9.4

Planificación de IBM MQ

Nota

Antes de utilizar esta información y el producto al que da soporte, lea la información en “Avisos” en la página 213.

Esta edición se aplica a la versión 9 release 4 de IBM® MQ y a todos los releases y modificaciones posteriores hasta que se indique lo contrario en nuevas ediciones.

Contenido

Planificación 5
Tipos de release de IBM MQ : consideraciones de planificación 6
Consideraciones de la instalación local de IBM MQ y IBM MQ Appliance para la preparación para el GDPR 9
Arquitecturas basadas en un único gestor de colas 18
Arquitecturas basadas en varios gestores de colas 19
Planificación de sus gestores de colas y clústeres distribuidos 20
Planificación de su red de publicación/suscripción distribuida 73
Planificación de los requisitos de almacenamiento y rendimiento en Multiplatforms 113
Requisitos de espacio de disco en Multiplatforms 114
Planificación del soporte del sistema de archivos en Multiplatforms 117
Planificación del soporte del sistema de archivos para MFT en Multiplatforms 145
Elección de registro circular o lineal en Multiplatforms 146
Memoria compartida en AIX 146
IBM MQ y los recursos IPC de UNIX System V 147
IBM MQ y prioridad de procesos en UNIX 147
Planning your IBM MQ environment on z/OS 147
Planning for your queue manager 148
Planning your channel initiator 177
Planning your queue sharing group (QSG) 181
Planning for backup and recovery 194
Planning your z/OS UNIX environment 203
Planning for Advanced Message Security 203
Planning for Managed File Transfer 204
Planning to use the IBM MQ Console and REST API on z/OS 210

Avisos 213
Información acerca de las interfaces de programación 214
Marcas registradas 215

Planificación de una arquitectura de IBM MQ

Cuando planifique su entorno de IBM MQ, tenga en cuenta el soporte que proporciona IBM MQ para las arquitecturas de uno o varios gestores de colas y para los estilos de mensajería de punto a punto y de publicación/suscripción. Además planifique los requisitos de recursos y su uso de los recursos de registro y copia de seguridad.

Acerca de esta tarea

Antes de planificar su arquitectura de IBM MQ, debe familiarizarse con los conceptos básicos de IBM MQ. Consulte Visión general técnica de IBM MQ.

Las arquitecturas de IBM MQ comprenden desde arquitecturas simples que utilizan un solo gestor de colas, hasta redes más complejas de gestores de colas interconectados. Se conectan varios gestores de colas entre sí utilizando técnicas de gestión de colas distribuidas. Para obtener más información sobre la planificación de las arquitecturas con un solo gestor de colas y con varios gestores de colas, consulte los temas siguientes :

- “Arquitecturas basadas en un único gestor de colas” en la página 18
- “Arquitecturas basadas en varios gestores de colas” en la página 19
  - “Planificación de sus gestores de colas y clústeres distribuidos” en la página 20
  - “Planificación de su red de publicación/suscripción distribuida” en la página 73

z/OS En IBM MQ for z/OS puede utilizar colas compartidas y grupos de compartición de colas para poder aplicar el reparto de la carga de trabajo y para que sus aplicaciones de IBM MQ puedan ser escalables y de alta disponibilidad. Para obtener información sobre las colas compartidas y los grupos de compartimiento de colas, consulte Colas compