

  Vous pouvez également programmer l'activation des statistiques avec l'interface StatsAccessor, qui est extraite avec la classe StatsAccessorFactory. Utilisez cette interface dans un environnement client ou lorsque vous devez surveiller une grille de données qui s'exécute dans le processus en cours.

Procédure

- **Accédez aux statistiques de bean géré en utilisant l'outil wsadmin.**

  Pour plus d'informations, voir «Accès aux beans gérés (MBeans) à l'aide de l'outil wsadmin», à la page 434.

- **Accédez aux statistiques de bean géré à l'aide d'un programme.**

  Pour plus d'informations, voir «Accès aux beans gérés (MBeans) à l'aide d'un programme», à la page 435.

Exemple

Pour obtenir un exemple d'utilisation des beans gérés, voir Exemple : utilitaire **xsadmin**.

Surveillance à l'aide d'outils fournis par une tierce partie

WebSphere eXtreme Scale peut être surveillé à l'aide de plusieurs solutions de surveillance d'entreprise couramment utilisées. Des agents de plug-in sont intégrés pour IBM Tivoli Monitoring et Hyperic HQ dont le rôle consiste à surveiller WebSphere eXtreme Scale à l'aide de beans de gestion accessibles publiquement. CA Wily Introscope utilise l'instrumentation de méthode Java pour capturer les statistiques.

Surveillance à l'aide d'IBM Tivoli Enterprise Monitoring Agent for WebSphere eXtreme Scale

L'agent IBM Tivoli Enterprise Monitoring est une solution de surveillance riche en fonctions que vous pouvez utiliser pour surveiller les bases de données, les systèmes d'exploitation et les serveurs dans des environnements hôte et répartis. WebSphere eXtreme Scale inclut un agent personnalisé que vous pouvez utiliser pour introspecter les beans de gestion d'eXtreme Scale. Cette solution fonctionne correctement pour les déploiements eXtreme Scale autonomes et les déploiements WebSphere Application Server.

Avant de commencer

- Installez WebSphere eXtreme Scale Version 7.0.0 ou ultérieure.

  Par ailleurs, les statistiques doivent être activées pour permettre la collecte de données statistiques à partir des serveurs WebSphere eXtreme Scale. Les diverses options d'activation des statistiques sont décrites dans «Surveillance à l'aide de beans gérés (MBeans)», à la page 476 et dans Exemple : utilitaire **xsadmin**.
- Installez IBM Tivoli Version 6.2.1 avec le Fix Pack 2 ou ultérieur.
- Installez l'agent du système d'exploitation Tivoli sur chaque serveur ou hôte sur lequel des serveurs eXtreme Scale sont exécutés.
- Installez l'agent WebSphere eXtreme Scale, que vous pouvez télécharger gratuitement à partir du site IBM Open Process Automation Library (OPAL).

Effectuez les étapes suivantes pour installer et configurer Tivoli Monitoring Agent :

Procédure

1. Installez Tivoli Monitoring Agent for WebSphere eXtreme Scale.

   Téléchargez l'image d'installation de Tivoli et extrayez ses fichiers dans un répertoire temporaire.
2. Installez les fichiers du support d'application d'eXtreme Scale.

   Installez le support d'application d'eXtreme Scale sur chacun des déploiements ci-après.
   - Tivoli Enterprise Portal Server (TEPS)
   - Client Enterprise Desktop (TEPD)
   - Tivoli Enterprise Monitoring Server (TEMS)

   a. Dans le répertoire temporaire que vous avez créé, démarrez une nouvelle fenêtre de commande et exécutez le fichier exécutable approprié pour votre plateforme. Le script d'installation détecte automatiquement votre type de

déploiement Tivoli (TEMS, TEPD ou TEPS). Vous pouvez installer tout type de déploiement sur un ou plusieurs hôtes ; ces trois types de déploiement requièrent l'installation des fichiers du support d'application de l'agent eXtreme Scale.

b. Dans la fenêtre **Programme d'installation**, vérifiez que la sélection des composants Tivoli déployés est correcte. Cliquez sur **Suivant**.

c. Si vous y êtes invité, soumettez votre nom d'hôte et vos données d'identification administratives. Cliquez sur **Suivant**.

d. Sélectionnez **Monitoring Agent for WebSphere eXtreme Scale**. Cliquez sur **Suivant**.

e. Le système vous indique les actions d'installation à effectuer. Cliquez sur **Suivant** ; vous pouvez voir la progression de l'installation jusqu'à sa fin.

Une fois que vous avez terminé cette procédure, tous les fichiers du support d'application requis par l'agent WebSphere eXtreme Scale sont installés.

3. Installez l'agent sur chacun des noeuds eXtreme Scale.

   Vous installez un agent de système d'exploitation Tivoli sur chacun des ordinateurs. Vous n'avez pas besoin de configurer ou de démarrer cet agent. Utilisez l'image d'installation de l'étape précédente pour exécuter le fichier exécutable spécifique à la plateforme.

   Vous n'avez besoin d'installer qu'un seul agent par hôte. Chaque agent peut prendre en charge plusieurs instances de serveur eXtreme Scale. Pour de meilleures performances, utilisez une instance d'agent pour surveiller environ 50 serveurs eXtreme Scale.

   a. Dans l'écran de bienvenue de l'assistant d'installation, cliquez sur **Suivant** pour ouvrir l'écran et spécifier les informations sur le chemin d'installation.

   b. Dans la zone **Répertoire d'installation d'IBM Tivoli Monitoring**, entrez ou recherchez `C:\IBM\ITM` (ou `/opt/IBM/ITM`). Ensuite, dans la zone de l'**emplacement du support installable**, vérifiez que la valeur affichée est correcte et cliquez sur **Suivant**.

   c. Sélectionnez les composants à ajouter, tels que **Effectuer une installation locale de la solution**, et cliquez sur **Suivant**.

   d. Sélectionnez les applications pour lesquelles vous souhaitez ajouter le support en les sélectionnant (par exemple, **Monitoring Agent for WebSphere eXtreme Scale**), puis en cliquant sur **Suivant**.

   e. La progression s'affiche jusqu'à ce que la prise en charge de l'application ait été ajoutée.

   Remarque : Répétez ces étapes sur chacun des noeuds eXtreme Scale. Vous pouvez également utiliser une installation en mode silencieux. Pour plus d'informations sur l'installation en mode silencieux, voir le Centre de documentation d'IBM Tivoli Monitoring.

4. Configurez l'agent WebSphere eXtreme Scale.

   Chacun des agents installés doit être configuré pour surveiller un serveur de catalogues et/ou un serveur eXtreme Scale.

   Les étapes de configuration des plateformes Windows et UNIX sont différentes. La configuration de la plateforme Windows s'exécute avec l'interface utilisateur **Manage Tivoli Monitoring Services**. La configuration des plateformes UNIX s'effectue depuis la ligne de commande.

   Windows Procédez comme suit pour configurer initialement l'agent sous Windows.

a. Dans la fenêtre **Gérer les services Tivoli Enterprise Monitoring**, cliquez sur **Démarrer** > **Tous les programmes** > **IBM Tivoli Monitoring** > **Gérer les services Tivoli Monitoring**.
b. Cliquez à l'aide du bouton droit de la souris sur **Monitoring Agent for WebSphere eXtreme Scale** et sélectionnez **Configure using default**, qui ouvre une fenêtre permettant de créer une instance unique de l'agent.
c. Choisissez un nom unique (par exemple, `instance1` et cliquez sur **Suivant**).

- Si vous prévoyez de surveiller des serveurs eXtreme Scale autonomes, effectuez les étapes suivantes :
  a. Mettez à jour les paramètres Java et assurez-vous que la valeur **Java Home** est correcte. Les arguments JVM peuvent rester vides. Cliquez sur **Suivant**.
  b. Sélectionnez le type **Type de connexion de serveur MBean** et utilisez `Serveur conforme à JSR-160` pour les serveurs eXtreme Scale autonomes. Cliquez sur **Suivant**.
  c. Si la sécurité est activée, mettez à jour les valeurs **ID utilisateur** et **Mot de passe**. Ne modifiez pas la valeur d'**URL de service JMX**. Vous remplacerez cette valeur ultérieurement. Ne modifiez pas la zone **Informations de chemin de classes JMX**. Cliquez sur **Suivant**.

  Pour configurer les serveurs pour l'agent sous Windows, procédez comme suit :
  a. Configurez des instances de sous-noeud des serveurs eXtreme Scale dans la sous-fenêtre **Serveurs de grille WebSphere eXtreme Scale**. S'il n'existe pas de serveurs de catalogues sur votre ordinateur, cliquez sur **Suivant** pour passer à la sous-fenêtre du service de catalogue.
  b. S'il existe plusieurs serveurs conteneurs eXtreme Scale sur votre ordinateur, configurez l'agent pour qu'il surveille chacun d'eux.
  c. Vous pouvez ajouter autant de serveurs eXtreme Scale que nécessaire, si leurs noms et ports sont uniques, en cliquant sur **Nouveau**. (Si un serveur eXtreme Scale est démarré, une valeur JMXPort doit être spécifiée.)
  d. Une fois que vous avez configuré les serveurs conteneurs, cliquez sur **Suivant** pour accéder à la sous-fenêtre **Serveurs de catalogues WebSphere eXtreme Scale**.
  e. En l'absence de serveurs de catalogues, cliquez sur **OK**. Si vous possédez des serveurs de catalogues, ajoutez une nouvelle configuration pour chaque serveur, comme pour les serveurs conteneurs. Choisissez de nouveau un nom unique, de préférence, celui utilisé au démarrage du service de catalogue. Cliquez sur **OK** pour terminer.
- Si vous prévoyez de surveiller les serveurs de l'agent sur des serveurs eXtreme Scale imbriqués dans un processus WebSphere Application Server, procédez comme suit :
  a. Mettez à jour les paramètres Java et assurez-vous que la valeur **Java Home** est correcte. Les arguments JVM peuvent rester vides. Cliquez sur **Suivant**.
  b. Sélectionnez le **type de connexion du serveur MBean**. Sélectionnez la version WebSphere Application Server qui convient pour votre environnement. Cliquez sur **Suivant**.
  c. Vérifiez que les informations WebSphere Application Server du panneau sont correctes. Cliquez sur **Suivant**.
  d. N'ajoutez qu'une définition de sous-noeud. Nommez cette définition de sous-noeud, mais ne mettez pas à jour la définition du port. Dans un

environnement WebSphere Application Server, les données peuvent être collectées sur tous les serveurs d'applications gérés par l'agent de noeud exécuté sur l'ordinateur. Cliquez sur **Suivant**.

e. S'il n'existe pas de serveurs de catalogues dans l'environnement, cliquez sur **OK**. S'il en existe, ajoutez une nouvelle configuration pour chaque serveur de catalogues, comme pour les serveurs conteneurs. Choisissez un nom unique pour le service de catalogue, de préférence, celui que vous avez utilisé au démarrage du service de catalogue. Cliquez sur **OK** pour terminer.

Remarque : Les serveurs conteneurs n'ont pas besoin d'être regroupés avec le service de catalogue.

Maintenant que l'agent et les serveurs sont configurés et prêts, dans la fenêtre qui suit, cliquez à l'aide du bouton droit de la souris sur `instance1` pour démarrer l'agent.

UNIX Pour configurer l'agent sur la plateforme UNIX sur la ligne de commande, procédez comme suit :

Un exemple est illustré ci-après pour les serveurs autonomes qui utilisent un type de connexion compatible JSR160. Cet exemple illustre trois conteneurs eXtreme Scale sur l'hôte unique (rhea00b02) et les adresses du programme d'écoute JMX sont respectivement 15000, 15001 et 15002. Il n'existe pas de serveurs de catalogues.

La sortie de l'utilitaire de configuration est affichée en *italiques à espacement fixe*, tandis que la réponse de l'utilisateur est en **gras à espacement fixe**. (Si aucune réponse utilisateur n'est requise, la valeur par défaut est sélectionnée en appuyant sur la touche Entrée.)

```
rhea00b02 # ./itmcmd config -A xt
Configuration de l'agent démarrée...
Entrez un nom d'instance (la valeur par défaut est : ) : inst1
Modifiez les paramètres "Agent de surveillance pour WebSphere eXtreme Scale" ? [ 1=Oui, 2=Non ] (la sélection par défaut est 1) :
Modifier les paramètres 'Java' ? [ 1=Oui, 2=Non ] (la sélection par défaut est 1) :
Répertoire de base Java (la sélection par défaut est C:\Program Files\IBM\Java50) : /opt/OG61/java
Niveau de trace Java [ 1=Erreur, 2=Avertissement, 3=Information, 4=Débogage minimum, 5=Débogage moyen, 6=Débogage maximum,
    7=Tous ] (la sélection par défaut est 1) :
Arguments JVM (la sélection par défaut est ):
Modifier les paramètres 'Connexion' ? [ 1=Oui, 2=Non ] (la sélection par défaut est 1) :
Type de connexion de serveur MBean [ 1=Serveur conforme à JSR-160, 2=WebSphere Application Server version 6.0,
3=WebSphere Application Server version 6.1, 4=WebSphere Application Server version 7.0 ] (la sélection par défaut est 1) : 1
Modifier les paramètres 'Serveur conforme à JSR-160' ? [ 1=Oui, 2=Non ] (la sélection par défaut est 1) :
ID utilisateur JMX (la sélection par défaut est ) :
Entrez Mot de passe JMX (la sélection par défaut est ) :
Entrez de nouveau : Mot de passe JMX (la sélection par défaut est ) :
Adresse URL de service JMX (la valeur par défaut est :
service:jmx:rmi:///jndi/rmi://localhost:port/objectgrid/MBeanServer) :
----------------------------------------
Informations de chemin de classe JMX
Chemins de base JMX (la sélection par défaut est ) :
Chemin de classes JMX (la sélection par défaut est ) :
Répertoires JAR JMX  (la sélection par défaut est ) :
Modifier les paramètres 'Service de catalogue WebSphere eXtreme Scale' ? [ 1=Oui, 2=Non ] (la sélection par défaut est 1) : 2
Modifier les paramètres 'Serveurs de grille WebSphere eXtreme Scale' ? [ 1=Oui, 2=Non ] (la sélection par défaut est 1) : 1
Aucun paramètre 'Serveurs de grille WebSphere eXtreme Scale' disponible
Modifiez les paramètres 'Serveurs de grille WebSphere eXtreme Scale', [1=Ajouter, 2=Modifier, 3=Supprimer, 4=Suivant,
5=Quitter] (la sélection par défaut est : 4) :
1
WebSphere eXtreme Scale Grid Servers (la valeur par défaut est ): rhea00b02_c0
Adresse URL de service JMX (la valeur par défaut est : service:jmx:rmi:///jndi/rmi://localhost:<port>/objectgrid/MBeanServer) :
service:jmx:rmi:///jndi/rmi://localhost:15000/objectgrid/MBeanServer

Paramètres 'Serveurs de grille WebSphere eXtreme Scale' :  WebSphere eXtreme Scale Grid Servers=ogx
Modifiez les paramètres 'Serveurs de grille WebSphere eXtreme Scale', [1=Ajouter, 2=Modifier, 3=Supprimer, 4=Suivant,
5=Quitter] (la sélection par défaut est : 4) : 1
Serveurs de grille WebSphere eXtreme Scale (la sélection par défaut est ) : rhea00b02_c1
Adresse URL de service JMX (la valeur par défaut est : service:jmx:rmi:///jndi/rmi://localhost:<port>/objectgrid/MBeanServer) :
service:jmx:rmi:///jndi/rmi://localhost:15001/objectgrid/MBeanServer
```

```
Paramètres 'Serveurs de grille WebSphere eXtreme Scale' :  WebSphere eXtreme Scale Grid Servers= rhea00b02_c1
Modifiez les paramètres 'Serveurs de grille WebSphere eXtreme Scale', [1=Ajouter, 2=Modifier, 3=Supprimer, 4=Suivant,
5=Quitter] (la sélection par défaut est : 4) : 1
Serveurs de grille WebSphere eXtreme Scale (la sélection par défaut est ) :
rhea00b02_c2
Adresse URL de service JMX (la valeur par défaut est : service:jmx:rmi:///jndi/rmi://localhost:<port>/objectgrid/MBeanServer) :
service:jmx:rmi:///jndi/rmi://localhost:15002/objectgrid/MBeanServer

Paramètres 'Serveurs de grille WebSphere eXtreme Scale' :  WebSphere eXtreme Scale Grid Servers= rhea00b02_c2
Modifiez les paramètres 'Serveurs de grille WebSphere eXtreme Scale', [1=Ajouter, 2=Modifier, 3=Supprimer, 4=Suivant,
5=Quitter] (la sélection par défaut est : 4) : 5

Cet agent se connectera-t-il à un TEMS ? [1=OUI, 2=NON]
(la sélection par défaut est 1) :
Nom d'hôte TEMS (la sélection par défaut est rhea00b00) :

Protocole de réseau [ip, sna, ip.pipe ou ip.spipe] (la sélection par défaut est ip.pipe) :

     Choisissez maintenant le prochain numéro de protocole parmi l'un des suivants :
     - ip
     - sna
     - ip.spipe
     - 0 pour aucun
Protocole de réseau 2 (la sélection par défaut est 0) :
Numéro de port IP.PIPE (la sélection par défaut est 1918) :
Entrez le nom de KDC_PARTITION (la sélection par défaut est null) :

Configurer la connexion TEMS secondaire ? [1=OUI, 2=NON] (la sélection par défaut est 2) :
Entrez Nom du réseau primaire optionnel ou 0 pour "aucun" (la sélection par défaut est 0) :
Configuration de l'agent terminée...
```

L'exemple précédent crée une instance d'agent appelée “inst1” et met à jour les paramètres de Java Home. Les serveurs de conteneur eXtreme Scale sont configurés, mais le service de catalogue n'est pas configuré.

Remarque : La procédure précédente crée un fichier texte au format suivant dans le répertoire : `<install_ITM>/config/<hôte>_xt_<nom de l'instance>.cfg`.

Exemple : `rhea00b02_xt_inst1.cfg`

Il est recommandé d'éditer ce fichier à l'aide de l'éditeur de texte en clair de votre choix. Voci un exemple de contenu d'un tel fichier :

```
INSTANCE=inst2 [SECTION=KQZ_JAVA [ { JAVA_HOME=/opt/OG61/java }  { JAVA_TRACE_LEVEL=ERROR }  ]
SECTION=KQZ_JMX_CONNECTION_SECTION [ { KQZ_JMX_CONNECTION_PROPERTY=KQZ_JMX_JSR160_JSR160 }  ]
SECTION=KQZ_JMX_JSR160_JSR160 [ { KQZ_JMX_JSR160_JSR160_CLASS_PATH_TITLE= }
{ KQZ_JMX_JSR160_JSR160_SERVICE_URL=service:jmx:rmi:///jndi/rmi://localho
st:port/objectgrid/MBeanServer }  { KQZ_JMX_JSR160_JSR160_CLASS_PATH_SEPARATOR= }  ]
SECTION=OGS:rhea00b02_c1 [ { KQZ_JMX_JSR160_JSR160_SERVICE_URL=service:jmx:
rmi:///jndi/rmi://localhost:15001/objectgrid/MBeanServer }  ]
SECTION=OGS:rhea00b02_c0 [ { KQZ_JMX_JSR160_JSR160_SERVICE_URL=service:jmx:
rmi:///jndi/rmi://localhost:15002/objectgrid/MBeanServer }  ]
SECTION=OGS:rhea00b02_c2 [ { KQZ_JMX_JSR160_JSR160_SERVICE_URL=service:jmx:
rmi:///jndi/rmi://localhost:15002/objectgrid/MBeanServer }  ]]
```

Voici un exemple illustrant une configuration sur un déploiement WebSphere Application Server :

```
rhea00b02 # ./itmcmd config -A xt
Configuration de l'agent démarrée...
Entrez un nom d'instance (la valeur par défaut est : ) : inst1
Modifiez les paramètres "Agent de surveillance pour WebSphere eXtreme Scale" ? [ 1=Oui, 2=Non ]
(la sélection par défaut est 1) : 1
Modifier les paramètres 'Java' ? [ 1=Oui, 2=Non ] (la sélection par défaut est 1) : 1
Java home (default is: C:\Program Files\IBM\Java50): /opt/WAS61/java
Niveau de trace Java [ 1=Erreur, 2=Avertissement, 3=Information, 4=Débogage minimum, 5=Débogage moyen, 6=Débogage maximum,
    7=Tous ] (la sélection par défaut est 1) :
Arguments JVM (la sélection par défaut est ):
Modifier les paramètres 'Connexion' ? [ 1=Oui, 2=Non ] (la sélection par défaut est 1) :
Type de connexion de serveur MBean [ 1=Serveur conforme à JSR-160, 2=WebSphere Application Server version 6.0,
3=WebSphere Application Server version 6.1, 4=WebSphere Application Server version 7.0 ]
(la sélection par défaut est 1) : 4
Modifier les paramètres 'WebSphere Application Server version 7.0' ? [ 1=Oui, 2=Non ]
(la sélection par défaut est 1) :ID utilisateur WAS (la sélection par défaut est ) :
Entrez Mot de passe WAS (la sélection par défaut est ) :
Entrez de nouveau : Mot de passe WAS (la sélection par défaut est ) :
```

```
Nom d'hôte WAS
(la sélection par défaut est localhost) : rhea00b02
Port WAS (la sélection par défaut est 2809) :
Protocole de connecteur WAS [ 1=rmi, 2=soap ]
(la sélection par défaut est 1) :
Nom de profil WAS (la sélection par défaut est ) : valeur par défaut
----------------------------------------
Informations de chemin de classe WAS
Chemins de base WAS (la sélection par défaut est C:\Program Files\IBM\WebSphere\AppServer;/opt/IBM/WebSphere/AppServer) :
/opt/WAS61
Chemin de classes WAS (la sélection par défaut est
runtimes/com.ibm.ws.admin.client_6.1.0.jar;runtimes/com.ibm.ws.ejb.thinclient_7.0.0.jar) :
Répertoires JAR WAS (la sélection par défaut est lib;plugins) :
Modifier les paramètres 'Serveurs de grille WebSphere eXtreme Scale' ? [ 1=Oui, 2=Non ] (la sélection par défaut est 1) :
Aucun paramètre 'Serveurs de grille WebSphere eXtreme Scale' disponible
Modifiez les paramètres 'Serveurs de grille WebSphere eXtreme Scale', [1=Ajouter, 2=Modifier, 3=Supprimer, 4=Suivant,
5=Quitter] (la sélection par défaut est : 4) : 1
Serveurs de grille WebSphere eXtreme Scale
(la sélection par défaut est ) : rhea00b02
Adresse URL de service JMX (la valeur par défaut est :
service:jmx:rmi:///jndi/rmi://localhost:<port>/objectgrid/MBeanServer) :

Paramètres 'Serveurs de grille WebSphere eXtreme Scale' settings:
WebSphere eXtreme Scale Grid Servers=rhea00b02
Modifiez les paramètres 'Serveurs de grille WebSphere eXtreme Scale', [1=Ajouter, 2=Modifier, 3=Supprimer, 4=Suivant,
5=Quitter] (la sélection par défaut est : 4) :
5
Modifier les paramètres 'Service de catalogue WebSphere eXtreme Scale' ? [ 1=Oui, 2=Non ] (la sélection par défaut est 1) : 2
Cet agent se connectera-t-il à un TEMS ? [1=OUI, 2=NON] (la sélection par défaut est 1) :
Nom d'hôte TEMS (la sélection par défaut est rhea00b02) :

Protocole de réseau [ip, sna, ip.pipe ou ip.spipe] (la sélection par défaut est ip.pipe) :

     Choisissez maintenant le prochain numéro de protocole parmi l'un des suivants :
     - ip
     - sna
     - ip.spipe
     - 0 pour aucun
Protocole de réseau 2 (la sélection par défaut est 0) :
Numéro de port IP.PIPE (la sélection par défaut est 1918) :
Entrez le nom de KDC_PARTITION (la sélection par défaut est null) :

Configurer la connexion TEMS secondaire ? [1=OUI, 2=NON] (la sélection par défaut est 2) :
Entrez Nom du réseau primaire optionnel ou 0 pour "aucun" (la sélection par défaut est 0) :
Configuration de l'agent terminée...
rhea00b02 #
```

Pour les déploiements WebSphere Application Server, vous n'avez pas besoin de créer plusieurs sous-noeuds. L'agent eXtreme Scale se connecte à l'agent de noeud pour collecter toutes les informations des serveurs d'applications dont il est responsable.

`SECTION=CAT` signifie une ligne de service de catalogue, tandis que `SECTION=OGS` signifie une ligne de configuration de serveur eXtreme Scale.

5. Configurez le port JMX pour tous les serveurs conteneurs eXtreme Scale.

   Si des serveurs conteneurs eXtreme Scale sont démarrés, sans l'argument **-JMXServicePort**, un serveur MBean reçoit un port dynamique. L'agent doit savoir à l'avance avec quel port JMX communiquer. L'agent ne fonctionne pas avec des ports dynamiques.

   Au démarrage des serveurs, vous devez spécifier l'argument `-JMXServicePort <numéro_port>` lorsque vous démarrez le serveur eXtreme Scale à l'aide de la commande `startOgServer.sh | .bat`. L'exécution de cette commande garantit que le serveur JMX du processus écoute sur un port statique prédéfini.

   Pour les exemples précédents dans l'installation UNIX, deux serveurs eXtreme Scale doivent être démarrés avec des ports définis :

   a. "-JMXServicePort" "15000" (pour rhea00b02_c0)

   b. "-JMXServicePort" "15001" (pour rhea00b02_c1)

   a. Démarrez l'agent eXtreme Scale.

En supposant que l'instance `inst1` ait été créée, comme dans l'exemple précédent, exécutez les commandes ci-après.

1) `cd <install_ITM>/bin`
2) `itmcmd agent –o inst1 start xt`

b. Arrêtez l'agent eXtreme Scale.

En supposant que l'instance “inst1” correspond à l'instance créée, comme dans l'exemple précédent, exécutez les commandes ci-après.

1) `cd <install_ITM>/bin`
2) `itmcmd agent –o inst1 stop xt`

6. Activez les statistiques pour tous les serveurs conteneurs eXtreme Scale.

Pour enregistrer les statistiques, l'agent utilise les beans gérés de statistiques eXtreme Scale. La spécification des statistiques eXtreme Scale doit être activée à l'aide de l'une des méthodes suivantes :

- en configurant les propriétés des serveurs pour activer toutes les statistiques au démarrage de la totalité des serveurs : all=enabled
- à l'aide de l'utilitaire d'exemple xsadmin pour activer les statistiques pour tous les conteneurs actifs : paramètres -setstatsspec all=enabled

Résultats

Une fois que tous les serveurs sont configurés et démarrés, les données des beans gérés sont affichées sur la console d'IBM Tivoli Portal. Les espaces de travail prédéfinis montrent les graphiques et mesures de données au niveau de chaque noeud.

Les espaces de travail suivants sont définis : **noeud eXtreme Scale serveurs de grilles** pour tous les noeuds surveillés.

- vue Transactions eXtreme Scale
- vue Fragment primaire eXtreme Scale
- vue Mémoire eXtreme Scale
- vue ObjectMap eXtreme Scale

Vous pouvez également configurer votre propre espace de travail. Pour plus d'informations, reportez-vous aux informations sur la personnalisation des espaces de travail, dans le centre de documentation d'IBM Tivoli Monitoring.

Surveillance des applications eXtreme Scale à l'aide de CA Wily Introscope

CA Wily Introscope est un produit de gestion tiers qui permet de détecter et de diagnostiquer les problèmes de performances dans les environnements d'application d'entreprise. eXtreme Scale inclut des détails sur la configuration de CA Wily Introscope pour introspecter certaines portions de l'environnement d'exécution de eXtreme Scale afin d'afficher et de valider rapidement les applications eXtreme Scale. CA Wily Introscope fonctionne de manière efficace pour les déploiements autonomes et WebSphere Application Server.

Présentation

Pour surveiller les applications eXtreme Scale avec CA Wily Introscope, vous devez placer des paramètres dans les fichiers PBD (ProbeBuilderDirective) qui vous permettent d'accéder aux informations de surveillance de eXtreme Scale.

Avertissement : Les points d'instrumentation d'Introscope peuvent changer avec chaque corriger ou version. Lorsque vous installez un nouveau groupe de correctifs ou une nouvelle version, recherchez dans la documentation les modifications apportées aux points d'instrumentation.

Vous pouvez configurer des fichiers PBD (ProbeBuilderDirective) de CA Wily Introscope pour surveiller vos applications eXtreme Scale. CA Wily Introscope est un produit de gestion des applications à l'aide duquel vous pouvez détecter, prioriser et diagnostiquer de manière proactive les problèmes de performances dans vos environnements d'application Web, composite et complexe.

Paramètres des fichiers PBD pour la surveillance du service de catalogue

Vous pouvez utiliser un ou plusieurs des paramètres ci-après dans votre fichier PBD pour surveiller le service de catalogue.

```
TraceOneMethodOfClass: com.ibm.ws.objectgrid.hamanager.HAControllerImpl changeDefinedCompleted
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.hamanager.HAControllerImpl viewChangeCompleted
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.hamanager.HAControllerImpl viewAboutToChange
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeat
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatCluster
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatCurrentLeader
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatDeadServer
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatNewLeader
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatNewServer
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.catalog.placement.PlacementServiceImpl
importRouteInfo BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.catalog.placement.PlacementServiceImpl heartbeat
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.catalog.placement.PlacementServiceImpl joinPlacementGroup
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}" TraceOneMethodOfClass:
com.ibm.ws.objectgrid.catalog.placement.PlacementServiceImpl classifyServer
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.catalog.placement.BalanceGridEventListener shardActivated
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.catalog.placement.BalanceGridEventListener shardDeactivate
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
```

Classes de surveillance du service de catalogue

HAControllerImpl
La classe HAControllerImpl gère le cycle de vie du groupe central et les événements de retour d'informations. Vous pouvez surveiller cette classe pour déterminer les modifications et la structure du groupe central.

ServerAgent
La classe ServerAgent est chargée de communiquer les événements du groupe central avec le service de catalogue. Vous pouvez surveiller les divers appels de signal de présence pour identifier les événements principaux.

PlacementServiceImpl
La classe PlacementServiceImpl coordonne les conteneurs. Vous pouvez utiliser les méthodes de cette classe pour surveiller les événements de jointure et de positionnement.

BalanceGridEventListener
La classe BalanceGridEventListener contrôle la position de leader du catalogue. Vous pouvez surveiller cette classe pour déterminer quel service de catalogue sert actuellement de leader.

Paramètres des fichiers PBD pour la surveillance des conteneurs

Vous pouvez utiliser un ou plusieurs des paramètres ci-après dans votre fichier PBD pour surveiller les conteneurs.

```
TraceOneMethodOfClass: com.ibm.ws.objectgrid.ShardImpl processMessage
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.plugins.CommittedLogSequenceListenerProxy applyCommitted
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.plugins.CommittedLogSequenceListenerProxy sendApplyCommitted
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.map.BaseMap evictMapEntries
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.checkpoint.CheckpointMapImpl$CheckpointIterator activateListener
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.hamanager.HAControllerImpl changeDefinedCompleted
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.hamanager.HAControllerImpl viewChangeCompleted
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.hamanager.HAControllerImpl viewAboutToChange
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent batchProcess
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeat
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatCluster
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatCurrentLeader
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatDeadServer
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatNewLeader
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatNewServer
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
```

Classes de surveillance des conteneurs

ShardImpl

La classe ShardImpl contient la méthode processMessage. La méthode processMessage est celle des demandes client. Avec cette méthode, vous pouvez obtenir les temps de réponse côté serveur et le nombre de demandes. En observant les résultats sur tous les serveurs et en surveillant l'utilisation des segments de mémoire, vous pouvez déterminer si la grille est équilibrée.

CheckpointIterator

La classe CheckpointIterator contient l'appel de méthode activateListener qui place les fragments primaires en mode homologue. Lorsque les fragments primaires sont placés en mode homologue, le fragment réplique est au même niveau que le fragment primaire une fois la méthode exécutée. Lorsqu'une réplique est régénérée à partir d'un fragment primaire complet, cette opération peut durer un certain temps. Le système n'ayant pas intégralement récupéré tant que cette opération n'est pas terminée, vous pouvez utiliser cette classe pour surveiller la progression de l'opération.

CommittedLogSequenceListenerProxy

La classe CommittedLogSequenceListenerProxy contient deux méthodes intéressantes. La méthode applyCommitted est exécutée pour chaque transaction et la méthode sendApplyCommitted est exécutée lorsque le fragment réplique extrait des informations. Le ratio de fréquence d'exécution de ces deux méthodes peut vous indiquer dans quelle mesure le fragment réplique est capable de suivre le fragment primaire.

Paramètres des fichiers PBD pour la surveillance des clients

Vous pouvez utiliser un ou plusieurs des paramètres ci-après dans votre fichier PBD pour surveiller les clients.

```
TraceOneMethodOfClass: com.ibm.ws.objectgrid.client.ORBClientCoreMessageHandler sendMessage
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.corba.cluster.ClusterStore bootstrap
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.corba.cluster.ClusterStore epochChangeBootstrap
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.map.BaseMap evictMapEntries
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.cluster.orb.routing.SelectionServiceImpl routeFailed
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.cluster.orb.routing.SelectionServiceImpl routeFailed
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.SessionImpl getMap
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.ObjectGridImpl getSession
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TurnOn: ObjectMap
SetFlag: ObjectMap
IdentifyClassAs: com.ibm.ws.objectgrid.ObjectMapImpl ObjectMap
TraceComplexMethodsifFlagged: ObjectMap BlamePointTracerDifferentMethods
"OGclient|{classname}|{method}"
```

Classes de surveillance des clients

ORBClientCoreMessageHandler
La classe ORBClientCoreMessageHandler est chargée d'envoyer les demandes d'application aux conteneurs. Vous pouvez surveiller la méthode sendMessage pour le temps de réponse des clients et le nombre de demandes.

ClusterStore
La classe ClusterStore contient les informations de routage côté client.

BaseMap
La classe BaseMap contient la méthode evictMapEntries qui est appelée lorsque l'expulseur souhaite supprimer des entrées de la mappe.

SelectionServiceImpl
La classe SelectionServiceImpl effectue les décisions de routage. Si le client décide de procéder à un basculement, vous pouvez utiliser cette classe pour afficher les actions réalisées à partir de ces décisions.

ObjectGridImpl
La classe ObjectGridImpl contient la méthode getSession que vous pouvez surveiller pour afficher le nombre de demandes pour cette méthode.

Surveillance d'eXtreme Scale à l'aide de Hyperic HQ

Hyperic HQ est une solution de surveillance tiers disponible gratuitement comme solution à code source ouvert ou produit d'entreprise. WebSphere eXtreme Scale inclut un plug-in qui permet aux agents Hyperic HQ de reconnaître les serveurs conteneurs eXtreme Scale et de fournir et regrouper des statistiques à l'aide de beans de gestion eXtreme Scale. Vous pouvez utiliser Hyperic HQ pour surveiller les déploiements eXtreme Scale autonomes.

Avant de commencer

- Ce jeu d'instructions concerne Hyperic Version 4.0. Si vous possédez une version plus récente de Hyperic, reportez-vous à la documentation de Hyperic pour plus d'informations, telles que les noms de chemin et la méthode de démarrage des agents et des serveurs.
- Téléchargez les installations des agents et serveurs Hyperic. Une installation de serveur doit être en cours d'exécution. Pour détecter tous les serveurs eXtreme Scale, un agent Hyperic doit être en cours d'exécution sur chaque machine sur laquelle un serveur eXtreme Scale est en cours d'exécution. Pour les informations de téléchargement et le support de documentation, voir le site Web Hyperic.
- Vous devez avoir accès aux fichiers `objectgrid-plugin.xml` et `hqplugin.jar`. Ces fichiers se trouvent dans le répertoire *racine_install_wxs*`/hyperic/etc`.

Pourquoi et quand exécuter cette tâche

En intégrant eXtreme Scale au logiciel de surveillance Hyperic HQ, vous pouvez surveiller et afficher graphiquement les mesures sur les performances de votre environnement. Vous configurez cette intégration en utilisant une implémentation de plug-in sur chaque agent.

Procédure

1. Démarrez vos serveurs eXtreme Scale. Le plug-in Hyperic recherche les processus locaux à connecter aux machines virtuelles Java qui exécutent eXtreme Scale. Pour se connecter correctement aux machines virtuelles Java, chaque serveur doit être démarré avec l'option **-jmxServicePort**. Pour plus d'informations sur le démarrage des serveurs à l'aide de l'option **-jmxServicePort**, voir «Script **startOgServer**», à la page 401.
2. Placez le fichier `extremescale-plugin.xml` et le fichier `wxshyperic.jar` dans les répertoires de plug-in appropriés du serveur et des agents, dans votre configuration Hyperic. Pour être intégrées à Hyperic, les installations des agents et du serveur doivent avoir accès au plug-in et aux fichiers JAR (archive Java). Le serveur peut permuter dynamiquement les configurations, mais vous devez effectuer l'intégration avant de démarrer l'un des agents.
   a. Placez le fichier `extremescale-plugin.xml` dans le répertoire `plugin` du serveur, qui se trouve à l'emplacement suivant :

      hyperic_home`/server_home/hq-engine/server/default/deploy/hq.ear/hq-plugins`
   b. Placez le fichier `extremescale-plugin.xml` dans le répertoire `plugin` de l'agent, qui se trouve à l'emplacement suivant :

      agent_home`/bundles/gent-4.0.2-939/pdk/plugins`
   c. Placez le fichier `wshyperic.jar` dans le répertoire `lib` de l'agent, qui se trouve à l'emplacement suivant :

      agent_home`/bundles/gent-4.0.2-939/pdk/lib`
3. Configurez l'agent. Le fichier `agent.properties` fait office de point de configuration pour l'environnement d'exécution de l'agent. Cette propriété se trouve dans le répertoire *rép_base_agent*`/conf`. Les clés suivantes sont facultatives, mais importantes pour le plug-in eXtreme Scale :
   - `autoinventory.defaultScan.interval.millis=<`*durée_en_millisecondes*`>`

     Définit l'intervalle en millisecondes entre les reconnaissances d'Agent.
   - `log4j.logger.org.hyperic.hq.plugin.extremescale.XSServerDetector=DEBUG`

     : Active les instructions de débogage prolixes à partir du plug-in eXtreme Scale.
   - `username=<`*username*`>` : Définit le nom d'utilisateur JMX (Java Management Extensions) si la sécurité est activée.
   - `password=<`*motdepasse*`>` : Définit le mot de passe JMX si la sécurité est activée.
   - `sslEnabled=<`*true*`|`*false*`>` : Indique au plug-in s'il doit utiliser SSL (Secure Sockets Layer). La valeur est `false` par défaut.
   - `trustPath=<`*chemin*`>` : Définit le chemin sécurisé de la connexion SSL.
   - `trustType=<`*type*`>` : Définit le type sécurisé de la connexion SSL.
   - `trustPass=<`*motdepasse*`>` : Définit le mot de passe sécurisé de la connexion SSL.

4. Démarrez la reconnaissance des agents. Les agents Hyperic envoient des informations de reconnaissance et des mesures au serveur. Utilisez le serveur pour personnaliser les vues de données et regrouper les objets d'inventaire logiques afin de générer des informations utiles. Une fois que le serveur est disponible, vous devez exécuter le script de lancement ou démarrer le service Windows pour l'agent :

   - Linux *agent_home*/bin/hq-agent.sh start
   - Windows Démarrez l'agent avec le service Windows.

   Une fois que vous avez démarré les agents, les serveurs sont détectés et les groupes sont configurés. Vous pouvez vous connecter à la console du serveur et choisir les ressources à ajouter à la base de données d'inventaire du serveur. La console du serveur se trouve à l'URL suivante par défaut : `http://<nom_hôte_serveur>:7080/`

5. Les statistiques doivent être activées pour que Hyperic puisse collecter des données statistiques.

   Utilisez l'action de contrôle **SetStatsSpec** sur la console Hyperic pour eXtreme Scale. Allez à la ressource, puis utilisez la liste déroulante **Action de contrôle** sous l'onglet **Contrôle** afin de spécifier un paramètre `SetStatsSpec` avec `ALL=enabled` dans la zone de texte **Arguments du contrôle**.

   Les serveurs de catalogues ne sont pas détectés par le filtre défini sur la console Hyperic. Voir les informations concernant la propriété **statsSpec** dans Fichier de propriétés du serveur, qui activent les statistiques au démarrage des conteneurs. Diverses options d'activation des statistiques sont décrites dans «Surveillance à l'aide de beans gérés (MBeans)», à la page 476 et dans Exemple : utilitaire **xsadmin**.

6. Surveillez les serveurs à l'aide de la console Hyperic. Une fois que les serveurs ont été ajoutés au modèle d'inventaire, leurs services ne sont plus requis.
   - **Vue Tableau de bord** : Lorsque vous avez affiché les événements de détection des ressources, vous vous êtes connecté à la vue du tableau de bord principal. Il s'agit d'une vue générique qui sert de centre de messagerie que vous pouvez personnaliser. Vous pouvez exporter des graphiques ou des objets d'inventaire dans ce tableau de bord principal.
   - **Vue Ressources** : Vous pouvez interroger et afficher l'intégralité du modèle d'inventaire à partir de cette page. Une fois que les services ont été ajoutés, chaque serveur eXtreme Scale est correctement libellé et répertorié sous la section des serveurs. Vous pouvez cliquer sur chacun des serveurs pour consulter les mesures de base.
7. Affichez l'intégralité de l'inventaire du serveur dans la page d'affichage des ressources. Dans cette page, vous pouvez sélectionner plusieurs serveurs ObjectGrid et les regrouper. Une fois que vous avez regroupé un ensemble de ressources, leurs mesures communes peuvent être représentées graphiquement pour montrer les superpositions et les différences entre les membres du groupe. Pour afficher une superposition, sélectionnez les mesures dans l'écran de votre groupe de serveurs. La mesure est affichée dans la zone de représentation graphique. Pour afficher une superposition pour tous les membres du groupe, cliquez sur le nom de mesure souligné. Vous pouvez exporter les graphiques, vues de noeud et superpositions comparatives de votre choix dans le tableau de bord principal, à l'aide du menu **Outils**.

Surveillance des informations eXtreme Scale dans DB2

Lorsque le chargeur JPALoader ou JPAEntityLoader est utilisé avec DB2 comme base de données dorsale, des informations spécifiques de eXtreme Scale peuvent être transmises à DB2. Vous pouvez afficher ces informations à l'aide d'un outil de contrôle des performances tel que DB2 Performance Expert qui permet de surveiller les applications eXtreme Scale qui accèdent à la base de données.

Avant de commencer

Voir «Collecte de trace», à la page 536 pour plus d'informations sur les différentes méthodes de définition de trace que vous pouvez utiliser.

Pourquoi et quand exécuter cette tâche

Lorsque le chargeur est configuré pour utiliser DB2 comme base de données dorsale, les informations eXtreme Scale suivantes peuvent être transmises à DB2 à des fins de surveillance :

- **Utilisateur :** spécifie le nom de l'utilisateur qui s'authentifie auprès de eXtreme Scale. Si l'authentification standard n'est pas utilisée, ce sont les principaux de l'authentification qui sont utilisés.
- **Nom du poste de travail :** spécifie le nom d'hôte, l'adresse IP du serveur de conteneur eXtreme Scale.
- **Nom de l'application :** spécifie le nom de l'unité de persistance ObjectGrid (si spécifié).
- **Informations de comptabilité :** indique l'ID unité d'exécution, le type de transaction, l'ID de transaction et la chaîne de connexion.

Pour savoir comment surveiller l'accès à la base de données, informez-vous sur DB2 Performance Expert.

Procédure

- Pour activer toutes les informations du client eXtreme Scale, définissez les chaînes de trace suivantes :

  ```
  ObjectGridClientInfo*=event=enabled
  ```

- Pour tout activer sauf les informations utilisateur, utilisez l'un des paramètres suivants :
  -
    ```
    ObjectGridClientInfo*=event=enabled,ObjectGridClientInfoUser=event=disabled
    ```
    ou
  -
    ```
    ObjectGridClientInfo=event=enabled
    ```

Résultats

Une fois que vous avez activé la fonction de trace, les données s'affichent dans l'outil de surveillance des performances, tel que DB2 Performance Expert.

Exemple

Dans l'exemple suivant, l'utilisateur `bob` est authentifié en tant qu'utilisateur eXtreme Scale. L'application accède à la grille de données `mygrid` en utilisant l'unité de persistance `DB2Hibernate`. Le serveur de conteneur s'appelle `XS_Server1`. Informations résultantes :

- **Utilisateur**=`bob`

- **Nom de poste de travail**=XS_Server1,192.168.1.101
- **Nom d'application**=mygrid,DB2Hibernate
- **Informations comptabilité**=1, DEFAULT,FE7954BD-0126-4000-E000-2298094151DB,com.ibm.db2.jcc.t4.b@71787178

Dans l'exemple suivant, l'utilisateur bob est authentifié en utilisant un jeton WebSphere Application Server.. L'application accède à la grille de données mygrid en utilisant l'unité de persistance DB2OpenJPA. Le serveur de conteneur s'appelle XS_Server2. Informations résultantes :

- **Utilisateur**

  =acme.principal.UserPrincipal[Bob],acme.principal.GroupPrincipal[admin]
- **Nom de poste de travail**=XS_Server2,192.168.1.102
- **Nom d'application**=mygrid,DB2OpenJPA
- **Informations comptabilité**=188,DEFAULT,FE72BC63-0126-4000-E000-851C092A4E33,com.ibm.ws.rsadapter.jdbc.WSJccSQLJConnection@2b432b43

Chapitre 9. Optimisation des performances

Vous pouvez optimiser les paramètres dans l'environnement pour augmenter les performances générales de votre environnement WebSphere eXtreme Scale.

Optimisation des systèmes d'exploitation et des paramètres réseau

En modifiant les paramètres de connexion, il est possible de réduire les temps d'attente TCP (Transmission Control Protocol) et la modification des tampons TCP permet d'améliorer les débits de transmission.

Systèmes d'exploitation

De tous les systèmes d'exploitation, Windows est celui qui a le moins besoin d'être optimisé, au contraire de Solaris, qui nécessite un maximum d'optimisation. Les informations suivantes, qui concernent chacun des systèmes spécifiés, sont susceptibles d'améliorer les performances de WebSphere eXtreme Scale. Vous devez procéder à l'optimisation en fonction de la charge de votre réseau et de vos applications.

Windows

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
Tcpip\Parameters
MaxFreeTcbs = dword:00011940
MaxHashTableSize = dword:00010000
MaxUserPort = dword:0000fffe
TcpTimedWaitDelay = dword:0000001e
```

Solaris

```
ndd -set /dev/tcp tcp_time_wait_interval 60000
fndd -set /dev/tcp tcp_keepalive_interval 15000
ndd -set /dev/tcp tcp_fin_wait_2_flush_interval 67500
ndd -set /dev/tcp tcp_conn_req_max_q 16384
ndd -set /dev/tcp tcp_conn_req_max_q0 16384
ndd -set /dev/tcp tcp_xmit_hiwat 400000
ndd -set /dev/tcp tcp_recv_hiwat 400000
ndd -set /dev/tcp tcp_cwnd_max 2097152
ndd -set /dev/tcp tcp_ip_abort_interval 20000
ndd -set /dev/tcp tcp_rexmit_interval_initial 4000
ndd -set /dev/tcp tcp_rexmit_interval_max 10000
ndd -set /dev/tcp tcp_rexmit_interval_min 3000
ndd -set /dev/tcp tcp_max_buf 4194304
```

AIX

```
/usr/sbin/no -o tcp_sendspace=65536
/usr/sbin/no -o tcp_recvspace=65536
/usr/sbin/no -o udp_sendspace=65536
/usr/sbin/no -o udp_recvspace=65536
/usr/sbin/no -o somaxconn=10000
/usr/sbin/no -o tcp_nodelayack=1
/usr/sbin/no –o tcp_keepinit=40
/usr/sbin/no –o tcp_keepintvl=10
```

LINUX

```
sysctl -w net.ipv4.tcp_timestamps=0
sysctl -w net.ipv4.tcp_tw_reuse=1
sysctl -w net.ipv4.tcp_tw_recycle=1
sysctl -w net.ipv4.tcp_fin_timeout=30
sysctl -w net.ipv4.tcp_keepalive_time=1800
sysctl -w net.ipv4.tcp_rmem="4096 87380 8388608"
sysctl -w net.ipv4.tcp_wmem="4096 87380 8388608"
sysctl -w net.ipv4.tcp_max_syn_backlog=4096
```

HP-UX

```
ndd -set /dev/tcp tcp_ip_abort_cinterval 20000
```

Propriétés ORB

Les propriétés ORB (Object Request Broker) modifient le comportement du transport de la grille de données. Ces propriétés peuvent être définies avec un fichier `orb.properties`, comme paramètres dans la console d'administration de WebSphere Application Server ou en tant que propriétés personnalisées dans ORB dans la console d'administration de WebSphere Application Server.

orb.properties

Le fichier `orb.properties` se trouve dans le répertoire `java/jre/lib`. Lorsque vous modifiez le fichier `orb.properties` dans un répertoire WebSphere Application Server `java/jre/lib`, les propriétés ORB sont mises à jour sur l'agent de noeud et toutes les autres machines virtuelles Java (JVM) qui utilisent l'environnement d'exécution Java (JRE). Si vous ne souhaitez pas ce comportement, utilisez des propriétés personnalisées ou les paramètres ORB de la console d'administration WebSphere Application Server.

Paramètres WebSphere Application Server par défaut

Par défaut, WebSphere Application Server a certaines propriétés définies dans ORB. Ces paramètres se trouvent dans les services de conteneur de serveur d'applications et le gestionnaire de déploiement. Ces paramètres par défaut remplacent les paramètres que vous créez dans le fichier `orb.properties`. Pour chaque propriété décrite, reportez-vous à la section **Où définir** pour déterminer l'emplacement de définition de la valeur suggérée.

Paramètres de descripteur de fichier

Pour UNIX et Linux, il existe une limite pour le nombre de fichiers ouverts autorisés par processus. C'est le système d'exploitation qui spécifie le nombre permis de fichiers ouverts. Si cette valeur est trop basse, une erreur d'allocation de mémoire se produit sur AIX, et trop de fichiers ouverts sont consignés.

Dans la fenêtre de terminal UNIX, augmentez cette valeur au-dessus de la valeur par défaut du système. Dans le cas de grosses machines SMP avec des clones, fixez une valeur illimitée.

Pour les configurations AIX, définissez la valeur `-1` (illimité) avec la commande `ulimit -n -1`.

Pour les configurations Solaris définissez la valeur `16384` avec la commande `ulimit -n 16384`.

Pour afficher la valeur en cours, utilisez la commande `ulimit –a`.

Paramètres de référence

Les paramètres ci-après peuvent servir de référence, mais il ne s'agit pas nécessairement des meilleurs paramètres pour chaque environnement. Il convient de comprendre les paramètres pour définir des valeurs adaptées à votre environnement.

```
com.ibm.CORBA.RequestTimeout=30
com.ibm.CORBA.ConnectTimeout=10
com.ibm.CORBA.FragmentTimeout=30
com.ibm.CORBA.LocateRequestTimeout=10
com.ibm.CORBA.ThreadPool.MinimumSize=256
com.ibm.CORBA.ThreadPool.MaximumSize=256
com.ibm.CORBA.ThreadPool.IsGrowable=false
com.ibm.CORBA.ConnectionMultiplicity=1
com.ibm.CORBA.MinOpenConnections=1024
com.ibm.CORBA.MaxOpenConnections=1024
com.ibm.CORBA.ServerSocketQueueDepth=1024
com.ibm.CORBA.FragmentSize=0
com.ibm.CORBA.iiop.NoLocalCopies=true
com.ibm.CORBA.NoLocalInterceptors=true
```

Descriptions des propriétés

Paramètres d'expiration

Les paramètres ci-après concernent le délai d'attente respecté par l'ORB avant d'abandonner des opérations de demande. Utilisez ces paramètres pour empêcher la création d'un trop grand nombre d'unités d'exécution dans une situation anormale.

Délai d'expiration de la demande

Nom de la propriété : com.ibm.CORBA.RequestTimeout

Valeur valide : entier indiquant un nombre de secondes.

Valeur suggérée : 30

Où définir : WebSphere Application Server administrative console

Description : indique le délai d'attente en secondes d'une réponse à une demande. Cette propriété influence la durée de la reprise en ligne du client en cas d'indisponibilité du réseau. Si vous spécifiez une valeur trop faible pour cette propriété, les demandes risquent d'arriver à expiration par inadvertance. Définissez soigneusement la valeur de cette propriété pour éviter les dépassements de délai d'attente.

Délai d'expiration de la connexion

Nom de la propriété : com.ibm.CORBA.ConnectTimeout

Valeur valide : entier indiquant le nombre de secondes.

Valeur suggérée : 10

Où définir : fichier `orb.properties`

Description : indique le délai d'attente en secondes d'une connexion de socket. Cette propriété, comme celle du délai d'expiration de la demande, peut influencer la durée de la reprise en ligne du client en cas d'indisponibilité du réseau. En règle générale, cette propriété doit avoir une valeur inférieure à la valeur de délai d'attente de la demande, car le délai d'établissement des connexions est relativement constant.

Délai d'expiration des fragments

Nom de la propriété : com.ibm.CORBA.FragmentTimeout

Valeur valide : entier indiquant le nombre de secondes.

Valeur suggérée : 30

Où définir : `orb.properties` file

Description : indique le délai d'attente en secondes d'une demande de fragment. Cette propriété est similaire à celle du délai d'expiration de la demande.

Paramètres du pool d'unités d'exécution

Ces propriétés restreignent la taille du pool d'unités d'exécution à un nombre spécifique d'unités d'exécution. Les unités d'exécution sont utilisées par l'ORB pour distribuer les demandes du serveur une fois qu'elles ont été reçues sur le socket. Si vous affectez des valeurs de propriété trop basses, vous allongez la file d'attente de sockets et éventuellement les délais.

Multiplicité des connexions

Nom de la propriété : com.ibm.CORBA.ConnectionMultiplicity

Valeur valide : entier indiquant le nombre de connexions entre le client et le serveur. La valeur par défaut est 1. Si vous spécifiez une valeur supérieure, le multiplexage est défini entre plusieurs connexions.

Valeur suggérée : 1

Où définir : fichier `orb.properties`**Description :** permet à ORB d'utiliser plusieurs connexions à un serveur. En théorie, la définition de cette valeur promeut le parallèlisme sur les connexions. En pratique, la multiplicité des connexions n'améliore pas les performances. Ne spécifiez pas ce paramètre.

Connexions ouvertes

Noms des propriétés : com.ibm.CORBA.MinOpenConnections, com.ibm.CORBA.MaxOpenConnections

Valeur valide : entier indiquant le nombre de connexions.

Valeur suggérée : 1024

Où définir : console d'administration WebSphere Application Server**Description :** indique les nombres minimal et maximal de connexions ouvertes. L'ORB conserve un cache des connexions établies avec les clients. Ces connexions sont purgées lorsque cette valeur est transmise. La purge des connexions peut nuire au comportement de la grille de données.

Peut augmenter

Nom de la propriété : com.ibm.CORBA.ThreadPool.IsGrowable

Valeur valide : booléenne. Affectée de `true` ou `false`.

Valeur suggérée : `false`

Où définir : `orb.properties` file**Description :** si la valeur est `true`, le pool d'unités d'exécution qu'utilise ORB pour les demandes entrantes peuvent croître au-delà de ce que le pool prend en charge. Si la taille du pool est dépassée, des unités d'exécution sont créées pour traiter la demande, mais elles ne sont pas placées dans un pool. Empêchez le pool d'unités d'exécution de croître en définissant la valeur `false`.

Longueur de la file d'attente des sockets de serveur

Nom de la propriété : com.ibm.CORBA.ServerSocketQueueDepth

Valeur valide : entier pour le nombre de connexions.

Valeur suggérée : `1024`

Où définir : fichier `orb.properties`**Description :** Indique la longueur de la file d'attente des connexions entrantes des clients. L'ORB place les connexions entrantes des clients en file d'attente. Si la file d'attente est saturée, les connexions sont refusées. Le refus de connexion peut affecter le comportement de la grille de données.

Taille du fragment

Nom de la propriété : com.ibm.CORBA.FragmentSize

Valeur valide : entier qui définit le nombre d'octets. La valeur par défaut est 1024.

Valeur suggérée : `0`

Où définir : fichier `orb.properties`**Description :** Indique la taille de paquet maximale utilisée par l'ORB lors de l'envoi d'une demande. Si la taille d'une demande est supérieure à la taille limite du fragment, la demande est divisée en fragments de demande qui sont envoyés séparément, puis réassemblés sur le serveur. La fragmentation des demandes est utile sur les réseaux non fiables où les packets doivent parfois être renvoyés. Toutefois, si le réseau est fiable, la division des demandes en fragments peut entraîner un traitement superflu.

Copies non locales

Nom de la propriété : com.ibm.CORBA.iiop.NoLocalCopies

Valeur valide : booléenne. Affectée de `true` ou `false`.

Valeur suggérée : `true`

Où définir : console d'administration WebSphere Application Server, paramètre de **transmission par référence**. **Description :** Indique si l'ORB utilise la transmission par référence. Par défaut, l'ORB utilise un appel de transmission par valeur. Un appel de transmission par valeur occupe plus de place et entraîne une sérialisation plus importante dans le chemin lorsqu'une interface est démarrée localement. Si vous spécifiez la valeur true, l'ORB utilise une méthode de transmission par référence, qui est plus efficace que l'appel de transmission par valeur.

Intercepteurs non locaux

Nom de la propriété : com.ibm.CORBA.NoLocalInterceptors

Valeur valide : booléenne. Affectée de `true` ou `false`.

Valeur suggérée : `true`

Où définir : fichier `orb.properties` **Description :** indique si l'ORB démarre les intercepteurs de demande, même lorsque des demandes locales (processus internes) sont faites. Les intercepteurs utilisés par WebSphere eXtreme Scale pour la gestion de la sécurité et des routes ne sont pas obligatoires si la demande est gérée au sein du processus. Les intercepteurs qui circulent entre les processus ne sont requis que pour les opérations RPC (Remote Procedure Call). En indiquant la non-utilisation d'intercepteurs locaux, vous pouvez éviter le traitement supplémentaire que génère l'utilisation d'intercepteurs locaux.

Avertissement : Si vous utilisez la sécurité WebSphere eXtreme Scale, affectez à la propriété com.ibm.CORBA.NoLocalInterceptors la valeur `false`. L'infrastructure de sécurité utilise des intercepteurs pour l'authentification.

Optimisation des machines virtuelles Java

Vous devez prendre en compte plusieurs aspects spécifiques de l'optimisation des machines virtuelles Java (JVM) pour optimiser les meilleures performances WebSphere eXtreme Scale. Dans la plupart des cas, quelques paramètres JVM spéciaux sont nécessaires ou aucun. Si de nombreux objets sont stockés dans la grille de données, définissez une taille de pile appropriée pour éviter de manquer de mémoire.

7.1.1+ En configurant eXtremeMemory, vous pouvez stocker des objets dans la mémoire native plutôt que dans le segment de mémoire Java. La configuration eXtremeMemory active eXtremeIO, un nouveau mécanisme de transport. En retirant des objets du segment de mémoire Java, vous pouvez éviter les pauses de récupération d'espace, ce qui permet de bénéficier de performances plus constantes et de temps de réponse plus prévisibles. Pour plus d'informations, voir «Configuration de IBM eXtremeMemory et de IBM eXtremeIO», à la page 278.

Plateformes testées

Les tests de performance ont été réalisés principalement sur des ordinateurs AIX (32 voies), Linux (quatre voies) et Windows (huit voies). Avec des ordinateurs AIX haut de gamme, vous pouvez tester des scénarios qui font appel à de nombreuses unités d'exécution pour identifier et corriger les points de conflit.

Récupération de place

WebSphere eXtreme Scale crée des objets temporaires associés à chaque transaction tels que la demande et la réponse ainsi la séquence de journal. Ces objets affectant l'efficacité de la récupération de place, l'optimisation de la récupération de place est cruciale.

Toutes les machines virtuelles Java actuelles utilisent des algorithmes de récupération de place en parallèle, ce qui implique que l'utilisation d'un plus grand nombre de coeurs peut réduire les pauses dans la récupération de place. Un serveur physique avec huit coeurs a une récupération plus rapide qu'un serveur physique avec quatre coeurs.

Lorsque l'application doit gérer une large quantité de données pour chaque partition, la récupération de place peut être un facteur déterminant. Un scénario de lecture principalement est performant même avec de grands segments (20 Go ou plus) si un collecteur générationnel est utilisé. Toutefois, une fois que le segment de réservation est rempli, une pause proportionnelle à la taille de pile réelle et au nombre de processeurs sur l'ordinateur se produit. Cette pause peut être longue sur les petites ordinateurs avec de grands segments de mémoire.

IBM virtual machine for Java garbage collection

Pour la machine virtuelle IBM pour Java, utilisez le collecteur **optavgpause** pour les scénarios impliquant des mises à jour fréquentes (100 % des transactions modifient les entrées). Le collecteur **gencon** fonctionne d'une manière similaire au collecteur **optavgpause** pour les scénarios où les données sont mises à jour peu fréquemment (10 % du temps au plus). Expérimentez les deux collecteurs pour savoir lequel est le mieux adapté à vos besoins. Utilisez la récupération de place prolixe pour vérifier le pourcentage de temps passé à la récupération de place. Des cas ont été relevés où 80 % de l'exécution sont consacrés à la récupération de place jusqu'à ce que l'optimisation corrige le problème.

Utilisez le paramètre **-Xgcpolicy** pour changer le mécanisme de collecte de place. La valeur du paramètre **-Xgcpolicy** peut être **-Xgcpolicy:gencon** ou **-Xgcpolicy:optavgpause**, selon le récupérateur de place que vous voulez utiliser.

- Dans une configuration WebSphere Application Server, définissez le paramètre **-Xgcpolicy** dans la console d'administration. Cliquez sur **Serveurs** > **Serveurs d'applications** > **server_name** > **Définition de processus** > **Java Virtual Machine**. Ajoutez le paramètre dans le champ des **arguments JVM génériques**.
- Dans une configuration autonome, envoyez le paramètre **-jvmArgs** au script **startOgServer** pour spécifier le récupérateur de place. Le paramètre **-jvmArgs** doit être le dernier paramètre envoyé au script.

Autres options de récupération de place

Avertissement : Si vous utilisez une machine virtuelle Java Sun, il peut être nécessaire d'ajuster la récupération de place et d'optimiser la stratégie.

WebSphere eXtreme Scale prend en charge WebSphere Real Time Java. Avec WebSphere Real Time Java, la réponse du traitement des transactions pour WebSphere eXtreme Scale est plus cohérente et prévisible. En conséquence, l'impact de la récupération de place et de la planification des unités d'exécution est considérablement réduit. L'impact est réduit au point où l'écart type du temps de réponse est inférieur à 10 % du langage Java classique.

Performance de la JVM

WebSphere eXtreme Scale peut être exécuté sur différentes versions de Java Platform, Standard Edition. WebSphere eXtreme Scale prend en charge la Java SE Version 5 et les versions suivantes. Pour optimiser la productivité et les performances des développeurs, utilisez Java SE 5 ou version ultérieure pour tirer parti des annotations et de la récupération de place améliorée. WebSphere eXtreme Scale fonctionnent dans des machines virtuelles Java 32 ou 64 bits.

WebSphere eXtreme Scale est testé avec un sous-ensemble de machines virtuelles disponibles mais la liste de prise en charge n'est pas exhaustive. Vous pouvez exécuter WebSphere eXtreme Scale sur n'importe quelle machine JVM de fournisseur au niveau d'édition 5 ou suivant. Toutefois, en cas de problème avec la machine JVM d'un fournisseur, vous devez contacter le fournisseur de la machine JVM pour obtenir une assistance. Si possible, utilisez la machine JVM de l'environnement d'exécution WebSphere sur n'importe quelle plateforme qui prend en charge WebSphere Application Server.

Pour la plupart des scénarios dans lesquels WebSphere eXtreme Scale est utilisé, la machine JVM Java SE Version 6 fonctionne mieux que l'édition 5. En règle générale, utilisez la dernière version disponible de Java Platform, Standard Edition pour obtenir de meilleures performances.

Taille de pile

Il recommandé d'utiliser des segments de mémoire de 1 à 2 Go avec une machine virtuelle Java pour quatre coeurs. La taille de segment de mémoire optimal dépend des facteurs suivants :

- le nombre d'objets actifs présents dans le segment
- le degré de complexité des objets actifs présents dans le segment
- le nombre de coeurs utilisables par la machine virtuelle Java

Par exemple, une application qui stocke des tableaux de 1 Ko peut utiliser une segment de mémoire beaucoup plus grand qu'une application qui utilise des graphiques complexes de POJO.

Nombre d'unités d'exécution

Le nombre d'unités d'exécution dépend de quelques facteurs. Une limite existe pour le nombre d'unités d'exécution pouvant être gérées par un seul fragment. Un fragment est une instance de partition et peut être un fragment primaire ou une réplique. Avec un nombre plus important de fragments pour chaque JVM, vous disposez de plusieurs unités d'exécution avec chaque fragment supplémentaire fournissant plus de chemins simultanés d'accès aux données. Chaque fragment est aussi concurrent que possible même si l'accès simultané est limité.

Exigences de la fonction ORB (Object Request Broker)

Le kit de développement de logiciels IBM comprend une implémentation ORB IBM qui a été testée avec WebSphere Application Server et WebSphere eXtreme Scale. Pour faciliter le processus de prise en charge, utilisez une machine virtuelle Java IBM. Les autres implémentations de machines virtuelles Java utilisent une autre fonction ORB. L'ORB IBM est fourni uniquement avec les machines virtuelles IBM Java. WebSphere eXtreme Scale requiert une fonction ORB opérationnelle. Vous pouvez utiliser WebSphere eXtreme Scale avec les ORB d'autres fournisseurs. Toutefois, si vous avez un problème avec un ORB de fournisseur, vous devez contacter le fournisseur ORB pour obtenir une assistance. L'implémentation ORB IBM est compatible avec des machines virtuelles Java tierces et peut être remplacée si nécessaire.

Optimisation d'orb.properties

En laboratoire, le fichier suivant a été utilisé sur les grilles de données de jusqu'à 1 500 machines virtuelles Java. Le fichier `orb.properties` se trouve dans le dossier `lib` de l'environnement d'exécution.

```
# IBM JDK properties for ORB
org.omg.CORBA.ORBClass=com.ibm.CORBA.iiop.ORB
org.omg.CORBA.ORBSingletonClass=com.ibm.rmi.corba.ORBSingleton

# WS Interceptors
org.omg.PortableInterceptor.ORBInitializerClass.com.ibm.ws.objectgrid.corba.ObjectGridInitializer

# WS ORB & Plugins properties
com.ibm.CORBA.ForceTunnel=never
com.ibm.CORBA.RequestTimeout=10
com.ibm.CORBA.ConnectTimeout=10

# Needed when lots of JVMs connect to the catalog at the same time
com.ibm.CORBA.ServerSocketQueueDepth=2048

# Clients and the catalog server can have sockets open to all JVMs
com.ibm.CORBA.MaxOpenConnections=1016

# Thread Pool for handling incoming requests, 200 threads here
com.ibm.CORBA.ThreadPool.IsGrowable=false
com.ibm.CORBA.ThreadPool.MaximumSize=200
com.ibm.CORBA.ThreadPool.MinimumSize=200
com.ibm.CORBA.ThreadPool.InactivityTimeout=180000

# No splitting up large requests/responses in to smaller chunks
com.ibm.CORBA.FragmentSize=0
```

Optimisation de la valeur de l'intervalle des pulsations pour la détection des basculements

Le paramètre d'intervalle du signal de présence permet de configurer le laps de temps séparant deux vérifications par le système des serveurs en panne.

Pourquoi et quand exécuter cette tâche

La configuration des basculements varie en fonction du type d'environnement que vous utilisez. Si vous utilisez un environnement autonome, vous pouvez configurer les basculements à l'aide de la ligne de commande. Si vous utilisez un environnement WebSphere Application Server Network Deployment, vous devez les configurer à partir de la console d'administration de WebSphere Application Server Network Deployment.

Procédure

- Configurez les basculements pour les environnements autonomes.

  Vous pouvez configurer les intervalles des pulsations sur la ligne de commande à l'aide du paramètre **-heartbeat** dans le fichier de script **startOgServer**. Affectez à ce paramètre l'une des valeurs suivantes :

  Tableau 30. Intervalles de signal de présence

Valeur	Action	Description
0	Standard (par défaut)	Les basculements sont généralement détectés dans les 30 secondes.
-1	Elevé	Les basculements sont généralement détectés dans les 5 secondes.
1	Souple	Les basculements sont généralement détectés dans les 180 secondes.

  Un intervalle élevé entre les signaux de présence peut être utile si les processus et le réseau sont stables. Si le réseau ou les processus ne sont pas configurés de manière optimale, il peut manquer des signaux de présence, ce qui peut fausser la détection des incidents.

- Configurez les basculements pour les environnements WebSphere Application Server.

  Vous pouvez configurer WebSphere Application Server Network Deployment Version 6.0.2 ou ultérieure pour permettre des basculements très rapides de WebSphere eXtreme Scale. La durée par défaut de pour les incidents matériels est d'environ 200 secondes. Un incident matériel est un ordinateur physique, une panne du serveur, déconnexion de câble réseau ou une erreur du système d'exploitation. Les incidents dus aux pannes de processus ou à des échecs logiciels sont généralement basculés en moins d'une seconde. La détection des incidents logiciels est effectuée lorsque les sockets réseau du processus inactif sont fermés automatiquement par le système d'exploitation du serveur qui héberge le processus.

  Configuration des signaux de présence du groupe central

  Si WebSphere eXtreme Scale est exécuté dans un processus WebSphere Application Server, il hérite des caractéristiques de reprise en ligne des paramètres du groupe central du serveur d'applications. Les sections suivantes décrivent comment configurer les paramètres des signaux de présence du groupe central pour différentes versions de WebSphere Application Server Network Deployment :

  – **Mise à jour des paramètres des groupes centraux de WebSphere Application Server Network Deployment Version 6.x et 7.x :**

    Spécifiez l'intervalle des signaux de présence en secondes sur les versions 6.0 à 6.1.0.12 de WebSphere Application Server ou en millisecondes à partir de la version 6.1.0.13. Vous devez également spécifier le nombre de signaux de présence manqués. Cette valeur indique le nombre maximal de signaux de présence manquants avant qu'une machine virtuelle Java (JVM) ne soit

considérée comme défectueuse. Le délai de détection des incidents matériels est approximativement égal au produit de l'intervalle des signaux de présence par le nombre de signaux de présence manqués.

Ces propriétés sont spécifiées à l'aide des propriétés personnalisées sur le groupe central à l'aide de la console d'administration de WebSphere. Pour des informations de configuration détaillées, voir la rubrique Propriétés personnalisées de groupe central. Ces propriétés doivent être spécifiées pour tous les groupes centraux utilisés par l'application :

- L'intervalle des pulsations est spécifié à l'aide de la propriété personnalisée IBM_CS_FD_PERIOD_SEC pour les secondes ou de la propriété personnalisée IBM_CS_FD_PERIOD_MILLIS pour les millisecondes (nécessite la version 6.1.0.13 ou une version ultérieure).
- Le nombre de signaux de présence manqués est spécifié à l'aide de la propriété personnalisée IBM_CS_FD_CONSECUTIVE_MISSED.

La valeur par défaut de la propriété IBM_CS_FD_PERIOD_SEC est de 20 et celle de la propriété IBM_CS_FD_CONSECUTIVE_MISSED, de 10. Si la propriété IBM_CS_FD_PERIOD_MILLIS est spécifiée, elle remplace les propriétés personnalisées IBM_CS_FD_PERIOD_SEC définies. Les valeurs de ces propriétés correspondent à des entiers.

Utilisez les paramètres suivants pour spécifier un délai de détection des incidents de 1500 ms pour les serveurs WebSphere Application Server Network Deployment Version 6.x :

- Spécifiez `IBM_CS_FD_PERIOD_MILLIS = 750` (WebSphere Application Server Network Deployment V6.1.0.13 et versions ultérieures)
- Spécifiez `IBM_CS_FD_CONSECUTIVE_MISSED = 2`

– **Mise à jour des paramètres des groupes centraux de WebSphere Application Server Network Deployment Version 7.0**

WebSphere Application Server Network Deployment Version 7.0 fournit deux paramètres de groupe central qui peuvent être ajustés pour augmenter ou réduire le délai de détection des incidents :

- **Période de transmission du signal de présence.** La valeur par défaut est de 30000 millisecondes.
- **Période d'expiration du signal de présence.** La valeur par défaut est de 180000 millisecondes.

Pour plus de détails sur la manière de modifier ces paramètres, voir la rubrique relative à la WebSphere Application Server Network Deployment reconnaissance et de détection des incidents dans le centre de documentation.

Utilisez les paramètres suivants pour spécifier un délai de détection des incidents de 1500 ms pour les serveurs WebSphere Application Server Network Deployment Version 7 :

- Spécifiez une période de transmission du signal de présence de 750 millisecondes.
- Spécifiez une période d'expiration du signal de présence de 1500 millisecondes.

Que faire ensuite

Lorsque vous modifiez ces paramètres pour réduire les délais de basculement, certains points d'optimisation du système sont à prendre en compte. Tout d'abord, Java n'est pas un environnement en temps réel. Des unités d'exécution peuvent être retardées si la JVM connaît des délais de récupération de place importants. Les unités d'exécution risquent également d'être retardées si la charge de la machine

qui héberge la JVM est considérable (à cause de la JVM elle-même ou d'autres processus exécutés sur cette machine). Si les unités d'exécution sont retardées, les signaux de présence risquent de ne pas être envoyés à temps. Au pire, ils risquent d'être retardés du délai requis pour la reprise en ligne. Si des unités d'exécution sont retardées, des incidents sont détectés à tort. Le système doit être optimisé et dimensionné de sorte à éviter la détection de faux incidents en production. Il est recommandé pour cela de tester la charge de manière adéquate.

Remarque : La version actuelle d'eXtreme Scale prend en charge WebSphere Real Time.

Optimisation de la récupération de place avec WebSphere Real Time

Utiliser WebSphere eXtreme Scale avec WebSphere Real Time augmente la cohérence et la prévisibilité des performances de débit qu'offre la stratégie standard de récupération de place employée dans le JRE (IBM Java SE Runtime Environment). Le ratio coûts/avantages est variable. WebSphere eXtreme Scale crée un grand nombre d'objets temporaires associés à chaque transaction. Ces objets temporaires s'occupent des demandes, des réponses, des séquences de journaux et des sessions. Sans WebSphere Real Time, les temps de réponse des transactions peuvent grimper à des centaines de millisecondes. Mais WebSphere Real Time utilisé avec WebSphere eXtreme Scale peut augmenter l'efficacité de la récupération de place et faire tomber à 10% les temps de réponse de la configuration autonome.

WebSphere Real Time en environnement autonome

Il est possible d'utiliser WebSphere Real Time avec WebSphere eXtreme Scale. En activant WebSphere Real Time, l'on obtient une récupération de place plus prévisible grâce à des temps de réponses stables et cohérents et des débits de transactions dans un environnement eXtreme Scale autonome.

Avantages de WebSphere Real Time

WebSphere eXtreme Scale crée un grand nombre d'objets temporaires associés à chaque transaction. Ces objets temporaires s'occupent des demandes, des réponses, des séquences de journaux et des sessions. Sans WebSphere Real Time, les temps de réponse des transactions peuvent grimper à des centaines de millisecondes. Mais WebSphere Real Time utilisé avec WebSphere eXtreme Scale peut augmenter l'efficacité de la récupération de place et faire tomber à 10% les temps de réponse de la configuration autonome.

Activer WebSphere Real Time

Installez WebSphere Real Time et WebSphere eXtreme Scale autonome sur les ordinateurs sur lesquels vous prévoyez d'exécuter eXtreme Scale. Définissez la variable d'environnement JAVA_HOME pour qu'elle pointe sur un JRE standard Java SE standard.

Définissez la variable d'environnement JAVA_HOME pour qu'elle pointe sur le WebSphere Real Time installé. Puis activez WebSphere Real Time comme indiqué ci-après.

1. Dans le fichier `objectgridRoot/bin/setupCmdLine.sh | .bat`, supprimez le commentaire de la ligne suivante :

   ```
   WXS_REAL_TIME_JAVA="-Xrealtime -Xgcpolicy:metronome
   -Xgc:targetUtilization=80"
   ```

2. Enregistrez le fichier.

WebSphere Real Time est à présent activé. Pour le désactiver, il vous suffit de repasser la même ligne en commentaire.

Pratiques recommandées

WebSphere Real Time confère aux transactions eXtreme Scale des temps de réponse plus prévisibles. Les résultats montrent que les temps de réponse d'une transaction eXtreme Scale s'améliorent de manière significative avec WebSphere Real Time si on les compare à ceux obtenus par le récupérateur de place par défaut de Java. L'activation de WebSphere Real Time avec eXtreme Scale est un must si la stabilité et les temps de réponse de votre application sont essentiels.

Les pratiques recommandées développées ci-après expliquent comment rendre WebSphere eXtreme Scale encore plus efficace grâce à une optimisation et une programmation fonction de la charge attendue.

- Définissez le bon niveau d'utilisation du processeur et de récupération de place.

  WebSphere Real Time donne les moyens de contrôler l'utilisation du processeur de manière à contrôler et à réduire l'impact de la récupération de place sur votre application. Le paramètre `-Xgc:targetUtilization=NN` permet de spécifier `NN` comme pourcentage du processeur qui est utilisé par votre application toutes les 20 secondes. Par défaut, cette valeur est de 80 % pour WebSphere eXtreme Scale, mais vous pouvez modifier le script dans le fichier `objectgridRoot/bin/setupCmdLine.sh` pour définir un autre chiffre, `70`, par exemple, qui libère davantage de capacité processeur pour le récupérateur de place. Déployez suffisamment de serveurs pour maintenir la charge processeur en dessous de 80 % pour vos applications.
- Augmentez la taille de la mémoire dynamique.

  WebSphere Real Time utilise davantage de mémoire que le Java standard, aussi, prévoyez plus de mémoire dynamique pour votre WebSphere eXtreme Scale et définissez la taille du segment mémoire lors du démarrage des serveurs de catalogue avec le paramètre `–jvmArgs –XmxNNNM` dans la commande **ogStartServer**. Vous pouvez, par exemple, utiliser le paramètre `–jvmArgs –Xmx500M` pour démarrer des serveurs de catalogue et utilisez une taille mémoire appropriée pour démarrer les conteneurs. Vous pouvez fixer la taille de la mémoire à 60-70 % de la taille prévue par machine virtuelle Java pour vos données . Si vous ne définissez pas cette valeur, une erreur OutOfMemoryError risque de se produire. Vous pouvez également, si vous le souhaitez, utiliser le paramètre `–jvmArgs –Xgc:noSynchronousGCOnOOM` pour empêcher le comportement non déterministe lorsque la machine virtuelle Java est à court de mémoire.
- Ajustez les unités d'exécution pour la récupération de place.

  WebSphere eXtreme Scale crée un grand nombre d'objets temporaires associés à chaque transaction et aux unités d'exécution RPC (Remote Procedure Call). La récupération de place présente des avantages pour les performances si votre ordinateur dispose de suffisamment de cycles processeur. Le nombre d'unités d'exécution est de 1 par défaut. L'argument `–Xgcthreads n` permet de modifier ce nombre. La valeur suggérée pour cet argument est le nombre de coeurs qui sont disponibles en prenant en considération le nombre de machines virtuelles Java par ordinateur.
- Ajustez les performances pour les applications à exécution courte avec WebSphere eXtreme Scale.

  WebSphere Real Time est optimisé pour les applications à exécution longue. D'ordinaire, vous avez besoin d'exécuter des transactions WebSphere eXtreme Scale pendant deux heures en continu pour obtenir des données de

performances fiables. Le paramètre –Xquickstart donne de meilleures performances à vos applications à exécution courte. Ce paramètre indique au compilateur JIT (just-in-time) d'utiliser un bas niveau d'optimisation.

- Réduisez la file d'attente des clients WebSphere eXtreme Scale et les relais clients WebSphere eXtreme Scale.

  Le principal avantage d'utiliser WebSphere eXtreme Scale avec WebSphere Real Time est de bénéficier de temps de réponse des transactions extrêmement fiables, qui représentent usuellement une amélioration de l'ordre de plusieurs fois l'écart constaté dans les temps de réponse des transactions. Toutes les demandes clients mises en file d'attente et tous les relais de ces demandes effectuées via d'autres logiciels ont un impact sur les temps de réponse qui échappent au contrôle de WebSphere Real Time et de WebSphere eXtreme Scale. Vous devez modifier les paramètres de vos unités d'exécution et de vos sockets pour conserver une charge à la fois ferme et fluide sans retards significatifs et vous devez diminuer la profondeur des files d'attente.

- Ecrivez des applications WebSphere eXtreme Scale qui utilisent les unités d'exécution WebSphere Real Time.

  Sans modifier votre application, vous pouvez obtenir des temps de réponse WebSphere eXtreme Scale des transactions extrêmement fiable représentant une amélioration de l'ordre de plusieurs fois l'écart standard dans les temps de réponse des transactions. Vous pouvez exploiter davantage les unités d'exécution de vos applications transactionnelles en passant du threading Java standard au RealtimeThread, qui fournit un meilleur contrôle de la priorité des unités d'exécution et de la planification.

  Actuellement, votre application comporte le code suivant :

  ```
  public class WXSCacheAppImpl extends Thread implements WXSCacheAppIF
  ```

  Vous pouvez remplacer ce code par le code suivant :

  ```
  public class WXSCacheAppImpl extends RealtimeThread implements
  WXSCacheAppIF
  ```

WebSphere Real Time sur WebSphere Application Server

Vous pouvez utiliser WebSphere Real Time avec eXtreme Scale dans un environnement WebSphere Application Server Network Deployment de version 7.0. L'activation de WebSphere Real Time permet d'obtenir une récupération de place plus prévisible avec des temps de réponses et des débits de transactions stables et cohérents.

Avantages

Utiliser WebSphere eXtreme Scale avec WebSphere Real Time augmente la cohérence et la prévisibilité des performances de débit qu'offre la stratégie standard de récupération de place employée dans le JRE (IBM Java SE Runtime Environment). Le ratio coûts/avantages est variable en fonction de plusieurs critères. Voici quelques-uns des principaux critères :

- capacités en serveurs : mémoire disponible, vitesse et taille des processeurs, vitesse et utilisation du réseau
- charges des serveurs : charge processeur soutenue, charge processeur de pointe
- configuration Java : taille des segments, utilisation cible, unités d'exécution de récupération de place
- configuration du mode copie de WebSphere eXtreme Scale : tableau d'octets ou stockage POJO

- points propres aux applications : utilisation des unités d'exécution, conditions requises et tolérance des réponses, taille des objets, etc.

En plus de la stratégie métronome de récupération de place utilisable dans WebSphere Real Time, il existe des stratégies optionnelles proposées par le JRE IBM standard. Ces stratégies, optthruput (stratégie par défaut), gencon, optavgpause et subpool sont spécifiquement conçues pour résoudre les différents besoins et environnements des applications. Pour plus d'informations sur ces stratégies, voir «Optimisation des machines virtuelles Java», à la page 496. En fonction des besoins de l'application et de l'environnement, ainsi que des ressources et des restrictions, le prototypage d'une ou plusieurs de ces stratégies peut vous garantir la satisfaction de ces besoins et vous aider à déterminer à coup sûr une stratégie optimale.

Possibilités avec WebSphere Application Server Network Deployment

1. Voici quelques-unes des versions prises en charge :
   - WebSphere Application Server Network Deployment version 7.0.0.5 et au-dessus
   - WebSphere Real Time V2 SR2 for Linux et au-dessus. Pour plus d'informations, voir IBM WebSphere Real Time V2 for Linux
   - WebSphere eXtreme Scale version 7.0.0.0 et au-dessus
   - Linux 32 et 64 bits
2. Les serveurs WebSphere eXtreme Scale ne peuvent cohabiter avec WebSphere Application Server DMgr.
3. Real Time ne prend pas en charge DMgr.
4. Real Time ne prend pas en charge les agents de noeuds WebSphere.

Activer WebSphere Real Time

Installez WebSphere Real Time et WebSphere eXtreme Scale sur les ordinateurs sur lesquels vous prévoyez d'exécuter eXtreme Scale. Mettez au niveau SR2 le Java de WebSphere Real Time.

Vous pouvez spécifier comme suit les paramètres des machines virtuelles Java pour chaque serveur via la console WebSphere Application Server version 7.0.

Sélectionnez **Serveurs** > **Types de serveur** > **Serveurs d'applications WebSphere** > **<serveur installé requis>**.

Dans la page qui s'affiche, choisissez Définition des processus.

Dans la page qui s'affiche alors, cliquez sur Machine virtuelle Java en haut de la colonne de droite (c'est là que vous pouvez définir pour chaque serveur la taille des segments, la récupération de place et d'autres indicateurs).

Définissez les indicateurs suivants dans la zone Arguments JVM génériques :

```
-Xrealtime -Xgcpolicy:metronome  -Xnocompressedrefs -Xgc:targetUtilization=80
```

Appliquez les modifications et enregistrez-les.

Pour utiliser Real Time dans WebSphere Application Server 7.0 avec des serveurs eXtreme Scale incluant les indicateurs JVM ci-dessus, vous devez créer une variable d'environnement JAVA_HOME.

Définissez JAVA_HOME comme suit :

1. Développez Environnement.
2. Sélectionnez Variables WebSphere.
3. La case Toutes les portées en dessous de Afficher la portée doit être cochée.
4. Sélectionnez le serveur requis dans la liste déroulante (ne sélectionnez pas de serveurs DMgr ou d'agents de noeuds).
5. Si la variable d'environnement JAVA_HOME n'apparaît pas dans la liste, sélectionnez Nouveau et spécifiez JAVA_HOME comme nom de la variable. Dans la zone Valeur, entrez le nom complet du chemin d'accès à Real Time.
6. Appliquez les modifications et enregistrez-les.

Pratiques recommandées

Vous trouverez un ensemble de pratiques recommandées dans la section Pratiques recommandées du chapitre «Optimisation de la récupération de place avec WebSphere Real Time», à la page 501. Il y a dans cette liste de pratiques recommandées pour un environnement WebSphere eXtreme Scale autonome des points qui diffèrent pour un déploiement dans un environnement WebSphere Application Server Network Deployment.

Vous devez placer des paramètres supplémentaires de ligne de commande JVM au même endroit que les paramètres de stratégie de récupération de place évoqués à la précédente section.

Une cible initiale acceptable pour les charges processeur soutenues est de 50 % avec des charges de pointe de courte durée grimpant jusqu'à 75 %. Au-delà, vous devez ajouter des capacités supplémentaires avant de constater une dégradation mesure de la prévisibilité et de la cohérence. Vous pouvez augmenter légèrement les performances si vous pouvez tolérer des temps de réponse plus longs. Au-delà d'un seuil de 80 % conduit souvent à une dégradation significative de la cohérence et de la prévisibilité.

Optimisation du fournisseur de cache dynamique

Le fournisseur de cache dynamique de WebSphere eXtreme Scale prend en charge les paramètres de configuration ci-après pour l'optimisation des performances.

Pourquoi et quand exécuter cette tâche

- **com.ibm.websphere.xs.dynacache.ignore_value_in_change_event** : lorsque vous enregistrez un programme d'écoute d'événements de modification auprès du fournisseur de cache dynamique et que vous générez une instance ChangeEvent, une charge supplémentaire est associée à la désérialisation de l'entrée de cache pour que la valeur puisse être placée dans l'événement de modification. Si vous affectez la valeur `true` à ce paramètre facultatif dans l'instance de cache, la désérialisation de l'entrée de cache est ignorée lors de la génération des événements de modification. La valeur retournée est null pour une opération de suppression ou un tableau d'octets contenant la forme sérialisée de l'objet. Les instances InvalidationEvent font l'objet d'une pénalité de performances similaire, que vous pouvez éviter en affectant à com.ibm.ws.cache.CacheConfig.ignoreValueInInvalidationEvent la valeur `true`.
- **com.ibm.websphere.xs.dynacache.enable_compression** : par défaut, le fournisseur de cache dynamique eXtreme Scale comprime les entrées de cache en mémoire pour augmenter la densité du cache, ce qui peut permettre d'économiser une quantité de mémoire significative pour les applications telles

que la mise à en cache de servlet. Si vous savez que la majorité des données de cache ne peuvent pas être compressées, définissez la valeur `false`.

Chapitre 10. Sécurité

WebSphere eXtreme Scale permet de sécuriser l'accès aux données et l'intégration de fournisseurs de sécurité externes. Les éléments de sécurité comprennent l'authentification, l'autorisation, la sécurité du transport, la sécurité de la grille de données, la sécurité locale et la sécurité JMX (MBean).

Authentification du client d'application

L'authentification du client d'application consiste à activer la sécurité client-serveur et l'authentification des données d'identification et à configurer un authentificateur et un générateur de données d'identification.

Activation de la sécurité client-serveur

Vous devez activer la sécurité sur le client et sur le serveur pour pouvoir vous authentifier auprès de la grille d'objets.

Activation de la sécurité du client

WebSphere eXtreme Scale fournit un exemple de fichier de propriétés client, le fichier `sampleClient.properties`, dans le répertoire *racine_was*`/optionalLibraries/ObjectGrid/properties` pour une installation WebSphere Application Server ou le répertoire `/ObjectGrid/properties` dans une installation de serveurs mixtes. Vous pouvez modifier ce fichier en y entrant les valeurs de votre choix. Associez la propriété securityEnabled du fichier `objectgridClient.properties` à la valeur `true`. La propriété securityEnabled indique si la sécurité est activée. Lorsqu'un client se connecte à un serveur, les valeurs du côté client et du côté serveur doivent toutes deux être égales à `true` ou à `false`. Par exemple, si la sécurité du serveur connecté est activée, la valeur de la propriété doit être associée à `true` du côté client pour que le client puisse se connecter au serveur.

L'interface com.ibm.websphere.objectgrid.security.config.ClientSecurityConfiguration représente le fichier `security.ogclient.props`. Vous pouvez utiliser l'API publique com.ibm.websphere.objectgrid.security.config.ClientSecurityConfigurationFactory pour créer une instance de cette interface avec les valeurs par défaut ou vous pouvez créer une instance en transmettant le fichier des propriétés de sécurité du client ObjectGrid. Le fichier `security.ogclient.props` contient d'autres propriétés. Pour plus de détails, voir la documentation de l'API ClientSecurityConfiguration et celle de l'API ClientSecurityConfigurationFactory.

Activation de la sécurité du serveur

Pour activer la sécurité côté serveur, vous pouvez donner la valeur `true` à la propriété **`securityEnabled`** dans le fichier `security.xml`. Un fichier XML de descripteur de sécurité vous permettra de spécifier la configuration de la sécurité de la grille en isolant celle-ci de tous les éléments de configuration n'ayant pas trait à la sécurité.

Activation de l'authentification des données d'identification

Une fois que le client eXtreme Scale a extrait l'objet Credential en utilisant l'objet CredentialGenerator, l'objet Credential est envoyé avec la demande du client au serveur eXtreme Scale. Le serveur authentifie l'objet Credential avant de traiter la demande. Si l'authentification de l'objet Credential réussit, un objet Subject est renvoyé pour représenter cet objet Credential. Cet objet Subject est alors utilisé pour autoriser la demande.

Définissez la propriété **credentialAuthentication** dans les fichiers de propriétés du client et du serveur afin d'activer l'authentification des données d'identification. Pour plus d'informations, voir Fichier de propriétés du client et Fichier de propriétés du serveur.

Le tableau suivant présente les mécanismes d'authentification à utiliser selon les paramètres.

Tableau 31. Authentification des données d'identification dans les paramètres du client et du serveur

Authentification des données d'identification du client	Authentification des données d'identification du serveur	Résultat
Non	Jamais	Désactivée
Non	Prise en charge	Désactivée
Non	Obligatoire	Cas d'erreur
Prise en charge	Jamais	Désactivée
Prise en charge	Prise en charge	Activé
Prise en charge	Obligatoire	Activé
Obligatoire	Jamais	Cas d'erreur
Obligatoire	Prise en charge	Activé
Obligatoire	Obligatoire	Activé

Configuration d'un authentificateur

Le serveur eXtreme Scale utilise le plug-in Authenticator pour authentifier l'objet Credential. Une implémentation de l'interface Authenticator obtient l'objet Credential qu'elle authentifie ensuite auprès d'un registre d'utilisateurs, un serveur LDAP, par exemple, et ainsi de suite. eXtreme Scale ne fournit aucune configuration de registre. La connexion à un registre d'utilisateurs et l'authentification auprès de celui-ci doivent être implémentées dans ce plug-in.

Par exemple, une implémentation d'Authenticator extrait l'ID utilisateur et le mot de passe des données d'identification, les utilise pour la connexion et la validation auprès d'un serveur LDAP et crée un objet Subject résultant de l'authentification. L'implémentation peut utiliser les modules de connexion JAAS (Java Authentication and Authorization Service). Un objet Subject est retourné comme résultat de l'authentification.

Vous pouvez configurer l'authentificateur dans le fichier XML descripteur de sécurité, comme dans l'exemple qui suit :

```
<?xml version="1.0" encoding="UTF-8"?>
<securityConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config/security ../objectGridSecurity.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config/security">
```

```
 <security securityEnabled="true"
        loginSessionExpirationTime="300">

  <authenticator className ="com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginAuthenticator">
        </authenticator>

 </security>

</securityConfig>
```

L'option **-clusterSecurityFile** lors du démarrage d'un serveur sécurisé permet de définir le fichier XML de sécurité. Voir le tutoriel de sécurité Java SE dans *Présentation du produit* pour plus d'informations.

Configuration d'un générateur de données d'identification système

Le générateur de données d'identification système permet de représenter la fabrique des données d'identification système. Les données d'identification système sont identiques aux données d'identification de l'administrateur. Vous pouvez configurer l'élément SystemCredentialGenerator dans le fichier XML de sécurité du catalogue, comme indiqué dans cet exemple :

```
<systemCredentialGenerator className ="com.ibm.websphere.objectgrid.security.plugins.
 builtins.UserPasswordCredentialGenerator">
          <property name="properties" type="java.lang.String" value="manager manager1"
      description="username password" />
 </systemCredentialGenerator>
```

Pour que cet exemple soit parlant, le nom d'utilisateur et le mot de passe sont indiqués en clair. Dans un environnement de production, ne stockez pas ces informations en clair.

WebSphere eXtreme Scale fournit un générateur de données d'identification système par défaut qui utilise les données d'identification du serveur. Si vous n'indiquez pas ce générateur de façon explicite, le générateur par défaut est utilisé.

Autorisation du client d'application

L'autorisation du client d'application consiste en des classes d'autorisation ObjectGrid, des mécanismes d'autorisation, une période de vérification des droits et une autorisation "accès réservé au créateur".

Pour eXtreme Scale, l'autorisation est accordée en fonction de l'objet et des droits du sujet. Le produit prend en charge deux sortes de mécanismes d'autorisation : le service JAAS (Java Authentication and Authorization Service) et l'autorisation personnalisée.

Classes d'autorisation ObjectGrid

L'autorisation est accordée en fonction des droits. Il existe quatre types de classes d'autorisation :

- La classe MapPermission représente les autorisations d'accès aux données dans les mappes ObjectGrid.
- La classe ObjectGridPermission représente les autorisations d'accès à ObjectGrid.
- La classe ServerMapPermission représente les autorisations d'accès aux mappes ObjectGrid sur le serveur à partir d'un client.
- La classe AgentPermission représente les autorisations de démarrage d'un agent sur le serveur.

Pour plus d'informations sur les API et les autorisations associées, voir la rubrique sur la programmation de l'autorisation client dans le *Guide de programmation*.

Période de vérification des droits

eXtreme Scale prend en charge la mise en cache des résultats de la vérification des droits d'accès aux mappes pour des raisons de performances. Sans ce mécanisme, lorsqu'une méthode, qui est répertoriée dans la liste des méthodes de votre classe d'autorisation, est appelée, l'environnement d'exécution appelle le mécanisme d'autorisation configuré pour autoriser l'accès. Lorsque cette période est définie, le mécanisme d'autorisation est appelé périodiquement. Pour la liste des méthodes de chaque classe d'autorisation, voir la rubrique sur la programmation d'autorisation client dans *Guide de programmation*.

Les informations relatives à l'autorisation des droits se fondent sur l'objet Subject. Lorsqu'un client essaie d'accéder aux méthodes, l'environnement d'exécution eXtreme Scale recherche l'objet Subject dans le cache. Si cet objet est introuvable, l'environnement d'exécution vérifie les droits qui lui sont accordés, puis stocke les droits dans un cache.

La période de vérification des droits doit être définie avant l'initialisation de la grille d'objets. Vous pouvez la configurer de deux manières différentes :

Vous pouvez utiliser le fichier XML ObjectGrid pour définir une grille d'objets ainsi que la période de vérification des droits. Dans l'exemple suivant, cette période est définie pour une durée de 45 secondes :

```
<objectGrids>
 <objectGrid name="secureClusterObjectGrid" securityEnabled="true"
 authorizationMechanism="AUTHORIZATION_MECHANISM_JAAS"
 permissionCheckPeriod="45">
  <bean id="bean id="TransactionCallback"
className="com.ibm.websphere.samples.objectgrid.HeapTransactionCallback" />
...
</objectGrids>
```

Si vous souhaitez créer une grille d'objets à l'aide des API, appelez la méthode suivante pour définir la période de vérification des droits. Cette méthode peut uniquement être appelée avant l'initialisation de l'instance ObjectGrid. Elle s'applique uniquement au modèle de programmation eXtreme Scale local lorsque vous instanciez directement une instance ObjectGrid.

```
/**
 * This method takes a single parameter indicating how often you
 * want to check the permission used to allow a client access. If the
 * parameter is 0 then every single get/put/update/remove/evict call
 * asks the authorization mechanism, either JAAS authorization or custom
 * authorization, to check if the current subject has permission. This might be
 * prohibitively expensive from a performance point of view depending on
 * the authorization implementation, but if you need to have ever call check the
 * authorization mechanism, then set the parameter to 0.
 * Alternatively, if the parameter is > 0 then it indicates the number
 * of seconds to cache a set of permissions before returning to
 * the authorization mechanism to refresh them. This value provides much
 * better performance, but if the back-end
 * permissions are changed during this time then the ObjectGrid can
 * allow or prevent access even though the back-end security
 * provider was modified.
 *
 * @param period the permission check period in seconds.
 */
void setPermissionCheckPeriod(int period);
```

Autorisation "accès réservé au créateur"

Avec l'autorisation "accès réservé au créateur", seul l'utilisateur (représenté par les objets Principal qui lui sont associés) ayant inséré une entrée dans une mappe ObjectGrid peut accéder (lecture, mise à jour, invalidation et suppression) à cette entrée.

Le modèle d'autorisation d'accès aux mappes ObjectGrid existant se fonde sur le type d'accès et non sur les entrées de données. En d'autres termes, un utilisateur dispose de droits d'accès d'un certain type (par exemple lecture, écriture, insertion, suppression ou invalidation) à toutes les données de la mappe ou ne détient aucun droit d'accès à aucune donnée. En revanche, eXtreme Scale n'autorise pas l'accès à certaines données seulement. Cette fonction constitue une nouvelle manière d'octroyer aux utilisateurs des droits d'accès aux entrées de données.

Dans un scénario où différents utilisateurs accèdent à différents jeux de données, ce modèle peut être utile. Lorsqu'un utilisateur charge des données à partir du stockage de persistance dans les mappes ObjectGrid, l'accès peut être autorisé par le stockage de persistance. Dans ce cas, il est inutile d'accorder une autre autorisation dans la couche de mappes ObjectGrid. Vous devez seulement faire en sorte que la personne qui charge les données dans la mappe peut y accéder en activant la fonction "réservé au créateur".

Valeurs des attributs en mode Réservé au créateur :

disabled
La fonction "accès réservé au créateur" est désactivée.

complement
La fonction "accès réservé au créateur" est activée et vient s'ajouter à l'autorisation d'accès aux mappes. En d'autres termes, les deux fonctions (autorisation d'accès aux mappes et fonction "accès réservé au créateur") sont opérationnelles. Vous pouvez donc limiter les opérations aux données. Le créateur ne peut par exemple pas invalider les données.

supersede
La fonction "accès réservé au créateur" est activée et remplace l'autorisation d'accès aux mappes. En d'autres termes, elle se substitue à cette autorisation, qui n'est plus opérationnelle.

Vous pouvez configurer le mode "accès réservé au créateur" de deux manières :

Avec un fichier XML :

Vous pouvez utiliser le fichier XML ObjectGrid pour définir une grille d'objets et choisir le mode `disabled`, `complement` ou `supersede`, comme dans l'exemple suivant :

```
<objectGrids>
    <objectGrid name="secureClusterObjectGrid" securityEnabled="true"
         accessByCreatorOnlyMode="supersede"
        <bean id="TransactionCallback"
              classname="com.ibm.websphere.samples.objectgrid.HeapTransactionCallback" />
    ...
</objectGrids>
```

A l'aide d'un programme :

Si vous souhaitez créer une grille d'objets à l'aide d'un programme, vous pouvez appeler la méthode suivante pour définir le mode "accès réservé au créateur".

L'appel de cette méthode s'applique uniquement au modèle de programmation eXtreme Scale local lorsque vous instanciez directement l'instance ObjectGrid :

```
/**
 * Set the "access by creator only" mode.
 * Enabling "access by creator only" mode ensures that only the user (represented
 * by the Principals associated with it), who inserts the record into the map,
 * can access (read, update, invalidate, and remove) the record.
 * The "access by creator only" mode can be disabled, or can complement the
 * ObjectGrid authorization model, or it can supersede the ObjectGrid
 * authorization model. The default value is disabled:
 * {@link SecurityConstants#ACCESS_BY_CREATOR_ONLY_DISABLED}.
 * @see SecurityConstants#ACCESS_BY_CREATOR_ONLY_DISABLED
 * @see SecurityConstants#ACCESS_BY_CREATOR_ONLY_COMPLEMENT
 * @see SecurityConstants#ACCESS_BY_CREATOR_ONLY_SUPERSEDE
 *
 * @param accessByCreatorOnlyMode the access by creator mode.
 *
 * @since WAS XD 6.1 FIX3
*/
void setAccessByCreatorOnlyMode(int accessByCreatorOnlyMode);
```

Autre exemple : imaginez un scénario selon lequel une grille de données bancaires contient un compte de mappe ObjectGrid dont les deux utilisateurs sont Manager1 et Employee1. Les règles d'autorisation d'eXtreme Scale accordent tous les droits d'accès à Manager1, mais uniquement les droits d'accès en lecture à Employee1. Les règles JAAS d'autorisation d'accès aux mappes ObjectGrid sont représentées ci-dessous :

```
grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
    Principal com.acme.PrincipalImpl "Manager1" {
    permission com.ibm.websphere.objectgrid.security.MapPermission
        "banking.account", "all"
};
grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
    Principal com.acme.PrincipalImpl "Employee1" {
    permission com.ibm.websphere.objectgrid.security.MapPermission
        "banking.account", "read, insert"
};
```

Observez quelle incidence la fonction "accès réservé au créateur" a sur l'autorisation :

- **disabled** Si la fonction "accès réservé au créateur" est désactivée, l'autorisation d'accès aux mappes reste inchangée. L'utilisateur "Manager1" peut accéder à toutes les données de la mappe "account". L'utilisateur "Employee1" peut lire toutes les données de la mappe et en insérer, mais il ne peut ni mettre à jour, ni invalider, ni supprimer ces données.
- **complement** Si la fonction "accès réservé au créateur" est activée avec l'option "complement", les deux fonctions sont opérationnelles. L'utilisateur "Manager1" peut accéder aux données de la mappe "account", mais uniquement s'il les a chargées dans celle-ci. L'utilisateur "Employee1" peut lire les données de cette mappe, mais uniquement s'il les a chargées. (Il ne peut cependant ni mettre à jour, ni invalider, ni supprimer les données de cette mappe.)
- **supersede** Si la fonction "accès réservé au créateur" est activée avec l'option "supersede", l'autorisation d'accès aux mappes n'est pas activée. L'autorisation "accès réservé au créateur" est alors la seule règle en vigueur. L'utilisateur "Manager1" détient les mêmes privilèges que ceux liés au mode "complement" : il peut accéder aux données de la mappe "account" uniquement s'il les a chargées dans la mappe. Toutefois, l'utilisateur "Employee1" détient maintenant les droits d'accès complet aux données de la mappe "account" s'il les a chargés dans la mappe. En d'autres termes, les règles d'autorisation définies dans les règles Java Authentication and Authorization Service (JAAS) ne sont pas appliquées.

Authentification d'une grille de données

Vous pouvez utiliser le plug-in du gestionnaire de jetons de sécurité pour activer l'authentification serveur à serveur, ce qui signifie d'implémenter l'interface SecureTokenManager.

La méthode generateToken(Object) prend un objet protect, puis génère un jeton pouvant être compris par les autres. La méthode verifyTokens(byte[]) procède en sens contraire : elle reconvertit le jeton en objet d'origine.

Une implémentation SecureTokenManager simple utilise un algorithme de codage de base, tel qu'un algorithme XOR, pour coder l'objet dans un formulaire sérialisé et utiliser l'algorithme de décodage correspondant pour décoder le jeton. Cette implémentation n'est pas sécurisée et peut être facilement interrompue.

Implémentation par défaut de WebSphere eXtreme Scale

WebSphere eXtreme Scale met immédiatement à disposition une implémentation de cette interface. Cette implémentation par défaut utilise une paire de clés pour signer et vérifier la signature et une clé confidentielle pour chiffrer le contenu. Chaque serveur comporte un fichier de clés de type JCKES contenant la paire de clés, une clé privée et une clé publique ainsi qu'une clé confidentielle. Pour stocker les clés confidentielles, le fichier de clés doit être de type JCKES. En effet, ces clés servent à chiffrer et signer ou vérifier la chaîne secrète côté expéditeur. De plus, le jeton est associé à un délai d'expiration. Côté récepteur, les données sont vérifiées, déchiffrées et comparées à la chaîne secrète du récepteur. Des protocoles de communication SSL (Secure Sockets Layer) ne sont pas requis entre une paire de serveurs pour l'authentification car les clés privées et les clés publiques ont la même finalité. Toutefois, si la communication du serveur n'est pas chiffrée, les données peuvent être dérobées à la seule vue de la communication. Le jeton venant à expiration, la menace d'attaque par relecture est minimisée. Ce risque est considérablement réduit si tous les serveurs sont déployés derrière un pare-feu.

L'inconvénient de cette approche : les administrateurs de WebSphere eXtreme Scale doivent générer des clés et les transporter vers tous les serveurs, ce qui peut provoquer des failles de sécurité lors du transport.

Sécurité de grille de données

La sécurité d'une grille de données permet de garantir qu'un serveur qui y est ajouté dispose des données d'identifications correctes afin qu'elles ne contiennent aucun serveur malveillant. La sécurité de la grille de données utilise un mécanisme de chaîne secrète partagée.

Tous les serveurs WebSphere eXtreme Scale, y compris les serveurs de catalogues, choisissent une chaîne de secret partagé commune. Lorsqu'un serveur rejoint la grille de données, il est invité à présenter la chaîne secrète. Si cette chaîne correspond à celle du serveur président ou du serveur de catalogues, le serveur est accepté. Dans le cas contraire, la demande de jointure est rejetée.

L'envoi d'un texte en clair n'est pas sécurisé. L'infrastructure de sécurité WebSphere eXtreme Scale propose un plug-in de gestionnaire de jetons sécurisé permettant au serveur de sécuriser la valeur confidentielle avant son envoi. Vous devez choisir le mode d'implémentation de cette opération. WebSphere eXtreme Scale propose une implémentation prête à l'emploi : l'opération de sécurité est implémentée pour chiffrer et signer la valeur confidentielle.

La chaîne confidentielle est définie dans le fichier `server.properties`. Reportez-vous à la rubrique Fichier de propriétés du serveur pour plus d'informations sur la propriété authenticationSecret.

Plug-in SecureTokenManager

Le plug-in de gestionnaire de jetons sécurisé est représenté par l'interface com.ibm.websphere.objectgrid.security.plugins.SecureTokenManager.

Pour plus d'informations sur ce plug-in, consultez la documentation relative à l'API SecureTokenManager.

La méthode generateToken(Object) prend un objet, puis génère un jeton ne pouvant être compris par les autres. La méthode verifyTokens(byte[]) suit le processus inverse : la méthode reconvertit le jeton à son format d'origine.

Une implémentation SecureTokenManager simple utilise un algorithme de codage simple, par exemple un algorithme OU exclusif (XOR), pour coder l'objet au format sérialisé, puis utilise l'algorithme de décodage pour décoder le jeton. Cette implémentation n'est pas sécurisée.

WebSphere eXtreme Scale propose une implémentation disponible immédiatement pour cette interface.

L'implémentation par défaut utilise une paire de clés pour signer et vérifier la signature et utilise une clé confidentielle pour en chiffrer le contenu. Chaque serveur a un fichier de clés de type JCKES pour stocker cette paire (une clé privée et une clé publique), ainsi qu'une clé confidentielle. Le fichier de clés doit être de type JCKES pour pouvoir stocker les clés confidentielles.

Ces clés sont utilisées pour chiffrer et signer ou vérifier la chaîne secrète côté envoi. Le jeton est associé à un délai d'expiration. Côté réception, les données sont vérifiées, déchiffrées et comparées à la chaîne secrète du récepteur. Les protocoles de communication SSL (Secure Sockets Layer) ne sont pas obligatoires pour l'authentification entre une paire de serveurs car les clés privées et les clés publiques ont les mêmes fonctions. Toutefois, si la communication avec les serveurs n'est pas chiffrée, les données peuvent être volées en regardant la communication. Le jeton expirant bientôt, la menace pesant sur les attaques de type replay est minime. Cette possibilité décroît même de manière significative lorsque tous les serveurs sont déployés derrière un pare-feu.

L'inconvénient de cette approche est que les administrateurs WebSphere eXtreme Scale doivent générer des clés et en assurer le transport vers tous les serveurs, au cours duquel la sécurité risque d'être enfreinte.

Exemples de scripts permettant de créer les propriétés du gestionnaire de jetons par défaut

Comme indiqué dans la section précédente, vous pouvez créer un fichier de clés contenant une paire de clés pour signer et vérifier la signature, ainsi qu'une clé secrète pour chiffrer le contenu.

Vous pouvez par exemple utiliser la commande de l'outil de clé JDK 6 pour créer les clés, comme ci-dessous :

```
keytool -genkeypair -alias keypair1 -keystore key1.jck -storetype JCEKS -keyalg
rsa -dname "CN=sample.ibm.com, OU=WebSphere eXtreme Scale" -storepass key111 -keypass
keypair1 -validity 10000

keytool -genseckey -alias seckey1 -keystore key1.jck -storetype JCEKS -keyalg
DES  -storepass key111 -keypass seckey1 -validity 1000
```

Ces deux commandes créent une paire de clés "keypair1" et une clé confidentielle "seckey1". Vous pouvez alors configurer les éléments suivants dans le fichier de propriétés du serveur :

```
secureTokenKeyStore=key1.jck
secureTokenKeyStorePassword=key111
secureTokenKeyStoreType=JCEKS
secureTokenKeyPairAlias=keypair1
secureTokenKeyPairPassword=keypair1
secureTokenSecretKeyAlias=seckey1
secureTokenSecretKeyPassword=seckey1
secureTokenCipherAlgorithm=DES
secureTokenSignAlgorithm=RSA
```

Configuration

Consultez la rubrique Propriétés du serveur pour plus d'informations sur les propriétés utilisées pour configurer le gestionnaire de jetons sécurisé.

Protocole TLS et couche de connexion sécurisée

WebSphere eXtreme Scale prend en charge les protocoles TCP/IP et TLS/SSL pour assurer une communication sécurisée entre les clients et les serveurs.

Activation de TLS/SSL dans les deux sens

TLS/SSL est parfois activé dans un seul sens. Par exemple, le certificat public du serveur est importé dans le fichier de clés certifiées, mais pas le certificat public du client n'est pas importé vers le fichier de clés certifiées du serveur. Toutefois, WebSphere eXtreme Scale utilise largement des agents de grille de données. Un agent de grille de données se caractérise par le fait qu'il crée un connexion lorsque le serveur répond au client. Le serveur eXtreme Scale fait alors office de client. Par conséquent, vous devez importer le certificat public du client dans le fichier de clés certifiées du serveur.

Activation de la sécurité de transport pour kit JDK Sun

WebSphere eXtreme Scale nécessite IBM Java Secure Sockets Extension (IBMJSSE) ou IBM Java Secure Sockets Extension 2 (IBMJSSE2). Les fournisseurs IBMJSSE et IBMJSSE2 contiennent une implémentation de référence prenant en charge les protocoles SSL et TLS, ainsi qu'un framework d'API.

Le kit JDK Sun ne fournit pas les fournisseurs IBM JSSE et IBM JSSE2 et la sécurité de transport ne peut donc pas être activée avec un kit JDK Sun. A cet effet, un kit JDK Sun fourni avec WebSphere Application Server est nécessaire. Le JDK Sun WebSphere Application Server fourni contient les fournisseurs IBM JSSE et IBM JSSE2.

Voir «Configuration d'un ORB personnalisé», à la page 291 pour plus d'informations sur l'utilisation d'un JDK non-IBM pour WebSphere eXtreme Scale. Si `-Djava.endorsed.dirs` est configuré, il pointe vers les répertoires `objectgridRoot/lib/endorsed` et `JRE/lib/endorsed`. Le répertoire

`objectgridRoot/lib/endorsed` est requis de sorte que la fonction ORB IBM est utilisée et le répertoire `JRE/lib/endorsed` est requis pour le chargement des fournisseurs JSSE IBM et JSSE IBM.

Etudiez l'étape 4 du tutoriel de sécurité dans *Présentation du produit* pour plus d'informations sur la définition des propriétés SSL requises, créer des fichiers de clés et de clés certifiées et pour démarrer des serveurs sécurisés dans WebSphere eXtreme Scale.

Configuration des types de transports sécurisés

TLS (Transport layer security) fournit une communication sécurisée entre le client et le serveur. Le mécanisme de communication utilisé dépend de la valeur du paramètre **transportType** spécifié dans les fichiers de configuration du client et du serveur.

Pourquoi et quand exécuter cette tâche

Lorsque vous utilisez SSL (Secure Sockets Layer), les paramètres de configuration SSL doivent être définis dans le client et le serveur. Dans un environnement Java SE, la configuration du protocole SSL s'effectue dans les fichiers de propriétés du client ou du serveur. Si le client ou le serveur se trouve dans WebSphere Application Server, vous pouvez utiliser les paramètres de transport CSIV2 existants WebSphere Application Server de vos serveurs de conteneur et de vos clients. Pour plus d'informations, voir «Intégration de la sécurité dans WebSphere Application Server», à la page 525.

Tableau 32. Protocole de transport à utiliser avec les paramètres de transport client et serveur.

Si les paramètres transportType sont différents entre le client et le serveur, le protocole résultant peut varier ou entraîner une erreur.

Propriété de client transportType	Propriété de serveur transportType	Résultat du protocole
TCP/IP	TCP/IP	TCP/IP
TCP/IP	SSL pris en charge	TCP/IP
TCP/IP	SSL requis	Erreur
SSL pris en charge	TCP/IP	TCP/IP
SSL pris en charge	SSL pris en charge	SSL (en cas d'échec du protocole SSL, TCP/IP)
SSL pris en charge	SSL requis	SSL
SSL requis	TCP/IP	Erreur
SSL requis	SSL pris en charge	SSL
SSL requis	SSL requis	SSL

Procédure

1. Pour définir la propriété **transportType** dans la configuration de sécurité du client, voir Fichier de propriétés du client.
2. Pour définir la propriété **transportType** dans la configuration de sécurité du conteneur et du serveur de catalogue, voir Fichier de propriétés du serveur.

Définition des paramètres SSL (Secure Sockets Layer) des clients ou des serveurs

La manière de définir les paramètres SSL est différente pour les clients et les serveurs.

Pourquoi et quand exécuter cette tâche

TLS/SSL est parfois activé dans un seul sens. Par exemple, le certificat public du serveur est importé dans le fichier de clés certifiées, mais pas le certificat public du client n'est pas importé vers le fichier de clés certifiées du serveur. Toutefois, WebSphere eXtreme Scale utilise largement des agents de grille de données. Un agent de grille de données se caractérise par le fait qu'il crée une connexion lorsque le serveur répond au client. Le serveur eXtreme Scale fait alors office de client. Par conséquent, vous devez importer le certificat public du client dans le fichier de clés certifiées du serveur.

Procédure

- Définissez les paramètres SSL du client.

  Utilisez l'une des options suivantes pour définir les paramètres SSL sur le client :

  – Créez un objet com.ibm.websphere.objectgrid.security.config.SSLConfiguration à l'aide de la classe com.ibm.websphere.objectgrid.security.config.ClientSecurityConfigurationFactory.

  – Configurez les paramètres dans le fichier `client.properties`. Vous pouvez ensuite définir le fichier de propriétés comme propriété de client JVM ou utiliser les API WebSphere eXtreme Scale. Transmettez le fichier de propriétés dans la méthode ClientSecurityConfigurationFactory.getClientSecurityConfiguration(String) du client et utilisez l'objet retourné comme paramètre dans la méthode ObjectGridManager.connect(String, ClientSecurityConfiguration, URL).

- Configurez les paramètres SSL du serveur.

  Les paramètres SSL sont configurés pour les serveurs qui utilisent le fichier `server.properties`. Pour démarrer un serveur de conteneur ou de catalogue avec un fichier de propriétés spécifique, utilisez le paramètre **-serverProps** dans le script **startOgServer**. Pour plus d'informations sur les paramètres SSL que vous pouvez définir pour les serveurs eXtreme Scale, voir Propriétés du serveur de sécurité.

Sécurité JMX (Java Management Extensions)

Vous pouvez sécuriser les invocations de beans gérés (MBean) dans un environnement réparti.

Pour plus d'informations sur les beans gérés disponibles, voir «Administration avec les beans gérés (MBeans)», à la page 434.

Dans une topologie de déploiement réparti, les beans gérés sont directement hébergés sur les serveurs de catalogues et les serveurs conteneurs. En général, la sécurité JMX dans une topologie répartie suit la spécification de sécurité JMX telle que spécifiée dans la spécification JMX (Java Management Extensions). Elle est composée des trois parties suivantes :

1. Authentification : le client distant doit être authentifié dans le serveur de connecteur.
2. Contrôle d'accès : le contrôle de l'accès des beans gérés définit les privilèges d'accès aux informations de beans gérés et les droits d'exécution des opérations de beans gérés.
3. Transfert sécurisé : le transfert entre le client et le serveur JMX peut être sécurisé à l'aide du protocole TLS/SSL.

Authentification

JMX offre des méthodes aux serveurs de connecteur pour authentifier les clients distants. Pour le connecteur RMI, l'authentification est effectuée en fournissant un objet qui implémente l'interface JMXAuthenticator lors de la création du serveur de connecteur. Par conséquent, eXtreme Scale implémente cette interface JMXAuthenticator à utiliser le plug-in ObjectGrid Authenticator pour authentifier les clients distants. Voir «Tutoriel sur la sécurité Java SE - Etape 2», à la page 73Tutoriel sur la sécurité dans *Présentation du produit* pour obtenir des détails sur le mode d'authentification d'un client par eXtreme Scale.

Le client JMX suit les API JMX pour offrir des données d'identification permettant la connexion au serveur de connecteur. L'infrastructure JMX transmet les données d'identification au serveur de connecteur et appelle l'implémentation JMXAuthenticator pour l'authentification. Tel que décrit précédemment, l'implémentation JMXAuthenticator délègue ensuite l'authentification à l'implémentation de l'authentificateur ObjectGrid.

Passez en revue l'exemple présenté ci-dessous, qui décrit comment établir la connexion à un serveur de connecteur à l'aide de données d'identification :

```
javax.management.remote.JMXServiceURL jmxUrl = new JMXServiceURL(
        "service:jmx:rmi:///jndi/rmi://localhost:1099/objectgrid/MBeanServer");

    environment.put(JMXConnector.CREDENTIALS, new UserPasswordCredential("admin", "xxxxxx"));

    // Créez le JMXCconnectorServer
    JMXConnector cntor = JMXConnectorFactory.newJMXConnector(jmxUrl, null);

    // Connectez et appelez une opération sur le MBeanServer distant
    cntor.connect(environment);
```

Dans l'exemple précédent, un objet UserPasswordCredential est fourni avec l'ID utilisateur `admin` et le mot de passe `xxxxx`. Cet objet UserPasswordCredential est défini dans la mappe d'environnement qui est utilisée dans la méthode JMXConnector.connect(Map). Cet objet UserPasswordCredential est ensuite transmis au serveur par l'infrastructure JMX, puis à l'infrastructure d'authentification ObjectGrid pour authentification.

Le modèle de programmation client respecte strictement les spécifications JMX.

Contrôle d'accès

Un serveur de beans gérés JMX peut avoir accès aux informations sensibles et peut être en mesure d'effectuer des opérations sensibles. JMX offre le contrôle d'accès requis permettant d'identifier les clients pouvant accéder à telles ou telles informations et qui peut effectuer ces opérations. Le contrôle d'accès repose sur le modèle de sécurité Java standard en définissant des autorisations de contrôle d'accès au serveur de beans gérés et aux opérations correspondantes.

Pour le contrôle d'accès ou l'autorisation des opérations JMX, eXtreme Scale repose sur le support JAAS fourni par l'implémentation JMX. A n'importe quel stade de l'exécution d'un programme, il existe un ensemble d'autorisations maintenu dans une unité d'exécution. Lorsqu'une unité d'exécution appelle une opération de spécification JMX, celle-ci est connue sous le terme d'autorisation de maintien. Lorsqu'une opération JMX est effectuée, une vérification de sécurité est réalisée pour vérifier si l'autorisation requise est concernée par l'autorisation de maintien.

La définition des règles d'administration de beans gérés respecte le format de la stratégie Java. Par exemple, la stratégie suivante octroie à tous les signataires et

bases de code le droit d'extraire l'adresse JMX du serveur pour le PlacementServiceMBean, à l'exception du domaine com.ibm.websphere.objectgrid.

```
grant {
    permission javax.management.MBeanPermission
        "com.ibm.websphere.objectgrid.management.PlacementServiceMBean#retrieveServerJMXAddress
     [com.ibm.websphere.objectgrid:*,type=PlacementService]",
        "invoke";
}
```

Vous pouvez utiliser l'exemple de stratégie suivant pour compléter l'autorisation en fonction de l'identité du client distant. La stratégie octroie la même autorisation de bean géré que celle présentée dans l'exemple précédent, sauf pour les utilisateurs dont le nom X500Principal est CN=Administrator,OU=software,O=IBM,L=Rochester,ST=MN,C=US.

```
grant principal javax.security.auth.x500.X500Principal "CN=Administrator,OU=software,O=IBM,
    L=Rochester,ST=MN,C=US" {permission javax.management.MBeanPermission
        "com.ibm.websphere.objectgrid.management.PlacementServiceMBean#retrieveServerJMXAddress
    [com.ibm.websphere.objectgrid:*,type=PlacementService]",
        "invoke";
}
```

Les stratégies Java sont uniquement vérifiées si le gestionnaire de sécurité est activé. Démarrez les serveurs de catalogues et les serveurs conteneurs à l'aide de l'argument JVM -Djava.security.manager pour forcer le contrôle d'accès des opérations de beans gérés.

Transfert sécurisé

Le transfert entre le client et le serveur JMX peut être sécurisé à l'aide du protocole TLS/SSL. Si le type de transfert du serveur de catalogues ou du serveur conteneur est défini sur SSL_Required ou SSL_Supported, vous devez utiliser le protocole SSL pour établir la connexion au serveur JMX.

Pour utiliser le protocole SSL, vous devez configurer le fichier de clés certifiées, le type de fichier de clés certifiées et le mot de passe du fichier de clés certifiées sur le client MBean en utilisant les propriétés système -D :

1. -Djavax.net.ssl.trustStore=TRUST_STORE_LOCATION
2. -Djavax.net.ssl.trustStorePassword=TRUST_STORE_PASSWORD
3. -Djavax.net.ssl.trustStoreType=TRUST_STORE_TYPE

Si vous utilisez com.ibm.websphere.ssl.protocol.SSLSocketFactory comme fabrique de sockets SSL dans le fichier*rép_base_java*`/jre/lib/security/java.security`, utilisez les propriétés suivantes :

1. -Dcom.ibm.ssl.trustStore=TRUST_STORE_LOCATION
2. -Dcom.ibm.ssl.trustStorePassword=TRUST_STORE_PASSWORD
3. -Dcom.ibm.ssl.trustStoreType=TRUST_STORE_TYPE

Pour obtenir ces informations lorsque le protocole TLS/SSL (Transport Layer Security/Secure Sockets Layer) est activé, vous devez démarrer les serveurs de catalogue et de conteneur avec le port de service JMX défini. Pour le définir, vous pouvez utiliser l'option **-JMXServicePort** sur le script **startOgServer** ou appeler la méthode setJMXServicePort sur l'interface ServerProperties.

Pour activer le transport sécurisé JMX pour le serveur de conteneur, vous devez définir le port de service JMX. Vous devez définir le port de service JMX lorsque vous utilisez Transport Layer Security/Secure Sockets Layer (TLS/SSL) et que vous souhaitez afficher les informations du serveur de conteneur à partir du serveur de

catalogue. Par exemple, le port est requis lorsque vous utilisez la commande **xscmd -c showMapSizes**. Utilisez l'une des méthodes suivantes pour configurer le port de service JMX :

- Utilisez l'option **-JMXServicePort** sur le script **startOgServer**.
- Si vous utilisez un serveur embarqué, appelez la méthode setJMXServicePort dans l'interface ServerProperties pour définir le port de service JMX.

Vous devez utiliser un numéro de port différent pour chaque machine virtuelle Java dans votre configuration. Si vous souhaitez utiliser JMX/RMI, vous devez spécifier explicitement l'option **-JMXServicePort** et le numéro de port, même si vous souhaitez utiliser la valeur de port par défaut.

Intégration de la sécurité à des fournisseurs externes

Pour protéger vos données, le produit peut intégrer plusieurs fournisseurs de sécurité.

WebSphere eXtreme Scale peut intégrer un programme de sécurité externe. Ce programme externe doit fournir des services d'authentification et d'autorisation pour WebSphere eXtreme Scale. WebSphere eXtreme Scale dispose de points de plug-in pour intégrer une implémentation de sécurité.WebSphere eXtreme Scale a été intégré avec succès aux composants suivants :

- Protocole LDAP (Lightweight Directory Access Protocol)
- Kerberos
- Sécurité ObjectGrid
- Tivoli Access Manager
- Service JAAS (Java Authentication and Authorization Service)

eXtreme Scale utilise le fournisseur de sécurité pour les tâches suivantes :

- Authentifier les clients sur les serveurs.
- Autoriser les clients à accéder à certains artefacts eXtreme Scale ou à préciser ce qui peut être fait avec les artefacts eXtreme Scale.

eXtreme Scale propose les types d'autorisations suivants :

Autorisation de mappes
Les clients et les groupes peuvent être autorisés à insérer, lire, mettre à jour, expulser ou supprimer des opérations sur les mappes.

Autorisation ObjectGrid
Les clients ou les groupes peuvent être autorisés à effectuer des requêtes de type objet ou entité sur objectGrids.

Autorisation de l'agent DataGrid
Les clients ou les groupes peuvent être autorisés à permettre aux agents DataGrid d'être déployés en une base de données ObjectGrid.

Autorisation de mappes côté serveur
Les clients ou les groupes peuvent être autorisés à répliquer une mappe de serveur côté client ou à créer un index dynamique pour la mappe de serveur.

Autorisation d'administration
Les clients ou les groupes peuvent être autorisés à effectuer des tâches administratives.

Remarque : Si la sécurité est activée pour le dorsal, rappelez-vous que ces paramètres ne sont plus suffisants pour protéger vos données. Les paramètres de sécurité de votre base de données ou autre magasin de données ne sont transférés en aucune façon vers votre cache. Vous devez protéger séparément les données à présent mises en cache en utilisant le mécanisme de sécurité eXtreme Scale, qui inclut l'authentification, l'autorisation et la sécurité du niveau de transport.

Restriction : N'utilisez pas un kit de développement ou un environnement d'exécution 1.6 ou supérieur lorsque vous utilisez aussi la sécurité SSL (Transport Layer Security) avec une configuration autonome WebSphere eXtreme Scale. La version 1.6 et les versions suivantes ne sont pas compatibles avec les interfaces de programmation d'application WebSphere eXtreme Scale Version 7.1. Utilisez la version 1.5 ou une version antérieure pour les configurations requérant la sécurité de transport SSL pour les installations autonomes eXtreme Scale. Cette restriction s'applique uniquement lorsque vous utilisez la sécurité SSL dans une configuration autonome eXtreme Scale. La version 1.6 et les versions suivantes sont compatibles pour les configurations de transport non-SSL.

Sécurisation du service de données REST

Vous pouvez sécuriser un bon nombre d'aspects du service de données REST. L'accès au service de données REST d'eXtreme Scale peut être sécurisé via l'authentification et l'autorisation. Il peut également être contrôlé par des règles de configuration à portée de service, appelées règles d'accès. La sécurité des transports est le troisième élément concerné en matière de sécurisation.

Pourquoi et quand exécuter cette tâche

L'accès au service de données REST d'eXtreme Scale peut être sécurisé via l'authentification et l'autorisation. L'authentification et l'autorisation s'effectuent grâce à l'intégration à la sécurité d'eXtreme Scale.

L'accès peut également être contrôlé par des règles de configuration à portée service, appelées règles d'accès. Il existe deux types de règles d'accès : droits d'opérations du service qui contrôlent les opérations CRUD autorisées par le service et les droits d'accès aux entités qui contrôlent les opérations CRUD autorisées pour un type donné d'entité.

La sécurité des transports est fournie par la configuration du conteneur d'hébergement pour les connexions entre le client Web et le service REST. Et la sécurité du transport est fournie par la configuration client eXtreme Scale (pour le service REST aux connexions de grille eXtreme Scale).

Procédure

- Contrôlez l'authentification et l'autorisation.

  L'accès au service de données REST d'eXtreme Scale peut être sécurisé via l'authentification et l'autorisation. L'authentification et l'autorisation s'effectuent par l'intégration à la sécurité eXtreme Scale.

  Le service de données REST eXtreme Scale utilise la sécurité eXtreme Scale, pour l'authentification et l'autorisation, pour déterminer les utilisateurs qui peuvent accéder au service et les opérations qu'ils sont autorisés à effectuer via le service. Le service de données REST eXtreme Scale utilise soit des données d'identification globales configurées, avec utilisateur et mot de passe, soit des données d'identification dérivées d'une d'authentification HTTP BASIC qui est

envoyée avec chaque transaction à la grille de données eXtreme Scale où s'effectuent l'authentification et l'autorisation.

1. Configurez dans la grille l'authentification et l'autorisation des clients eXtreme Scale. Voir «Intégration de la sécurité à des fournisseurs externes», à la page 520 pour des explications détaillées sur la manière de configurer l'authentification et l'autorisation des clients eXtreme Scale.
2. Configurez le client eXtreme Scale, qui est utilisé par le service REST, pour la sécurité.

   Le service de données REST d'eXtreme Scale fait appel à la bibliothèque des clients eXtreme Scale lorsqu'il communique avec la grille eXtreme Scale. Il en résulte que le client eXtreme Scale doit être configuré pour la sécurité d'eXtreme Scale.

   L'authentification du client eXtreme Scale est activée via des propriétés dans le fichier des propriétés du client objectgrid. Au minimum, les attributs suivants doivent être activés lorsqu'on utilise la sécurité du client avec le service REST :

   ```
   securityEnabled=true
   credentialAuthentication=Supported [-ou-] Required
   credentialGeneratorProps=utilisateur:motdepasse [-ou-]
   {xor encoded utilisateur:motdepasse}
   ```

   A faire : L'utilisateur et le mot de passe spécifiés dans la propriété credentialGeneratorProps doivent correspondre à un ID du registre d'authentification et disposer de droits de règles ObjectGrid suffisants pour se connecter à des ObjectGrids et en créer.

   Un exemple de fichier de règles de client objectgrid se trouve dans *rép_base_servicerest*`/security/security.ogclient.properties`. Voir également Fichier de propriétés du client.
3. Configurez la sécurité du service de données REST d'eXtreme Scale.

   Le fichier des propriétés de configuration du service de données REST eXtreme Scale doit contenir les entrées suivantes pour pouvoir être intégré à la sécurité eXtreme Scale :

   ```
   ogClientPropertyFile=nom_fichier
   ```

   ogClientPropertyFile est l'adresse du fichier de propriétés qui contient les propriétés du client ObjectGrid mentionnées au point précédent. Lorsque la sécurité est activée, le service REST utilise ce fichier pour initialiser le client eXtreme Scale afin de communiquer avec la grille.

   ```
   loginType=basic [-ou-] none
   ```

   La propriété loginType configure le service REST pour le type d'ouverture de session. Si la valeur `none` est définie, l'ID utilisateur “global” et le mot de passe définis par credentialGeneratorProps seront envoyés à la grille pour chaque transaction. Si la valeur `basic` est spécifiée, le service REST présente au client une authentification HTTP BASIC en demandant des données d'identification qu'il envoie dans chaque transaction lorsqu'il communique avec la grille.

   Pour plus d'informations sur les propriétés ogClientPropertyFile et loginType, voir Fichier de propriétés du service de données REST.

- Appliquez des règles d'accès.

  L'accès peut également être contrôlé par des règles de configuration de portée service, désignées sous le nom de règles d'accès. Il existe deux types de règles d'accès : les droits d'opérations du service qui contrôlent les opérations CRUD autorisées par le service et les droits d'accès aux entités qui contrôlent les opérations CRUD autorisées pour un type donné d'entité.

Le service de données REST d'eXtreme Scale autorise, si on le souhaite, des règles d'accès configurables à accéder de manière restreinte au service et aux entités contenues dans ce dernier. Ces règles d'accès sont spécifiées dans le fichier des propriétés des droits d'accès du service REST. Le nom de ce fichier est spécifié dans le fichier des propriétés du service de données REST par la propriété wxsRestAccessRightsFile. Pour plus d'informations sur cette propriété, voir Fichier de propriétés du service de données REST. Ce fichier est un fichier de propriétés Java classique avec des paires clé/valeur. Il existe deux types de règles d'accès : les droits d'opérations du service qui contrôlent les opérations CRUD autorisées par le service et les droits d'accès aux entités qui contrôlent les opérations CRUD autorisées pour un type donné d'entité.

1. Configurez les droits d'opérations du service.

   Les droits d'opérations du service spécifient les droits d'accès qui s'appliquent à tous les ObjectGrids exposés via le service REST ou à toutes les entités de l'ObjectGrid individuel qui est spécifié.

   Utilisez la syntaxe suivante.

   ```
   serviceOperationRights=droit_opérations_service
   serviceOperationRights.nom_grille -ou- *=droit_opérations_service
   ```

   où

   – serviceOperationRights peut être l'un des suivants : [NONE, READSINGLE, READMULTIPLE, ALLREAD, ALL]
   – serviceOperationRights.*nom_grille -ou-* * implique que le droit d'accès s'applique à tous les ObjectGrids, autrement le nom d'un ObjectGrid spécifique peut être fourni.

   Par exemple :

   ```
   serviceOperationsRights=ALL
   serviceOperationsRights.*=NONE
   serviceOperationsRights.EMPLOYEEGRID=READSINGLE
   ```

   Le premier exemple spécifie que toutes les opérations du service sont autorisées pour tous les ObjectGrids exposés par ce service REST. Le deuxième exemple est semblable au premier car il s'applique également à tous les ObjectGrids exposés par le service REST, mais il spécifie des droits d'accès NONE, ce qui signifie qu'aucune opération du service n'est autorisée sur les ObjectGrids. Le dernier exemple spécifie comment contrôler les opérations du service pour une grille spécifique ; ici seules les opérations de lecture qui donnent un seul enregistrement sont autorisées pour toutes les entités de la grille EMPLOYEEGRID.

   La valeur par défaut utilisée par le service REST est `serviceOperationsRights=ALL`, ce qui signifie que toutes les opérations sont autorisées pour tous les ObjectGrids exposés par ce service. Cela diffère de l'implémentation Microsoft pour laquelle la valeur par défaut est `NONE`, si aucune opération n'est autorisée sur le service REST.

   Important : Les droits d'opérations du service sont évalués dans l'ordre dans lequel ils sont spécifiés dans ce fichier, ce qui fait que le dernier droit à être spécifié prendra le pas sur les droits qui viennent avant lui.

2. Configurez les droits d'accès aux entités.

   Les droits d'ensembles d'entités spécifient les droits d'accès qui s'appliquent aux entités de l'ObjectGrid spécifique qui est exposé via le service REST. Ces droits permettent d'imposer un contrôle bien plus étroit et bien plus granulaire de l'accès à des entités d'un ObjectGrid individuel que ne le permettent les droits d'opérations du service.

Utilisez la syntaxe suivante.

```
entitySetRights.nom_grille.nom_entité=droit_ensemble_entités
```

où

– *droit_ensemble_entités* peut être l'un des droits suivants

Tableau 33. Droits d'accès à des entités. Valeurs prises en charge.

Droit d'accès	Description
NONE	Refuse tout droit d'accès aux données
READSINGLE	Autorise la lecture d'un seul élément de données
READMULTIPLE	Autorise la lecture d'ensembles de données
ALLREAD	Autorise toutes les opérations de lecture (élément simple ou ensembles de données)
WRITEAPPEND	Autorise la création de nouveaux éléments de données dans les ensembles de données
WRITEREPLACE	Autorise le remplacement de données
WRITEDELETE	Autorise la suppression d'éléments de données dans les ensembles de données
WRITEMERGE	Autorise la fusion de données
ALLWRITE	Autorise toutes les opérations d'écriture (création, remplacement, fusion ou suppression) de données
ALL	Autorise la création, la lecture, la modification et la suppression de données

– *nom_entité* est le nom d'un ObjectGrid spécifique au sein du service REST

– *nom_grille* est le nom d'une entité spécifique au sein de l'ObjectGrid spécifié

Remarque : Si les droits d'opérations du service et les droits d'ensembles d'entités sont spécifiés en même temps pour un ObjectGrid et ses entités, le droit appliqué sera le plus restrictif des deux, comme le montrent les exemples qui suivent. Rappelez-vous également que les droits d'ensembles d'entités sont évalués dans l'ordre où ils sont spécifiés dans le fichier. Le dernier droit à être spécifié prendra le pas sur ceux qui viennent avant lui.

Exemple 1 : Si serviceOperationsRights.NorthwindGrid=READSINGLE et entitySetRights.NorthwindGrid.Customer=ALL sont spécifiés. READSINGLE sera appliqué pour l'entité Customer.

Exemple 2 : Si serviceOperationsRights.NorthwindGrid=ALLREAD est spécifié et qu'entitySetRights.NorthwindGrid.Customer=ALLWRITE l'est aussi, seules des opérations de lecture seront autorisées pour toutes les entités de NorthwindGrid. Mais, en ce qui concerne Customer, ses droits d'ensembles d'entités empêcheront toute lecture (puisque c'est ALLWRITE qui est spécifié) et de ce fait l'entité Customer aura NONE comme droit d'accès.

- Sécurisez les transports.

  La sécurité du transport est fournie par la configuration du conteneur d'hébergement pour les connexions entre le client Web et le service REST. La sécurité du transport est fournie par la configuration du client eXtreme Scale pour les connexions entre le service Web et la grille eXtreme Scale.

1. Sécurisez la connexion entre le client et le service REST. La sécurité des transports pour cette connexion est fournie par l'environnement du conteneur hébergeant et non dans eXtreme Scale.
2. Sécurisez la connexion entre le service REST et la grille eXtreme Scale. La sécurité des transports pour cette connexion est configurée dans eXtreme Scale. Voir «Protocole TLS et couche de connexion sécurisée», à la page 515.

Intégration de la sécurité dans WebSphere Application Server

Lorsque vous déployez WebSphere eXtreme Scale dans un environnement WebSphere Application Server, vous vous pouvez simplifier le flux d'authentification et la configuration de la sécurité de la couche de transport à partir de WebSphere Application Server.

Flux d'authentification simplifié

Lorsque les clients et serveurs eXtreme Scale sont exécutés dans WebSphere Application Server et dans le même domaine de sécurité, vous pouvez utiliser l'infrastructure de sécurité de WebSphere Application Server pour propager les données d'accès pour l'authentification du client sur le serveur eXtreme Scale. Par exemple, si un servlet agit en tant que client eXtreme Scale pour se connecter à un serveur eXtreme Scale du même domaine de sécurité et si le servlet est déjà authentifié, il est possible de propager le jeton d'authentification du client (servlet) vers le serveur, puis utiliser l'infrastructure de sécurité de WebSphere Application Server pour reconvertir ce jeton en données d'accès du client.

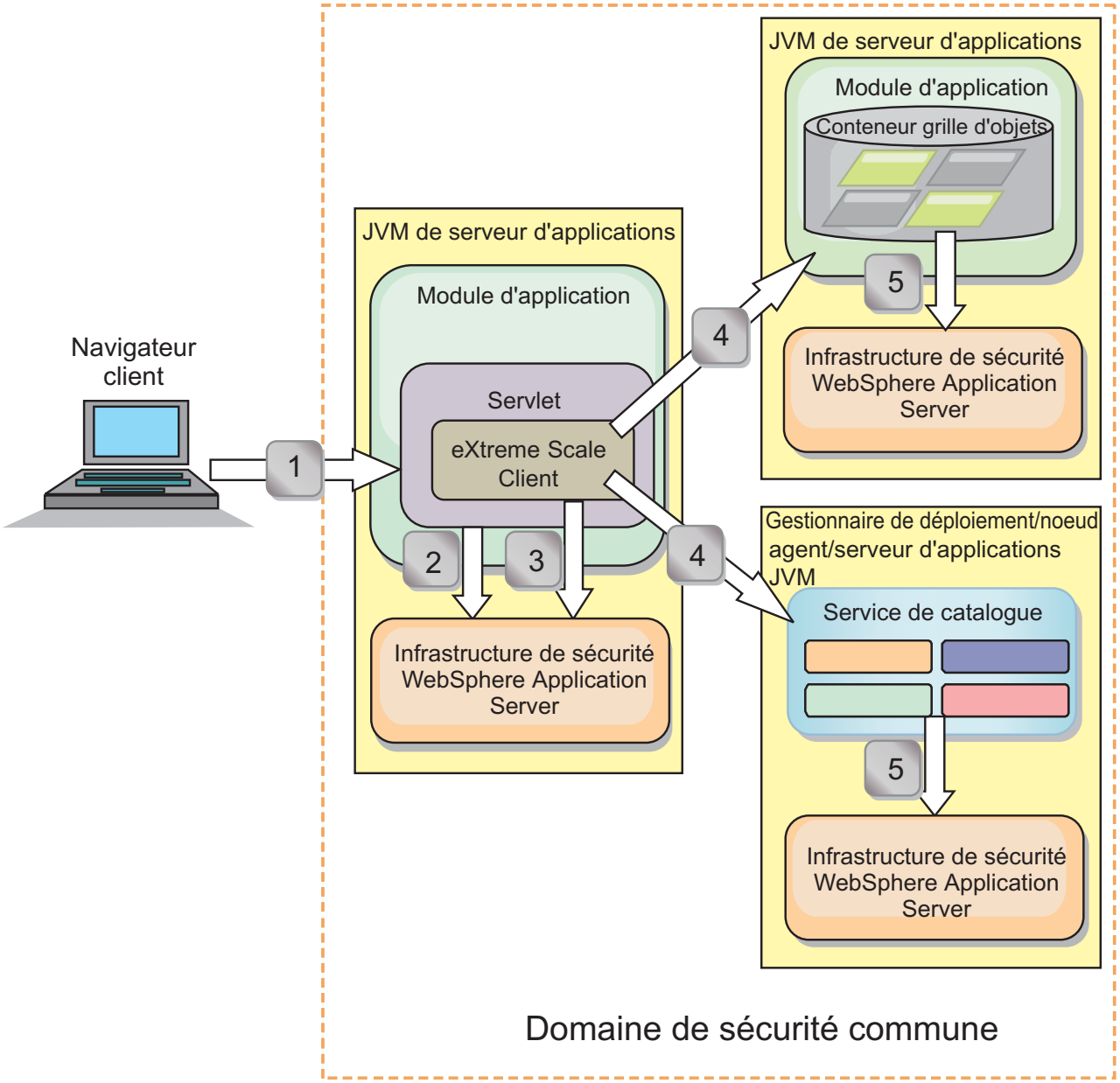


Figure 67. Flux d'authentification pour les serveurs dans le même domaine de sécurité

Dans le diagramme précédent, les serveurs d'applications se trouvent dans le même domaine de sécurité. Un serveur d'applications héberge l'application Web qui est également un client eXtreme Scale. L'autre serveur d'applications héberge le serveur de conteneur. Le gestionnaire de déploiement ou la machine virtuelle Java héberge le service de catalogue. Les flèches dans le diagramme indiquent le flux du processus d'authentification :

1. Un utilisateur d'application d'entreprise utilise un navigateur Web pour se connecter au premier serveur d'applications avec un nom d'utilisateur et un mot de passe.
2. Le premier serveur d'applications envoie le nom d'utilisateur et le mot de passe du client à l'infrastructure de sécurité WebSphere Application Server pour s'authentifier auprès du registre des utilisateurs. Par exemple, ce registre d'utilisateurs peut être un serveur LDAP. Par conséquent, les informations de sécurité sont stockées dans l'unité d'exécution du serveur d'applications.

3. Le fichier JSP (JavaServer Pages) fait office de client eXtreme Scale pour extraire les informations de sécurité à partir de l'unité d'exécution du serveur. Le fichier JSP appelle l'infrastructure de sécurité WebSphere Application Server pour obtenir les jetons de sécurité qui représentent l'utilisateur d'application d'entreprise.
4. Le client eXtreme Scale, ou un fichier JSP, envoie des jetons de sécurité avec la demande au serveur de conteneur et au service de catalogue qui est hébergé sur les autres machines virtuelles Java. Le serveur de catalogue et le serveur conteneur utilisent les jetons de sécurité WebSphere Application Server comme données d'identification du client eXtreme Scale.
5. Les serveurs de catalogue et de conteneur envoient les jetons de sécurité à l'infrastructure de sécurité WebSphere Application Server pour les convertir en informations de sécurité utilisateur. Ces informations sont représentées par un objet Subject qui contient les principaux, les données d'identification publiques et données d'identification privées. Cette conversion peut se produire, car les serveurs d'applications qui hébergent le client eXtreme Scale, le serveur de catalogue et le serveur de conteneur partagent les mêmes jetons LTPA (Lightweight Third-Party Authentication (LTPA) WebSphere Application Server.

Intégration de l'authentification

Intégration de la sécurité répartie à WebSphere Application Server :

Pour le modèle réparti, utilisez les classes suivantes :

- com.ibm.websphere.objectgrid.security.plugins.builtins. WSTokenCredentialGenerator
- com.ibm.websphere.objectgrid.security.plugins.builtins. WSTokenAuthenticator
- com.ibm.websphere.objectgrid.security.plugins.builtins. WSTokenCredential

Pour des exemples d'utilisation de ces classes, voir «Tutoriel : Intégration de la sécurité WebSphere eXtreme Scale à WebSphere Application Server», à la page 86.

Côté serveur, utilisez l'authentificateur WSTokenAuthentication pour authentifier l'objet WSTokenCredential.

Intégration de la sécurité locale à WebSphere Application Server:

Pour le modèle ObjectGrid local, utilisez les classes suivantes :

- com.ibm.websphere.objectgrid.security.plugins.builtins. WSSubjectSourceImpl
- com.ibm.websphere.objectgrid.security.plugins.builtins. WSSubjectValidationImpl

Pour plus d'informations sur ces classes, voir Programmation de la sécurité locale. Vous pouvez configurer la classe WSSubjectSourceImpl en tant que plug-in SubjectSource et la classe WSSubjectValidationImpl en tant que plug-in SubjectValidation.

Support de sécurité de la couche de transport dans WebSphere Application Server

Lorsqu'un client, un serveur de conteneur ou un serveur de catalogue eXtreme Scale s'exécute dans un processus WebSphere Application Server, la sécurité du transport eXtreme Scale est gérée par les paramètres de transport CSIV2 WebSphere Application Server. Pour le client ou le serveur eXtreme Scale, n'utilisez

pas les propriétés du client ou du serveur eXtreme Scale pour définir les paramètres SSL. Tous les paramètres SSL doivent être définis dans la configuration WebSphere Application Server.

Cependant, le serveur de catalogues est légèrement différent. Le serveur de catalogue dispose de ses propres chemins de transport propriétaires gérés par les paramètres de transport CSIV2 WebSphere Application Server. Par conséquent, il reste nécessaire de configurer les propriétés dans le fichier de propriétés du serveur pour le serveur de catalogue. Pour plus d'informations, voir «Tutoriel : Intégration de la sécurité WebSphere eXtreme Scale à WebSphere Application Server», à la page 86.

Configuration de la sécurité client dans un domaine de services de catalogue

La configuration de sécurité du client sur un domaine de services de catalogue permet de définir les propriétés de configuration de l'authentification client par défaut. Ces propriétés sont utilisées lorsque aucun fichier de propriétés du client n'est situé dans la machine virtuelle Java (JVM) qui héberge le client ou que le client ne dispose pas d'un programme pour spécifier les propriétés de sécurité. Si un fichier de propriétés client existe, les propriétés que vous spécifiez dans la console remplacent les valeurs dans le fichier. Vous pouvez remplacer ces propriétés en spécifiant un fichier `splicer.properties` avec la propriété personnalisée com.ibm.websphere.xs.sessionFilterProps ou en raccordant le fichier EAR d'application.

Avant de commencer

- Vous devez connaître l'implémentation CredentialGenerator que vous utilisez pour authentifier les clients avec la grille de données distante. Vous pouvez utiliser l'une des implémentations qui sont fournies par WebSphere eXtreme Scale : UserPasswordCredentialGenerator ou WSTokenCredentialGenerator.

  Vous pouvez également utiliser une implémentation personnalisée de l'interface CredentialGenerator. L'implémentation personnalisée doit se trouver dans le chemin d'accès aux classes du client d'exécution et le serveur. Si vous configurez un scénario de sessions HTTP avec WebSphere Application Server, vous devez placer l'implémentation dans le chemin d'accès aux classes du gestionnaire de déploiement et le chemin d'accès aux classes du serveur d'applications dans lequel le client est en cours d'exécution.
- Vous devez disposer d'un domaine de services de catalogue défini. Pour plus d'informations, voir «Création de domaines de services de catalogue dans WebSphere Application Server», à la page 258.

Pourquoi et quand exécuter cette tâche

Vous devez configurer la sécurité du client sur le domaine de services de catalogue lorsque vous avez activé l'authentification des données d'identification côté serveur, en configurant l'un des scénarios suivants :

- La stratégie de sécurité côté serveur a la propriété **`credentialAuthentication`** affectée de la valeur `Required`.
- La stratégie de sécurité côté serveur a la propriété **`credentialAuthentication`** affectée de la valeur `Supported` ET un mécanisme **`authorizationMechanism`** a été spécifié dans le fichier XML ObjectGrid.

Dans ces scénarios, des données d'identification doivent être transmises à partir du client. Ces informations sont extraites de la méthode getCredential dans une classe

qui implémente l'interface CredentialGenerator. Dans un scénario de configuration de session HTTP, l'environnement d'exécution doit connaître l'implémentation CredentialGenerator à utiliser pour générer une identification qui est envoyée à une grille de données distante. Si vous ne spécifiez pas la classe d'implémentation CredentialGenerator à utiliser, la grille de données distante refuserait les demandes du client, car le client ne peut pas être authentifié.

Procédure

Définissez les propriétés de sécurité du client. Dans la console d'administration WebSphere Application Server, cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **domaine de services de catalogue** > ***catalog_service_domain_name*** > **Propriétés de sécurité du client**. Spécifiez les propriétés de sécurité du client sur la page et enregistrez vos modifications. Voir «Propriétés de sécurité du client», à la page 273 pour la liste des propriétés que vous pouvez définir.

Résultats

Les propriétés de sécurité du client que vous avez configurées dans le domaine de services de catalogue sont utilisées comme valeurs par défaut. Les valeurs que vous indiquez remplacent les propriétés définies dans les fichiers `client.properties`.

Que faire ensuite

Configurez vos applications qu'elles utilisent WebSphere eXtreme Scale pour la gestion des sessions. Pour plus d'informations, voir «Fractionnement automatique des applications pour la gestion de session HTTP dans WebSphere Application Server», à la page 305.

Activation de la sécurité locale

WebSphere eXtreme Scale fournit plusieurs points de contact de sécurité permettant d'intégrer les mécanismes personnalisés. Dans le modèle de programmation local, la principale fonction de sécurité est l'autorisation, qui n'est associée à aucune prise en charge de l'authentification. Vous devez vous authentifier indépendamment de l'authentification WebSphere Application Server existante. Toutefois, vous pouvez utiliser les plug-ins fournis permettant d'obtenir et de valider des objets Subject.

Pourquoi et quand exécuter cette tâche

Vous pouvez activer la sécurité locale avec le fichier descripteur XML ObjectGrid ou à l'aide d'un programme.

Procédure

Activez la sécurité locale avec le fichier descripteur XML ObjectGrid.
Le fichier `secure-objectgrid-definition.xml` utilisé dans l'exemple d'application d'entreprise ObjectGridSample est présenté dans l'exemple suivant. Pour activer la sécurité, associez l'attribut securityEnabled à la valeur `true`.

```
<objectGrids>
    <objectGrid name="secureClusterObjectGrid" securityEnabled="true"
        authorizationMechanism="AUTHORIZATION_MECHANISM_JAAS">
    ...
</objectGrids>
```

Que faire ensuite

Démarrez le serveur de conteneur et le serveur de catalogue avec la sécurité activée.

Démarrage et arrêt des serveurs sécurisés

La sécurité est activée en spécifiant les configurations de sécurité lorsque vous démarrez et arrêtez des serveurs.

Démarrage des serveurs sécurisés dans un environnement autonome

Pour démarrer les serveurs autonomes sécurisés, vous envoyez les fichiers de configuration en définissant des paramètres dans la commande **startOgServer**.

Avant de commencer

Si vous utilisez un fournisseur de sécurité externe pour l'authentification ou l'autorisation du client, définissez la variable d'environnement *CLIENT_AUTH_LIB*. Ouvrez une fenêtre de ligne de commande ou de terminal et exécutez la commande correspondant à votre système d'exploitation :

- Windows `set CLIENT_AUTH_LIB=<path_to_security_JAR_or_classes>`
- UNIX `set CLIENT_AUTH_LIB=<path_to_security_JAR_or_classes> export CLIENT_AUTH_LIB`

Lorsque la commande **startOgServer** ou **stopOgServer** s'exécute, cette variable est ajoutée au chemin d'accès aux classes.

Procédure

- Démarrez les serveurs de conteneur.

  Le démarrage d'un serveur de conteneur sécurisé requiert le fichier de configuration de sécurité suivant :

  – **Fichier de propriétés du serveur :** ce fichier permet de configurer les propriétés de sécurité spécifiques au serveur. Pour plus d'informations, voir Fichier de propriétés du serveur.

  Indiquez l'emplacement de ce fichier de configuration en fournissant l'argument suivant dans le script **startOgServer** :

  -serverProps
  Spécifie l'emplacement du fichier de propriétés du serveur qui contient les propriétés de sécurité spécifiques du serveur. Le nom de fichier spécifié pour cette propriété correspond à un format de chemin de fichier classique, tel que `../security/server.properties`.

- Démarrez les serveurs de catalogue.

  Pour démarrer un service de catalogue sécurisé, vous devez disposer des fichiers de configuration suivants :

  – **Fichier descripteur XML de la sécurité :** décrit les propriétés de sécurité communes à tous les serveurs, y compris les serveurs de catalogue et de conteneur. Un exemple de propriété est la configuration de l'authentificateur qui représente le registre utilisateur et le mécanisme d'authentification.

  – **Fichiers des propriétés du serveur :** configure les propriétés de sécurité spécifiques du serveur.

Indiquez l'emplacement de ces fichiers de configuration en fournissant l'argument suivant dans le script **startOgServer** :

-clusterSecurityFile et -clusterSecurityUrl
Ces arguments indiquent l'emplacement du fichier XML du descripteur de sécurité. Utilisez le paramètre **-clusterSecurityFile** pour spécifier un fichier local ou le paramètre **-clusterSecurityUrl** pour indiquer l'adresse URL du fichier `objectGridSecurity.xml`.

-serverProps
Spécifie l'emplacement du fichier de propriétés du serveur qui contient les propriétés de sécurité spécifiques du serveur. Le nom de fichier spécifié pour cette propriété correspond à un chemin classique, tel que `c:/tmp/og/catalogserver.props`.

Démarrage des serveurs sécurisés dans WebSphere Application Server

Pour démarrer des serveurs sécurisés dans WebSphere Application Server, vous devez spécifier les fichiers de configuration de la sécurité dans les arguments génériques Java Virtual Machine (JVM).

Procédure

- Démarrez un service de catalogue sécurisé dans WebSphere Application Server.

  Un serveur de catalogue contient deux niveaux d'informations de sécurité :

  – `-Dobjectgrid.cluster.security.xml.url` : spécifie l'emplacement du fichier `objectGridSecurity.xml` qui décrit les propriétés de sécurité communes à tous les serveurs, y compris les serveurs de catalogue et les serveurs de conteneur. La configuration de l'authentificateur, qui représente le registre d'utilisateurs et le mécanisme d'authentification, est un exemple des propriétés de sécurité définies. Le nom de fichier spécifié pour cette propriété doit avoir le format URL, tel que`file:///tmp/og/objectGridSecurity.xml`.

  – `-Dobjectgrid.server.props` : spécifie le fichier de propriétés du serveur qui contient les propriétés de sécurité spécifiques du serveur. Le nom de fichier spécifié pour cette propriété correspond à un chemin classique, tel que `c:/tmp/og/catalogserver.props`.

  1. Dans la console d'administration WebSphere Application Server, cliquez sur **Administration du système**. Cliquez sur le processus sur lequel le serveur de catalogue est déployé, tel que le gestionnaire de déploiement.
  2. Cliquez sur **Java et gestion de processus** > **Définition de processus** > **Java Virtual Machine**.
  3. Entrez les propriétés dans la zone **Arguments JVM génériques**. Vous pouvez, par exemple, ajouter les valeurs suivantes :

     ```
     -Dobjectgrid.cluster.security.xml.url=file:///tmp/og/objectGridSecurity.xml
     -Dobjectgrid.server.props=/tmp/og/catalog.server.props
     ```

  4. Cliquez sur **OK** et enregistrez les modifications.

- Démarrez un serveur de catalogue sécurisé dans WebSphere Application Server.

  Un serveur de conteneur, lors de la connexion au serveur de catalogue, hérite de la configuration de sécurité qui se trouve dans le fichier `objectGridSecurity.xml`, telle que la configuration de l'authentificateur ou les paramètres de temporisation de session de connexion. Vous devez également définir les propriétés de sécurité de serveurs de conteneur spécifiques dans la propriété `-Dobjectgrid.server.props`.

Le nom de fichier spécifié pour cette propriété est au format de chemin de fichier simple, tel que `c:/tmp/og/server.props`.

Suivez la même procédure que ci-dessus pour ajouter les propriétés de sécurité aux arguments JVM génériques.

1. Ouvrez la page de la machine virtuelle Java du serveur. Dans la console d'administration WebSphere Application Server, cliquez sur **Serveurs** > **Serveurs d'applications** > *server_name* > **Java et gestion de processus** > **Définition de processus** > **Java Virtual Machine**
2. Entrez les propriétés dans la zone **Arguments JVM génériques**. Vous pouvez, par exemple, ajouter les valeurs suivantes :

   ```
   -Dobjectgrid.server.props=/opt/wxs/security/server2.props
   ```
3. Cliquez sur **OK** et enregistrez les modifications.

Arrêt des serveurs sécurisés

L'arrêt des serveurs de catalogue sécurisés ou des serveurs de conteneur requiert un fichier de configuration de sécurité.

Procédure

Arrêtez un serveur de catalogue sécurisé ou un serveur de conteneur.
L'arrêt d'un serveur sécurisé requiert le fichier de configuration de sécurité suivant :

- **Fichier de propriété client :** le fichier de propriétés client peut être utilisé pour définir les propriétés de sécurité du client. Ces propriétés de sécurité permettent à un client de se connecter à un serveur sécurisé. Pour plus d'informations, référez-vous à Fichier de propriétés du client.

Indiquez l'emplacement de ces fichiers de configuration en fournissant l'argument suivant dans le script **stopOgServer** :

-clientSecurityFile
Indique le chemin d'accès au fichier de propriétés client qui définit les propriétés de sécurité du client. Le nom de fichier que vous spécifiez pour cette propriété correspond au format de chemin de fichier classique, tel que `../security/objectGridClient.properties`.

Exemple

```
stopOgServer.bat|sh cs1 -catalogServiceEndPoints
cs1:MyServer1.company.com:6601:6602,
cs2:MyServer2.company.com:6601:6602,
cs3:MyServer3.company.com:6601:6602
-clientSecurityFile ../security/objectGridClient.properties
```

Configuration des profils de sécurité pour l'utilitaire `xscmd`

En créant un profil de sécurité, vous pouvez utiliser les paramètres de sécurité enregistrés pour utiliser l'utilitaire **xscmd** avec des environnements sécurisés.

Avant de commencer

Pour plus d'informations sur la configuration de l'utilitaire **xscmd**, voir «Administration avec l'utilitaire **xscmd**», à la page 415.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser le paramètre **-ssp** *profile_name* ou **--saveSecProfile** *profile_name* avec le reste de la commande **xscmd** pour enregistrer un profil de sécurité. Le profil peut contenir des paramètres pour les noms d'utilisateur et les mots des générateurs de données d'identification, des fichiers de clés, des fichiers de clés certifiées et des types de transport.

Le groupe de commandes **ProfileManagement** dans l'utilitaire **xscmd** contient des commandes de gestion de vos profils de sécurité.

Procédure

- Enregistrez un profil de sécurité.

  Pour enregistrer un profil de sécurité, utilisez le paramètre **-ssp** *profile_name* ou **--saveSecProfile** *profile_name* avec le reste de la commande. L'ajout de ce paramètre à la commande enregistre les paramètres suivants :

  ```
  -al,--alias <alias>
  -arc,--authRetryCount <integer>
  -ca,--credAuth <support>
  -cgc,--credGenClass <className>
  -cgp,--credGenProps <property>
  -cxpv,--contextProvider <provider>
  -ks,--keyStore <filePath>
  -ksp,--keyStorePassword <password>
  -kst,--keyStoreType <type>
  -prot,--protocol <protocol>
  -pwd,--password <password>
  -ts,--trustStore <filePath>
  -tsp,--trustStorePassword <password>
  -tst,--trustStoreType <type>
  -tt,--transportType <type>
  -user,--username <username>
  ```

  Les profils de sécurité sont enregistrés dans le répertoire *user_home*\.scmd\profiles\security\<*profile_name*>.properties.

- Utilisez un profil de sécurité enregistré.

  Pour utiliser un profil de sécurité enregistré, ajoutez le paramètre **-sp** *profile_name* ou **--securityProfile** *profile_name* à la commande que vous exécutez. Exemple de commande : `xscmd -c listHosts -cep myhost.mycompany.com -sp myprofile`

- Listez les commandes dans le groupe de commandes **ProfileManagement**.

  Exécutez la commande **xscmd -lc ProfileManagement**.

- Listez les profils de sécurité existants.

  Exécutez la commande **xscmd -c listProfiles -v**.

- Affichez les paramètres enregistrés dans un profil de sécurité.

  Exécutez la commande **xscmd -c showProfile -pn** *profile_name*.

- Supprimez un profil de sécurité existant.

  Exécutez la commande **xscmd -c RemoveProfile -pn** *profile_name*.

Chapitre 11. Résolution des incidents

Outre les journaux et de trace, les messages et les notes sur l'édition mentionnés dans la présente section, vous pouvez utiliser des outils de surveillance pour identifier et résoudre les incidents tels que l'emplacement des données dans l'environnement, la disponibilité des serveurs dans la grille de données, etc. Si vous utilisez un environnement WebSphere Application Server, vous pouvez utiliser PMI (Performance Monitoring Infrastructure). Si vous utilisez un environnement autonome, vous pouvez utiliser l'outil de surveillance d'un fournisseur, tel que CA Wily Introscope ou Hyperic HQ. Vous pouvez également utiliser et personnaliser l'utilitaire **xscmd** pour afficher des informations textuelles sur votre environnement.

Activation de la consignation

Vous pouvez utiliser des journaux pour surveiller et traiter les problèmes liés à votre environnement.

Pourquoi et quand exécuter cette tâche

Les journaux sont enregistrés dans des emplacements différents et les formats dépendant de votre configuration.

Procédure

- **Activez des journaux dans un environnement autonome.**

  Avec les serveurs de catalogue autonomes, les journaux se trouvent dans le répertoire dans lequel vous exécutez la commande **startOgServer**. Pour les serveurs de conteneur, vous pouvez utiliser l'emplacement par défaut ou définir un emplacement de journal personnalisé :

  – **Emplacement de journal par défaut :** les journaux se trouvent dans le répertoire où la commande serveur a été exécutée. Si vous démarrez les serveurs dans le répertoire *rép_base_wxs*/bin, les journaux et les fichiers de trace se trouvent dans les sous-répertoires logs/<*nom_serveur*> du répertoire bin.

  – **Emplacement de journal personnalisé :** pour définir un autre emplacement pour les journaux des serveurs de conteneur, créez un fichier de propriétés, tel que server.properties contenant :

  ```
  workingDirectory=<directory>
  traceSpec=
  systemStreamToFileEnabled=true
  ```

  La propriété **workingDirectory** est le répertoire racine des journaux et du fichier de trace facultatif. WebSphere eXtreme Scale crée un répertoire avec le nom du serveur de conteneur avec un fichier SystemOut.log, un fichier SystemErr.log et un fichier de trace. Pour utiliser un fichier de propriétés au démarrage des conteneurs, utilisez l'option **-serverProps** et spécifiez l'emplacement du fichier de propriétés du serveur.

- **Activez les journaux dans WebSphere Application Server.**

  Voir WebSphere Application Server: Activation et désactivation de la consignation pour plus d'informations.

- **Extrayez les fichiers FFDC.**

Les fichiers FFDC sont destinés au support technique d'IBM, pour le débogage. Ces fichiers peuvent être demandés par le support technique d'IBM en cas de problème. Ces fichiers se trouvent dans un répertoire libellé `ffdc` et contiennent des fichiers similaires au suivant :

```
server2_exception.log
server2_20802080_07.03.05_10.52.18_0.txt
```

Que faire ensuite

Affichage des fichiers journaux dans leur emplacement spécifié. Les messages courants à rechercher dans le fichier `SystemOut.log` sont les messages de confirmation du démarrage, comme dans l'exemple suivant :

```
CWOBJ1001I: ObjectGrid Server catalogServer01 is ready to process requests.
```

Pour plus d'informations sur un message spécifique dans les fichiers journaux, voir Messages.

Collecte de trace

Vous pouvez utiliser une trace pour surveiller et traiter les problèmes liés à votre environnement. Vous devez fournir une trace pour un serveur lorsque vous contactez le support IBM.

Pourquoi et quand exécuter cette tâche

La collecte d'une trace peut vous aider à surveiller et corriger les problèmes dans votre déploiement de WebSphere eXtreme Scale. La manière dont vous collectez la trace dépend de votre configuration. Voir «Options de trace», à la page 537 pour la liste des spécifications de trace que vous pouvez collecter.

Procédure

- **Collectez la trace dans un environnement WebSphere Application Server.**

  Si votre catalogue et les serveurs de conteneur se trouvent dans un environnement WebSphere Application Server, voir WebSphere Application Server : Utilisation avec la fonction de trace pour plus d'informations.

- **Collectez la trace avec le catalogue autonome ou la commande de démarrage de serveur.**

  Vous pouvez définir la trace sur un service de catalogue ou serveur de conteneur en utilisant les paramètres **`-traceSpec`** et **`-traceFile`** avec la commande **`startOgServer`**. Par exemple :

  ```
  startOgServer.sh catalogServer -traceSpec ObjectGridPlacement=all=enabled -traceFile  /home/user1/logs/trace.log
  ```

  Le paramètre **`-traceFile`** est facultatif. Si vous ne définissez pas un emplacement **`-traceFile`**, le fichier de trace est placé dans le même endroit que les fichiers journaux du système.Pour plus d'informations sur ces paramètres, voir la «Script **`startOgServer`**», à la page 401.

- **Collectez la trace sur le catalogue autonome ou serveur conteneur avec un fichier de propriétés.**

  Pour collecter une trace à partir d'un fichier de propriétés, créez un fichier, tel que `server.properties`, avec le contenu suivant :

  ```
  workingDirectory=<directory>
  traceSpec=<trace_specification>
  systemStreamToFileEnabled=true
  ```

La propriété **workingDirectory** est le répertoire racine des journaux et du fichier de trace facultatif. Si la valeur **workingDirectory** n'est pas définie, le répertoire de travail par défaut est l'emplacement utilisé pour démarrer les serveurs (par exemple, *rép_base_wxs*/bin). Pour utiliser un fichier de propriétés au cours du démarrage du serveur, utilisez le paramètre **-serverProps** avec la commande **startOgServer** et fournissez l'emplacement du fichier de propriétés du serveur.Pour plus d'informations sur le fichier de propriétés du serveur et l'utilisation du fichier, voir les Fichier de propriétés du serveur.

- **Collectez la trace sur un client autonome.**

  Vous pouvez démarrer la collecte de trace sur un client autonome en ajoutant des propriétés système au script de démarrage pour l'application client. Dans l'exemple suivant, les paramètres de trace sont spécifiés pour l'application com.ibm.samples.MyClientProgram :

  ```
  java -DtraceSettingsFile=MyTraceSettings.properties
  -Djava.util.logging.manager=com.ibm.ws.bootstrap.WsLogManager
  -Djava.util.logging.configureByServer=true com.ibm.samples.MyClientProgram
  ```

  Voir WebSphere Application Server : Activation de la trace sur les applications client et autonomes pour plus d'informations.

- **Collectez la trace avec l'interface ObjectGridManager.**

  Vous pouvez également définir la trace lors de la phase d'exécution sur une interface ObjectGridManager. La définition de la sur une interface ObjectGridManager permet d'extraire la trace sur un client eXtreme Scale lorsqu'il se connecte à eXtreme Scale et valide des transactions. Pour définir la trace sur une interface ObjectGridManager, fournissez une spécification de trace et un journal de trace.

  ```
  ObjectGridManager manager = ObjectGridManagerFactory.getObjectGridManager();
  ...
  manager.setTraceEnabled(true);
  manager.setTraceFileName("logs/myClient.log");
  manager.setTraceSpecification("ObjectGridReplication=all=enabled");
  ```

  Pour plus d'informations sur l'interface ObjectGridManager, voir les informations sur l'interaction avec ObjectGrid en utilisant l'interface ObjectGridManager dans *Guide de programmation*.

- **Collectez la trace sur les serveurs de conteneur avec l'utilitaire xscmd.**

  Pour collecter une trace à l'aide de l'utilitaire **xscmd**, utilisez la commande **setTraceSpec**. Utilisez l'utilitaire **xscmd** pour collecter la trace sur un environnement autonome lors de la phase d'exécution et non pas au démarrage. Vous pouvez collecter la trace sur tous les serveurs et services de catalogue ou filtrer les serveurs en fonction du nom ObjectGrid, et d'autres propriétés. Par exemple, pour collecter la trace ObjectGridReplication avec un accès au serveur de service de catalogue, exécutez :

  ```
  xscmd -c setTraceSpec "ObjectGridReplication=all=enabled"
  ```

  Vous pouvez également désactiver la trace en affectant à la spécification de trace la valeur *=all=disabled.

Résultats

Les fichiers de trace sont écrits dans l'emplacement défini.

Options de trace

Vous pouvez activer la trace pour fournir des informations sur votre environnement au service d'assistance IBM.

A propos de la trace

La trace de WebSphere eXtreme Scale est divisée en plusieurs composants. Vous pouvez définir le niveau de trace à utiliser. Les niveaux de trace courants sont les suivants : all, debug, entryExit et event.

Voici un exemple de chaîne de trace :

```
ObjectGridComponent=level=enabled
```

Vous pouvez concaténer les chaînes de trace. Utilisez le symbole `*` (astérisque) pour spécifier une valeur générique, telle que `ObjectGrid*=all=enabled`. Si vous devez fournir une trace au service d'assistance IBM, une chaîne de trace spécifique est demandée. Par exemple, en cas de problème de réplication, la trace `ObjectGridReplication=debug=enabled` peut être demandée.

Spécification de la trace

ObjectGrid
Moteur général du cache central.

ObjectGridCatalogServer
Service de catalogue général.

ObjectGridChannel
Communications statiques de la topologie de déploiement.

ObjectGridClientInfo
Informations sur le client DB2.

ObjectGridClientInfoUser
Informations sur l'utilisateur DB2.

ObjectgridCORBA
Communications dynamiques de la topologie de déploiement.

ObjectGridDataGrid
API AgentManager.

ObjectGridDynaCache
Fournisseur de cache dynamique de WebSphere eXtreme Scale.

ObjectGridEntityManager
API EntityManager. A utiliser avec l'option Projector.

ObjectGridEvictors
Expulseurs pré-intégrés d'ObjectGrid.

ObjectGridJPA
Chargeurs JPA (Java Persistence API).

ObjectGridJPACache
Plug-in de cache JPA.

ObjectGridLocking
Gestionnaire de verrouillage des entrées de cache d'ObjectGrid.

ObjectGridMBean
Beans de gestion.

ObjectGridMonitor
Infrastructure de la surveillance de l'historique.

7.1.1+ **ObjectGridNative**
Trace de code natif WebSphere eXtreme Scale, y compris le code natif eXtremeMemory.

7.1.1+ **ObjectGridOSGi**
Les composants d'intégration OSGi WebSphere eXtreme Scale.

ObjectGridPlacement
Service de positionnement des fragments de serveur de catalogues.

ObjectGridQuery
Requête ObjectGrid.

ObjectGridReplication
Service de réplication.

ObjectGridRouting
Détails du routage client/serveur.

ObjectGridSecurity
Trace de la sécurité.

7.1.1+ **ObjectGridSerializer**
Infrastructure de plug-in DataSerializer.

ObjectGridStats
Statistiques d'ObjectGrid.

ObjectGridStreamQuery
API de la requête de flux.

7.1.1+ **ObjectGridTransactionManager**
Gestionnaire de transaction WebSphere eXtreme Scale.

ObjectGridWriteBehind
Ecriture différée d'ObjectGrid.

7.1.1+ **ObjectGridXM**
Trace IBM eXtremeMemory générale.

7.1.1+ **ObjectGridXMEviction**
Trace d'expulsion eXtremeMemory.

7.1.1+ **ObjectGridXMTransport**
Trace de transport générale eXtremeMemory.

7.1.1+ **ObjectGridXMTransportInbound**
Trace de transport entrant eXtremeMemory.

7.1.1+ **ObjectGridXMTransportOutbound**
Trace de transport sortant eXtremeMemory.

Projector
Moteur dans l'API EntityManager.

QueryEngine
Moteur de requête des API Object Query et EntityManager Query.

QueryEnginePlan
Trace du plan de requête.

7.1.1+ **TCPChannel**
Canal TCP/IP IBM eXtremeIO.

7.1.1+ **XsByteBuffer**
Trace de mémoire tampon d'octets WebSphere eXtreme Scale.

Analyse des journaux et des données de trace

Vous pouvez utiliser les outils d'analyse de journal pour analyser le fonctionnement de votre environnement d'exécution et résoudre les problèmes qui s'y produisent.

Pourquoi et quand exécuter cette tâche

Vous pouvez générer des rapports à partir des fichiers journaux et de trace existants dans l'environnement. Ces rapports graphiques peuvent être utilisés pour les objectifs suivants :

- **Analyse de l'état et des performances de l'environnement d'exécution :**
  - Cohérence de l'environnement de déploiement
  - Fréquence de consignation
  - Exécution de la topologie et topologie configurée
  - Modifications non planifiées de la topologie
  - Etat de quorum
  - Etat de la réplication des partitions
  - Statistiques de mémoire, rendement, utilisation du processeur, etc.
- **Pour traiter les problèmes dans l'environnement :**
  - Vues topologiques à des points spécifiques dans le temps
  - Statistiques de mémoire, rendement, utilisation du processeur au cours des problèmes
  - Niveaux de groupe de correctifs en cours, paramètres d'optimisation
  - Etat de quorum

Présentation de l'analyse du journal

Vous pouvez utiliser l'outil **xsLogAnalyzer** pour vous aider traiter les problèmes dans l'environnement.

Tous les messages de reprise en ligne

Affiche le nombre total de messages de reprise en ligne sous la forme d'un graphique dans le temps. Affiche également une liste des messages de reprise en ligne, y compris les serveurs qui ont été affectés.

Tous les messages critiques eXtreme Scale

Affiche les ID de message et les explications associées et les actions utilisateur qui peuvent vous faire gagner du temps dans la recherche des messages.

Toutes les exceptions

Affiche les cinq premières exceptions, y compris les messages et leur nombre et les serveurs affectés par l'exception.

Résumé de la topologie

Affiche le diagramme de la configuration de la topologie en fonction des fichiers journaux. Vous pouvez utiliser ce récapitulatif pour comparer avec votre configuration réelle, en identifiant les erreurs de configuration.

Cohérence de topologie : tableau de comparaison ORB (Object Request Broker)

Affiche les paramètres ORB de l'environnement. Vous pouvez utiliser ce tableau pour déterminer si les paramètres de l'environnement sont cohérents.

Vue Tableau chronologique d'événements

Affiche un diagramme chronologique des différentes actions qui ont eu lieu sur la grille de données, y compris les événements de cycle de vie, les exceptions, les messages critiques et les événements de diagnostic de premier niveau (FFDC).

Exécution de l'analyse du journal

Vous pouvez exécuter l'outil **xsLogAnalyzer** sur un ensemble de fichiers journaux et de trace à partir de n'importe quel ordinateur.

Avant de commencer

- Activez les journaux et la trace. Pour plus d'informations, reportez-vous aux rubriques «Activation de la consignation», à la page 535 et «Collecte de trace», à la page 536.
- Collectez vos fichiers journaux. Les fichiers journaux peuvent se trouver dans des emplacements différents selon la façon dont vous les avez configurés. Si vous utilisez les paramètres de journal par défaut, vous pouvez obtenir les fichiers journaux dans les emplacements suivants :
  - Dans une installation autonome : *racine_install_wxs*/bin/logs/<*server_name*>
  - Dans une installation intégrée à WebSphere Application Server : *racine_was*/logs/<*server_name*>
- Collectez vos fichiers de trace. Ils peuvent se trouver dans des emplacements différents selon la façon dont vous les avez configurés. Si vous utilisez les paramètres de trace par défaut, vous pouvez obtenir les fichiers de trace dans les emplacements suivants :
  - Dans une installation autonome : si aucune valeur de trace n'est définie, les fichiers de trace sont écrits dans le même emplacement que les fichiers journaux système.
  - Dans une installation intégrée à WebSphere Application Server : *racine_was*/profiles/*server_name*/logs.

  Copiez les fichiers journaux et de trace vers l'ordinateur à partir duquel vous avez l'intention d'utiliser l'outil d'analyse de journal.
- Si vous voulez créez des scanners personnalisés, créez un fichier de propriétés de spécifications de scanner et un fichier de configuration avant d'exécuter l'outil. Pour plus d'informations, voir «Création de scanners personnalisés pour l'analyse de journal», à la page 542.

Procédure

1. Exécutez l'outil **xsLogAnalyzer**.

   Le script se trouve dans les emplacements suivants :
   - Dans une installation autonome : *racine_install_wxs*/ObjectGrid/bin
   - Dans une installation intégrée à WebSphere Application Server : *racine_was*/bin

   Conseil : Si les fichiers journaux sont volumineux, utilisez les paramètres **-startTime**, **-endTime**, et **-maxRecords** lorsque vous exécutez le rapport pour

limiter le nombre d'entrées de journal analysées. L'utilisation de ces paramètres lorsque vous exécutez le rapport améliore la clarté et l'exécution du rapport. Vous pouvez exécuter plusieurs rapports sur un même groupe de fichiers journaux.

```
xsLogAnalyzer.sh|bat -logsRoot c:\myxslogs -outDir c:\myxslogs\out
-startTime 11.09.27_15.10.56.089 -endTime 11.09.27_16.10.56.089 -maxRecords 100
```

-logsRoot
Spécifie le chemin absolu du répertoire des journaux à évaluer (requis).

-outDir
Spécifie un répertoire pour y placer la sortie du rapport. Si vous ne définissez pas une valeur, le rapport est écrit dans l'emplacement racine de l'outil **xsLogAnalyzer**.

-startTime
Spécifie l'heure de début de l'évaluation dans les journaux. La date a le format *year.month.day.hour.minute.second.millisecond*

-endTime
Spécifie l'heure de fin de l'évaluation dans les journaux. La date a le format *year.month.day.hour.minute.second.millisecond*

-trace Spécifie une chaîne de trace, telle que `ObjectGrid*=all=enabled`.

-maxRecords
Spécifie le nombre maximal d'enregistrements pour générer le rapport. La valeur par défaut est `100`. Si vous définissez la valeur 50, les 50 premiers enregistrements sont générés pour la période définie.

2. Ouvrez les fichiers générés. Si vous n'avez pas défini de répertoire de sortie, les rapports sont générés dans le dossier `report_date_time`. Pour ouvrir la page principale des rapports, ouvrez le fichier `index.html`.
3. Utilisez les rapports pour analyser les données des journaux. Suivez les conseils ci-dessous pour optimiser les performances du rapport affiché :
   - Pour optimiser les performances des requêtes sur les données des journaux, utilisez des informations aussi spécifiques que possibles. Par exemple, la recherche de `server` dure plus longtemps et retourne plus de résultats que `server_host_name`.
   - Certaines vues ont un nombre limité de points de données affichés simultanément. Vous pouvez ajuster le segment de temps affiché en changeant les données en cours, telles que les heures de début et de fin, dans la vue.

Que faire ensuite

Pour plus d'informations sur le traitement de l'outil **xsLogAnalyzer** et les rapports générés, voir «Traitement des problèmes d'analyse de journal», à la page 544.

Création de scanners personnalisés pour l'analyse de journal

Vous pouvez créer des scanners personnalisés pour l'analyse de journal. Après avoir configuré le scanner, les résultats sont générés dans les rapports lorsque vous exécutez l'outil **xsLogAnalyzer**. Le scanner personnalisé recherche les enregistrements d'événement dans les journaux en fonction des expressions régulières que vous avez définies.

Procédure

1. Créez un fichier de propriétés de spécification de scanner qui définit l'expression générale à exécuter pour le scanner personnalisé.
   a. Créez et enregistrez un fichier de propriétés. Le fichier doit se trouver dans le répertoire *loganalyzer_root*/config/custom. Vous pouvez attribuer le nom de choix. Le fichier est utilisé par le nouveau scanner ; il est donc utile de nommer le scanner dans le fichier des propriétés. Par exemple, *my_new_server_scanner_spec*.properties.
   b. Incluez les propriétés suivantes dans le fichier *my_new_server_scanner_spec*.properties :

      ```
      include.regular_expression = REGULAR_EXPRESSION_TO_SCAN
      ```

      La variable *REGULAR_EXPRESSION_TO_SCAN* est une expression régulière en fonction de laquelle vous filtrez les fichiers journaux.

      Exemple : pour analyser les instances des lignes qui contiennent les chaînes "xception" et "rror", quel que soit l'ordre, affectez la valeur suivante à la propriété **include.regular_expression** :

      ```
      include.regular_expression = (xception.+rror)|(rror.+xception)
      ```

      Cette expression régulière permet d'enregistrer les événements si la chaîne "rror" se trouve avant ou après la chaîne "xception".

      Exemple :Pour analyser chaque ligne des journaux pour rechercher les lignes qui contiennent la chaîne "xception" ou "rror" quel que soit l'ordre, affectez la valeur suivante à la propriété **include.regular_expression** :

      ```
      include.regular_expression = (xception)|(rror)
      ```

      Cette expression régulière permet d'enregistrer les événements si la chaîne "rror" se trouve avant ou après la chaîne "xception".
2. Créez un fichier de configuration que l'outil **xsLogAnalyer** utilise pour créer le scanner.
   a. Créez et enregistrez un fichier de configuration. Le fichier doit se trouver dans le répertoire *loganalyzer_root*/config/custom. Vous pouvez nommer le fichier *scanner_name*Scanner.config, où *scanner_name* est le nom unique du nouveau scanner. Par exemple, vous pouvez nommer le fichier serverScanner.config
   b. Incluez les propriétés suivantes dans le fichier *scanner_name*Scanner.config :

      ```
      scannerSpecificationFiles = LOCATION_OF_SCANNER_SPECIFICATION_FILE
      ```

      La variable *LOCATION_OF_SCANNER_SPECIFICATION_FILE* est le chemin et l'emplacement du fichier de spécification que vous avez créé au cours de l'étape précédente. Par exemple : *loganalyzer_root*/config/custom/my_new_scanner_spec.properties. Vous pouvez aussi définir plusieurs fichiers de spécification de scanner en utilisant une liste d'éléments séparés par un point-virgule :

```
scannerSpecificationFiles = LOCATION_OF_SCANNER_SPECIFICATION_FILE1;LOCATION_OF_SCANNER_SPECIFICATION_FILE2
```

3. Exécutez l'outil **xsLogAnalyzer**. Pour plus d'informations, voir «Exécution de l'analyse du journal», à la page 541.

Résultats

Après avoir exécuté l'outil **xsLogAnalyzer**, le rapport contient de nouveaux onglets pour les scanners personnalisés que vous avez configurés. Chaque onglet contient les vues suivantes :

Graphiques
Graphique qui illustre les événements enregistrés. Les événements sont affichés dans leurs ordre de découverte.

Tableaux
Représentation tabulaire des événements enregistrés.

Etats récapitulatifs

Traitement des problèmes d'analyse de journal

Utilisez les informations de dépannage pour identifier et éliminer les problèmes avec l'outil **xsLogAnalyzer** et ses rapports générés.

Procédure

- **Problème :** manque de mémoire lors de l'utilisation de l'outil **xsLogAnalyzer** pour générer des rapports. Exemple d'erreur possible : `java.lang.OutOfMemoryError: GC overhead limit exceeded`.

  Solution : l'outil **xsLogAnalyzer** s'exécute dans une machine JVM (Java virtual machine). Vous pouvez configurer la machine JVM pour augmenter la taille de segment avant d'exécuter l'outil **xsLogAnalyzer** en définissant certains paramètres lorsque vous exécutez l'outil. L'augmentation de la taille du segment permet de stocker plus d'enregistrements dans la mémoire JVM. Commencez avec 2 048 M en supposant que le système d'exploitation dispose d'une mémoire principale suffisante. Dans la même instance de ligne de commande dans laquelle vous voulez exécuter l'outil **xsLogAnalyzer**, définissez la taille de segment de mémoire JVM maximale :

  ```
  java -Xmx<HEAP_SIZE>m
  ```

  La valeur *HEAP_SIZE* peut être un entier et représente le nombre de mégaoctets alloués au segment de mémoire JVM. Par exemple, vous pouvez exécuter `java -Xmx2048m`. Si vous continuez de recevoir des messages indiquant un manque de mémoire ou que vous ne disposez pas des ressources pour allouer 2 048 Mo ou plus, limitez le nombre d'événements stockés dans le segment de mémoire. Vous pouvez limiter le nombre d'événements dans le segment de mémoire en envoyant le paramètre **-maxRecords** dans la commande **xsLogAnalyzer**.
- **Problème :** lorsque vous ouvrez un rapport généré depuis l'outil **xsLogAnalyzer**, le navigateur se bloque et ne charge pas la page.

  Cause : les fichiers HTML générés sont trop volumineux et le navigateur ne peut pas les charger. Ces fichiers sont volumineux, car la portée des fichiers journaux que vous analysez est trop grande.

  Solution : utilisez les paramètres **-startTime**, **-endTime**, et **-maxRecords** lorsque vous exécutez l'outil **xsLogAnalyzer** pour limiter le nombre d'entrées de journal analysées. L'utilisation de ces paramètres lorsque vous exécutez le rapport améliore la clarté et l'exécution du rapport. Vous pouvez exécuter plusieurs rapports sur un même groupe de fichiers journaux.

Traitement des problèmes d'installation

Utilisez ces informations pour traiter les problèmes d'installation.

Procédure

- **Problème :** lorsque vous exécutez la commande d'installation à partir d'un ordinateur distant, tel que `\\mymachine\downloads\`, le message suivant s'affiche : `CMD.EXE was started with the above path as the current directory. UNC paths are not supported. Defaulting to Windows directory.` En conséquence, l'installation ne peut pas se terminer correctement.

  Solution : mappez l'ordinateur distant à une unité réseau. Par exemple, dans Windows, vous pouvez cliquer avec le bouton droit de la souris sur **Ordinateur**, choisir **Connecter un lecteur réseau** et inclure le chemin UNC (uniform naming conventions) vers l'ordinateur distant. Ensuite, vous pouvez exécuter le script d'installation depuis le lecteur réseau avec succès. Par exemple, `y:\mymachine\downloads\WXS\install.bat`.

- **Problème :** l'installation n'aboutit pas.

  Solution : vérifiez les fichiers journaux pour savoir où l'installation a échoué. Lorsque l'installation n'aboutit pas, les fichiers journaux se trouvent dans le répertoire *racine_install_wxs*`/logs/wxs`.

- **Problème :** échec catastrophique lors de l'installation.

  Solution : vérifiez les fichiers journaux pour savoir où l'installation a échoué. Lorsque l'installation a été partiellement exécutée, le journaux se trouvent généralement dans le répertoire *user_root*`/wxs_install_logs/`.

- Windows **Problème :** si vous installez WebSphere eXtreme Scale Client sur Windows, le texte suivant peut s'afficher dans les résultats de l'installation :

  ```
  Success: The installation of the following product was successful:
  WebSphere eXtreme Scale Client.  Some configuration steps have errors.
  For more information, refer to the following log file:
  <WebSphere Application Server install root>\logs\wxs_client\install\log.txt"
  Review the installation log (log.txt) and review the deployment manager
  augmentation log.
  ```

  Solution : si vous identifiez une erreur avec le fichier `iscdeploy.sh`, vous pouvez l'ignorer. Cette erreur ne pose pas de problème.

Traitement des problèmes d'intégration du cache

Utilisez ces informations pour traiter les problèmes de la configuration de l'intégration du cache, y compris ceux associés aux configurations de session HTTP et de cache dynamique.

Procédure

- 7.1.1+ **Problème :** les ID de session HTTP ne sont pas réutilisés.

  Cause : vous pouvez réutiliser les ID de session. Si vous créez une grille de données pour la persistance des sessions dans la version 7.1.1 ou une version ultérieure, la réutilisation des ID de session est automatiquement activée. Toutefois, si vous avez créé des configurations dans des versions antérieures, ce paramètre est peut être déjà défini avec une valeur incorrecte.

  Solution : vérifiez les paramètres suivants pour déterminer si vous avez activé la réutilisation des ID de session HTTP :

  – La propriété `reuseSessionId` dans le fichier `splicer.properties` doit avoir la valeur `true`.

  – La propriété personnalisée `HttpSessionIdReuse` doit avoir la valeur `true`. Cette propriété personnalisée peut être définie dans l'un des chemins suivants dans la console d'administration WebSphere Application Server :

    - **Serveurs** > *server_name* > **Getion de session** > **Propriétés personnalisées**

- **Clusters dynamiques** > *dynamic_cluster_name* > **Modèle de serveur** > **Gestin de session** > **Propriétés personnalisés**
- **Serveurs** > **Types de serveur** > **Serveurs d'applications WebSphere** > *server_name*, puis sous Infrastructure du serveur, cliquez sur **Java et gestion des processus** > **Définition de processus** > **Java virtual machine** > **Propriétés personnalisées**
- **Serveurs** > **Types de serveur** > **Serveurs d'applications WebSphere** > *server_name* > **Paramètres de conteneur Web** > **Conteneur Web**

Si vous mettez à jour les valeurs des propriétés personnalisées, reconfigurez la gestion des sessions eXtreme Scale afin que le fichier `splicer.properties` détecte la modification

- **Problème :** lorsque vous utilisez un grille de données pour stocker les sessions HTTP et que la charge des transactions est élevée, le message CWOBJ0006W figure dans le fichier `SystemOut.log`.

  ```
  CWOBJ0006W: An exception occurred:
  com.ibm.websphere.objectgrid.ObjectGridRuntimeException:
  java.util.ConcurrentModificationException
  ```

  Ce message apparaît lorsque le paramètre **`replicationInterval`** dans le fichier `splicer.properties` a une valeur supérieure à zéro et que l'application Web modifie l'objet List défini comme attribut dans la session HTTPSession.

  Solution : clonez l'attribut qui contient l'objet List modifié et placez l'attribut cloné dans l'objet session.

Traitement des problèmes du plug-in de mémoire cache JPA

Utilisez ces informations pour traiter les problèmes de configuration du plug-in de mémoire cache JPA. Ces problèmes peuvent se produire dans les deux configurations Hibernate et OpenJPA.

Procédure

- **Problème :** l'exception suivante s'affiche : `CacheException: Failed to get ObjectGrid server`.

  Avec la valeur d'attribut `EMBEDDED` ou `EMBEDDED_PARTITION` **`ObjectGridType`**, la mémoire cache eXtreme Scale tente d'obtenir une instance de serveur de l'environnement d'exécution. Dans un environnement Java Platform, Standard Edition, un serveur eXtreme Scale avec un service de catalogue intégré est démarré. Le service de catalogue intégré essaie d'écouter sur le port 2809. Si ce port est utilisé par un autre processus, l'erreur se produit.

  Solution : si des noeuds finaux de service de catalogue externes sont spécifiés, par exemple, avec le fichier `objectGridServer.properties`, cette erreur se produit si le nom d'hôte ou le port ne sont spécifiés correctement. Corrigez le conflit de port.

- **Problème :** l'exception suivante s'affiche : `CacheException: Failed to get REMOTE ObjectGrid for configured REMOTE ObjectGrid. objectGridName = [ObjectGridName], PU name = [persistenceUnitName]`

  Cette erreur se produit, car le cache ne peut pas obtenir l'instance ObjectGrid à partir des noeuds finaux de service de catalogue fournis.

  Solution : ce problème se produit généralement lorsque le nom ou le port hôte est incorrect.

- **Problème :** l'exception suivante s'affiche : `CacheException: Cannot have two PUs [persistenceUnitName_1, persistenceUnitName_2] configured with same ObjectGridName [ObjectGridName] of EMBEDDED ObjectGridType`

Cette exception se produit si un grand nombre d'unités de persistance sont configurées et que les mémoires caches eXtreme Scale de ces unités sont configurées avec le même nom ObjectGrid et la même valeur d'attribut `EMBEDDED` **`ObjectGridType`**. Ces configurations d'unités de persistance peuvent être dans les mêmes fichiers `persistence.xml` ou dans des fichiers différents.

Solution : vous devez vérifier que le nom d'ObjectGrid est unique pour chaque unité de persistance lorsque la valeur de l'attribut **`ObjectGridType`** est `EMBEDDED`.

- **Problème :** l'exception suivante s'affiche : `CacheException: REMOTE ObjectGrid [ObjectGridName] does not include required BackingMaps [mapName_1, mapName_2,...]`

  Avec un type ObjectGrid REMOTE, si l'ObjectGrid obtenu côté client ne dispose pas des mappes de sauvegarde complètes d'entités pour prendre en charge le cache de l'unité de persistance, cette exception se produit. Supposons par exemple que cinq classes d'entités sont répertoriées dans la configuration des unités de persistance mais que l'ObjectGrid obtenu ne dispose que de deux mappes de sauvegarde. Même si l'ObjectGrid obtenu peut avoir 10 BackingMaps, si l'une des cinq BackingMaps d'entité requises est introuvable dans le 10 mappes de sauvegarde, cette exception se produit toujours.

  Solution : vérifiez que votre configuration de mappes de sauvegarde prend en charge la mémoire cache de l'unité de persistance.

Traitement des problèmes d'administration

Utilisez les informations suivantes pour traiter les problèmes d'administration, notamment le démarrage et l'arrêt des serveurs, en utilisant l'utilitaire **`xscmd`**, etc.

Procédure

- **Problème :** les scripts d'administration manquent dans le répertoire *`profile_root`*`/bin` d'une installation WebSphere Application Server.

  Cause : lorsque vous mettez à jour l'installation, les nouveaux fichiers scripts ne sont pas installés automatiquement dans les profils.

  Solution : si vous voulez exécuter un script depuis le répertoire *`profile_root`*`/bin`, annulez l'extension et étendez de nouveau le profils avec la dernière version. Pour plus d'informations, voir Annulation de l'extension d'un profil en utilisant l'invite de commande et «Création et augmentation de profils pour WebSphere eXtreme Scale», à la page 184.

- **Problème :** lorsque vous exécutez une commande **`xscmd`**, le message suivant s'affiche :

```
java.lang.IllegalStateException: Placement service MBean not available.
[]
        at
com.ibm.websphere.samples.objectgrid.admin.OGAdmin.main(OGAdmin.java:1449)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:60)
        at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:37)
        at java.lang.reflect.Method.invoke(Method.java:611)
        at com.ibm.ws.bootstrap.WSLauncher.main(WSLauncher.java:267)
Ending at: 2011-11-10 18:13:00.000000484
```

  Cause : problème de connexion avec le serveur de catalogue.

  Solution : vérifiez que les serveurs de catalogue sont actifs et disponibles via le réseau. Ce message peut aussi être généré lorsque vous disposez d'un domaine de services de catalogue défini et que moins de deux serveurs de catalogue sont actifs. L'environnement n'est pas disponible tant que deux serveurs de catalogue ne sont pas démarrés.

Traitement des problèmes de plusieurs configurations de centre de données

Utilisez ces informations pour traiter les problèmes de plusieurs configurations, y compris de la liaison entre les domaine de services de catalogue.

Procédure

Problème : données manquantes dans un ou plusieurs domaine de services de catalogue. Par exemple, vous pouvez exécuter la commande `xscmd -c establishLink`. Lorsque vous examinez les données de chaque domaine de services de catalogue lié, les données sont différentes par rapport à la commande `xscmd -c showMapSizes`.
Solution : vous pouvez traiter ce problème avec la commande `xscmd -c showLinkedPrimaries`. Cette commande consigne chaque fragment primaire, y compris les fragments primaires externes qui sont liés.
Dans le scénario décrit, vous pouvez déterminer à partir de la commande `xscmd -c showLinkedPrimaries` que les fragments primaires du premier domaine de services de catalogue sont liés aux fragments primaires du second domaine de services de catalogue et que ce dernier n'a pas de liaison au premier domaine de services de catalogue. Vous pouvez réexécuter la commande `xscmd -c establishLink` depuis le second domaine de services de catalogue vers le premier domaine de services de catalogue.

Traitement des problèmes des chargeurs

Utilisez ces informations pour traiter les problèmes liés aux chargeurs de base de données.

Procédure

- **Problème :** Lorsque vous utilisez un chargeur OpenJPA avec DB2 dans WebSphere Application Server, une exception de curseur fermé se produit.

  L'exception suivante provient de DB2 et figure dans le fichier journal org.apache.openjpa.persistence.PersistenceException :

  ```
  [jcc][t4][10120][10898][3.57.82] Invalid operation: result set is closed.
  ```

  Solution : par défaut, le serveur d'applications attribue à la propriété personnalisée resultSetHoldability la valeur 2 (CLOSE_CURSORS_AT_COMMIT). Cette propriété amène DB2 à fermer son resultSet/curseur au niveau des limites de la transaction. Pour supprimer l'exception, affectez à la propriété personnalisée, la valeur 1 (HOLD_CURSORS_OVER_COMMIT). Définissez la propriété personnalisée resultSetHoldability dans le chemin suivant dans la cellule WebSphere Application Server : **Ressources** > **Fournisseur JDBC** > **DB2 Universal JDBC Driver Provider** > **DataSources** > *data_source_name* > **Propriétés personnalisées** > **Nouveau**.
- **Problème** DB2 affiche une exception : `The current transaction has been rolled back because of a deadlock or timeout. Reason code "2".. SQLCODE=-911, SQLSTATE=40001, DRIVER=3.50.152`

  Cette exception se produit en raison d'un problème de conflit de verrouillage lorsque vous exécutez OpenJPA avec DB2 dans WebSphere Application Server. Le niveau d'isolement par défaut pour WebSphere Application Server est Lecture reproductible (RR), qui obtient des verrous de longue durée avec DB2.**Solution :**

Définissez le niveau d'isolement Read Committed pour réduire les conflits de verrouillage. Définissez la propriété personnalisée de source de données webSphereDefaultIsolationLevel pour spécifier le niveau d'isolement 2(TRANSACTION_READ_COMMITTED) dans le chemin suivant dans la cellule WebSphere Application Server : **Ressources** > **Fournisseur JDBC** > ***JDBC_provider*** > **Sources de données** > ***data_source_name*** > **Propriétés personnalisées** > **Nouveau**. Pour plus d'informations sur la propriété personnalisée webSphereDefaultIsolationLevel et les niveaux d'isolement de transaction, voir Conditions de définition des niveaux d'isolement de l'accès aux données.

- **Problème :** lorsque vous utilisez la fonction de préchargement de JPALoader ou JPAEntityLoader, le message CWOBJ1511 suivant ne s'affiche pas pour la partition dans un serveur de conteneur : `CWOBJ1511I: GRID_NAME:MAPSET_NAME:PARTITION_ID (primary) is open for business.`

  A la place, une exception TargetNotAvailableException est généré dans le serveur de conteneur qui active la partition définie par la propriété preloadPartition.

  Solution : affectez à l'attribut preloadMode la valeur `true` si vous utilisez un chargeur JPALoader ou JPAEntityLoader pour précharger les données dans la mappe. Si la propriété preloadPartition du chargeur JPALoader et JPAEntityLoader a une valeur comprise entre `0` et `total_number_of_partitions - 1`, il tente de précharger les données à partir de la base de données dorsale dans la mappe. Le fragment de code ci-dessous illustre comment l'attribut preloadMode est défini pour activer le préchargement asynchrone :

  ```
  BackingMap bm = og.defineMap( "map1" );
  bm.setPreloadMode( true );
  ```

  Vous pouvez également définir l'attribut preloadMode à l'aide d'un fichier XML, comme le montre l'exemple suivant :

  ```
  <backingMap name="map1" preloadMode="true" pluginCollectionRef="map1"
   lockStrategy="OPTIMISTIC" />
  ```

Traitement des problèmes de configuration XML

Lorsque vous configurez eXtreme Scale, vous pouvez rencontrer un comportement inattendu avec vos fichiers XML. Les sections ci-après décrivent les problèmes qui peuvent se produire et les solutions.

Procédure

- **Problème :** votre stratégie de déploiement et les fichiers XML ObjectGrid doivent concorder.

  La règle de déploiement et les fichiers XML ObjectGrid doivent concorder. Si les noms ObjectGrid et les noms de la mappe ne concordent pas, des erreurs se produisent.

  Si la liste de la backingMap dans un fichier XML ObjectGrid ne correspond pas à la liste des références de la mappe d'un fichier XML de règle de déploiement, une erreur se produit sur le serveur de catalogue.

  Par exemple, le fichier XML ObjectGrid et le fichier XML de la règle de déploiement ci-dessous permettent de démarrer un processus de conteneur. Le fichier de la règle de déploiement contient davantage de références de mappe que celles listées dans le fichier XML ObjectGrid.

  ObjectGrid.xml - exemple incorrect

  ```
  <?xml version="1.0" encoding="UTF-8"?>
  <objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  ```

```
 xmlns="http://ibm.com/ws/objectgrid/config">
    <objectGrids>
        <objectGrid name="accounting">
            <backingMap name="payroll" readOnly="false" />
        </objectGrid>
    </objectGrids>
</objectGridConfig>
```

deploymentPolicy.xml - exemple incorrect

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
    <objectgridDeployment objectgridName="accounting">
        <mapSet name="mapSet1" numberOfPartitions="4" minSyncReplicas="1"
     maxSyncReplicas="2" maxAsyncReplicas="1">
            <map ref="payroll"/>
            <map ref="ledger"/>
        </mapSet>
    </objectgridDeployment>
</deploymentPolicy>
```

Messages : un message d'erreur apparaît dans le fichier `SystemOut.log` lorsque la stratégie de déploiement est incompatible avec le fichier XML ObjectGrid. Pour l'exemple précédent, ce message est le suivant :

```
CWOBJ3179E: La référence à la mappe Ledger dans le groupe de mappes mapSet1 du fichier de descripteur
de déploiement Accouting ne fait pas référence à une mappe de
sauvegarde valide dans le XML de l'ObjectGrid.
```

S'il manque dans la règle de déploiement des références aux backingMaps répertoriées dans le fichier XML ObjectGrid, un message d'erreur est écrit dans le fichier `SystemOut.log`. Par exemple :

```
CWOBJ3178E: La mappe Ledger dans Accounting de l'ObjectGrid référencée dans le XML de l'ObjectGrid
est introuvable dans le fichier de descripteur du déploiement.
```

Solution : identifiez le fichier contient la liste correcte et modifiez le code approprié en conséquence.

- **Problème :** des noms ObjectGrid incorrects entre les fichiers XML génèrent également une erreur.

  Le nom de l'ObjectGrid est référencé à la fois dans le fichier XML ObjectGrid et le fichier XML de la règle de déploiement.

  Message : une exception ObjectGridException se produit causée par l'exception IncompatibleDeploymentPolicyException. Exemple :

  Causée par :
  com.ibm.websphere.objectgrid.IncompatibleDeploymentPolicyException :
  L'objectgridDeployment avec objectGridName "accountin" n'a pas d'objectGrid correspondant dans le XML ObjectGrid.

  Le fichier XML ObjectGrid est la liste principale des noms ObjectGrid. Si une règle de déploiement a un nom ObjectGrid qui n'est pas contenu dans le fichier XML ObjectGrid, une erreur se produit.

  Solution : vérifiez les détails tels que l'orthographe du nom ObjectGrid. Supprimez les noms redondants ou ajoutez les noms ObjectGrid manquants dans le fichier XML ObjectGrid ou le fichier XML de la règle de déploiement. Dans l'exemple de message, la valeur de l'objectGridName est mal orthographiée ("accountin" au lieu de "accounting").

- **Problème :** des attributs dans le fichier XML ne peuvent pas être affectés de certaines valeurs. Les valeurs acceptées par ces attributs sont énumérées par le schéma. La liste suivante indique certains de ces attributs :
  - Attribut authorizationMechanism sur l'élément objectGrid
  - Attribut copyMode sur l'élément backingMap
  - Attribut lockStrategy sur l'élément backingMap
  - Attribut ttlEvictorType sur l'élément backingMap
  - Attribut type sur l'élément property
  - initialState sur l'élément objectGrid

– evictionTriggers sur l'élément backingMap

Si une valeur non valide est attribuée à l'un de ces attributs, la validation XML échoue. Dans l'exemple suivant de fichier XML, une valeur incorrecte INVALID_COPY_MODE est utilisée :

Exemple INVALID_COPY_MODE

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
    <objectGrids>
        <objectGrid name="accounting">
            <backingMap name="payroll" copyMode="INVALID_COPY_MODE"/>
        <objectGrid/>
   </objectGrids>
</objectGridConfig>
```

Le message suivant s'affiche dans le fichier journal.

```
CWOBJ2403E : le fichier XML n'est pas valide. Un problème a été détecté
avec
< null > à la ligne 5. Le message d'erreur est cvc-enumeration-valid :
Value 'INVALID_COPY_MODE' is not facet-valid with respect to enumeration
'[COPY_ON_READ_AND_COMMIT, COPY_ON_READ, COPY_ON_WRITE, NO_COPY, COPY_TO_BYTES]'.
La valeur doit être l'une des valeurs énumérées.
```

- **Problème :** des attributs ou des balises manquants ou incorrects dans le fichier XML provoquent des erreurs, telles que dans l'exemple suivant dans lequel le fichier XML ObjectGrid ne contient pas la balise de fin < /objectGrid > :

  attributs manquants- exemple XML

  ```
  <?xml version="1.0" encoding="UTF-8"?>
  <objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
   xmlns="http://ibm.com/ws/objectgrid/config">
      <objectGrids>
          <objectGrid name="accounting">
              <backingMap name="payroll" />

     </objectGrids>
  </objectGridConfig>
  ```

  Message :

  ```
  CWOBJ2403E : le fichier XML n'est pas valide. Un problème a été détecté avec
  < null > à la ligne 7. Le message d'erreur est le suivant :
  la balise de fin du type d'élément "objectGrid" doit se terminer
  avec le délimiteur '>'.
  ```

  Une exception ObjectGridException concernant le fichier XML non valide se produit avec le nom du fichier XML.

  Solution : vérifiez que les balises et attributs nécessaires figurent dans les fichiers XML avec le format correct.

- **Problème :** si un fichier XML est formaté avec une syntaxe incorrecte ou manquante, le message CWOBJ2403E apparaît dans le journal. Par exemple, le message suivant s'affiche lorsqu'un guillemet manque pour l'un des attributs XML.

  ```
  CWOBJ2403E : le fichier XML n'est pas valide. Un problème a été détecté avec
  < null > à la ligne 7. Le message d'erreur est le suivant :
  un guillemet ouvrant est attendu pour l'attribut "maxSyncReplicas" associé
  à un type d'élément "mapSet".
  ```

  Une exception ObjectGridException concernant le fichier XML non valide se produit également.

  Solution : il existe différentes solutions pour une erreur de syntaxe XML. Consultez la documentation appropriée sur l'écriture de script XML.

- **Problème :** le référencement d'une collection de plug-ins inexistante rend un fichier XML non valide. Par exemple, lorsque vous utilisez XML pour définir des plug-ins BackingMap, l'attribut pluginCollectionRef de l'élément backingMap doit faire référence à une collection backingMapPluginCollection. L'attribut pluginCollectionRef doit correspondre aux éléments backingMapPluginCollection.

  Message :

Si l'attribut pluginCollectionRef ne correspond à aucun attribut d'ID de l'un des éléments backingMapPluginConfiguration, le message suivant ou un message similaire s'affiche dans le fichier journal.

```
[7/14/05 14:02:01:971 CDT] 686c060e XmlErrorHandl E CWOBJ9002E :
Message informatif en anglais uniquement : fichier XML non valide.
Ligne : 14 ; URI :
null ; Message : la clé 'pluginCollectionRef' avec
la valeur 'bookPlugins' est introuvable pour la contrainte d'identité de
l'élément 'objectGridConfig'.
```

Le fichier XML suivant est utilisé pour produire l'erreur. Notez que l'attribut pluginCollectionRef de la BackingMap book est défini sur bookPlugins et que l'ID de la backingMapPluginCollection est collection1.

Référencement d'un attribut XML inexistant - Exemple

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
   <objectGrids>
      <objectGrid name="bookstore">
         <backingMap name="book" pluginCollectionRef="bookPlugin" />
      </objectGrid>
   </objectGrids>
   <backingMapPluginCollections>
    <backingMapPluginCollection id="collection1">
     <bean id="Evictor"
       className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" />
    </backingMapPluginCollection>
   </backingMapPluginCollections>
</objectGridConfig>
```

Solution :

Pour résoudre le problème, assurez-vous que la valeur de chaque pluginCollectionRef correspond à l'ID de l'un des éléments de la backingMapPluginCollection. Modifiez simplement le nom de pluginCollectionRef en collection1 pour ne pas recevoir cette erreur. Une autre solution consiste à modifier l'ID de la backingMapPluginCollection existante pour le faire correspondre à pluginCollectionRef, ou d'ajouter une backingMapPluginCollection ayant un ID correspond à pluginCollectionRef.

- **Problème :** IBM Software Development Kit (SDK) Version 5 contient une implémentation de la fonction Java API for XML Processing (JAXP) à utiliser pour la validation XML par rapport à un schéma. Lorsque vous utilisez un kit de développement de logiciels ne contenant pas cette implémentation, les tentatives de validation risquent d'échouer.

  Lorsque vous tentez de valider XML avec un kit de développement de logiciels n'ayant pas l'implémentation nécessaire, le fichier journal contient l'erreur suivante :

  ```
  La validation XML XmlConfigBuild est activée
  SystemErr R com.ibm.websphere.objectgrid
  SystemErr R at com.ibm.ws.objectgrid.ObjectGridManagerImpl.getObjectGridConfigurations
   (ObjectGridManagerImpl.java:182)
  SystemErr R at com.ibm.ws.objectgrid.ObjectGridManagerImpl.createObjectGrid(ObjectGridManagerImpl.java:309)
  SystemErr R at com.ibm.ws.objectgrid.test.config.DocTest.main(DocTest.java:128)
  SystemErr R Caused by: java.lang.IllegalArgumentException : aucun attribut implémenté
  SystemErr R at org.apache.crimson.jaxp.DocumentBuilderFactoryImpl.setAttribute(DocumentBuilderFactoryImpl.java:93)
  SystemErr R at com.ibm.ws.objectgrid.config.XmlConfigBuilder.<init>XmlConfigBuilder.java:133)
  SystemErr R at com.ibm.websphere.objectgrid.ProcessConfigXML$2.runProcessConfigXML.java:99)...
  ```

  Le kit de développement de logiciels utilisé ne contient pas l'implémentation de la fonction JAXP nécessaire pour valider les fichiers XML en fonction d'un schéma.

  Solution : si vous souhaitez valider XML en utilisant un kit SDK qui ne contient pas l'implémentation JAXP, téléchargez Apache Xerces, et incluez ses fichiers JAR (Java archive) dans le chemin d'accès aux classes. Pour éviter ce problème, après avoir téléchargé Xerces et inclus les fichiers JAR dans le chemin d'accès aux classes, vous pouvez valider le fichier XML.

Traitement des problèmes de sécurité

Utilisez ces informations pour traiter les problèmes de configuration de sécurité.

Procédure

- **Problème :** l'extrémité client de la connexion nécessite SSL (Secure Sockets Layer), avec le paramètre transportType affecté de la valeur `SSL-Required`. Toutefois, l'extrémité serveur de la connexion ne prend pas en charge SSL et le paramètre transportType a la valeur `TCP/IP`. En conséquence, l'exception suivante est chaînée à une autre exception dans les fichiers journaux :

  ```
  java.net.ConnectException: connect: Address is invalid on local machine, or
  port is not valid on remote machine
      at java.net.PlainSocketImpl.doConnect(PlainSocketImpl.java:389)
      at java.net.PlainSocketImpl.connectToAddress(PlainSocketImpl.java:250)
      at java.net.PlainSocketImpl.connect(PlainSocketImpl.java:237)
      at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:385)
      at java.net.Socket.connect(Socket.java:540)
      at
  com.ibm.rmi.transport.TCPTransportConnection.createSocket(TCPTransportConnection.java:155)
      at
  com.ibm.rmi.transport.TCPTransportConnection.createSocket(TCPTransportConnection.java:167)
  ```

  L'adresse dans cette exception peut être un serveur de catalogue, un serveur de conteneur ou un client.

  Solution: voir «Configuration des types de transports sécurisés», à la page 516 pour le tableau des configurations de sécurité valides entre les clients et les serveurs.
- Lorsqu'un agent est utilisé, le client envoie l'appel de l'agent au serveur et le serveur envoie la réponse au client pour accuser réception de l'appel de l'agent. Lorsque l'agent a terminé le traitement, le serveur établit une connexion pour envoyer les résultats de l'agent. Ainsi, le serveur de conteneur devient un client du point de vue de la connexion. Par conséquent, si TLS/SSL est configuré, veillez à importer le certificat public du client dans le fichier de clés certifiées du serveur.

IBM Support Assistant for WebSphere eXtreme Scale

IBM Support Assistant permet de collecter des données, d'analyser des symptômes et d'accéder à des informations sur les produits.

IBM Support Assistant Lite

IBM Support Assistant Lite for WebSphere eXtreme Scale assure une collecte automatique des données et l'analyse des symptômes pour l'identification des problèmes et de leurs causes.

IBM Support Assistant Lite réduit le temps consacré à la reproduction des problèmes en adaptant son ensemble de niveaux de traçabilité (niveaux de fiabilité, de disponibilité et de facilité de maintenance, qui sont définis automatiquement par l'outil) afin de simplifier l'identification des problèmes. Mais si cette assistance ne suffit pas et que vous ayez besoin de l'aide d'un technicien, IBM Support Assistant Lite réduit également le temps consacré à l'envoi des informations appropriées au support technique d'IBM.

IBM Support Assistant Lite est inclus dans chaque installation de WebSphere eXtreme Scale version 7.1.0

IBM Support Assistant

IBM Support Assistant (ISA) permet d'accéder rapidement à des ressources de produits, de formation et de support qui pourront vous aider à répondre de vous-mêmes à vos questions et à résoudre les problèmes rencontrés avec des logiciels IBM sans avoir besoin de contacter le support IBM. Différents plug-in spécifiques à différents produits vous permettent de personnaliser IBM Support Assistant en fonction des produits particuliers que vous avez installés. IBM Support Assistant peut également collecter des données système, des fichiers journaux et d'autres informations qui aideront le support technique d'IBM à déterminer la cause des problèmes.

IBM Support Assistant est un utilitaire qui s'installe sur le poste de travail et non sur le serveur WebSphere eXtreme Scale lui-même. En effet, sa mémoire et ses besoins en ressources risqueraient d'affecter de manière négative les performances du serveur WebSphere eXtreme Scale. Les composants portables de diagnostics qui sont inclus dans l'Assistant sont conçus pour avoir un impact minimal sur le fonctionnement normal d'un serveur.

IBM Support Assistant est utilisable des manières suivantes :

- pour effectuer des recherches dans des sources IBM et non IBM de connaissances et d'information sur plusieurs produits IBM afin de répondre à une question ou de résoudre un problème
- pour trouver des informations complémentaires dans des ressources Web dédiées à un produit donné (pages d'accueil du produit et de son support, groupes de discussions et forums d'utilisateurs, ressources d'acquisition de compétences et de formations, informations de résolution des problèmes et FAQ)
- pour renforcer vos capacités à diagnostiquer les problèmes d'un produit donné grâce aux outils de diagnostics ciblés proposés par l'Assistant
- pour simplifier la collecte des données de diagnostic afin de vous aider, IBM et vous, à résoudre vos problèmes (collecte de données générales ou liées à un symptôme particulier)
- pour vous aider à signaler des problèmes au support IBM via une interface personnalisée en ligne avec possibilité d'attacher aux incidents signalés les données de diagnostic mentionnées plus haut ou toute autre information

Enfin, la fonctionnalité Updater intégrée permet de mettre à jour le support pour d'autres produits logiciels et d'autres fonctionnalités au fur et à mesure de leur disponibilité. Pour configurer IBM Support Assistant afin de l'utiliser avec WebSphere eXtreme Scale, commencez par l'installer à l'aide des fichiers fournis dans l'image téléchargée à partir de la page Web IBM Support Overview (http://www-947.ibm.com/support/entry/portal/Overview/Software/Other_Software/IBM_Support_Assistant. Ensuite, utilisez IBM Support Assistant pour repérer et installer les mises à jour de produits qui vous intéressent. Vous pouvez également choisir d'installer des plug-in pour d'autres logiciels IBM de votre environnement. Vous trouverez des informations complémentaires et la dernière version d'IBM Support Assistant à la page Web IBM Support Assistant (http://www.ibm.com/software/support/isa/).

Remarques

Les références aux produits, logiciels et services d'IBM n'impliquent pas qu'ils soient distribués dans tous les pays dans lesquels IBM exerce son activité. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. L'évaluation et la vérification de son fonctionnement en conjonction avec d'autres produits, hormis ceux expressément désignés par IBM, relèvent de la responsabilité de l'utilisateur.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, New York 10594 USA

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Corporation
Mail Station P300
522 South Road
Poughkeepsie, NY 12601-5400
USA
Attention: Information Requests

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Marques

Les termes suivant sont des marques d'IBM Corporation aux Etats-Unis et/ou dans certains autres pays :

- AIX
- CICS
- Cloudscape
- DB2
- Domino
- IBM
- Lotus
- RACF
- Redbooks
- Tivoli
- WebSphere
- z/OS

Java ainsi que tous les logos et toutes les marques incluant Java sont des marques de Sun Microsystems, Inc. aux Etats-Unis et/ou dans certains autres pays.

LINUX est une marque de Linus Torvalds aux Etats-Unis et/ou dans certains autres pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.

UNIX est une marque enregistrée de The Open Group aux Etats-Unis et dans certains autres pays.

Les autres noms de sociétés, de produits et de services peuvent appartenir à des tiers.

Index

G

H

I

J

L

M

O

P

Q

R

S

T

U

V

W

X

Z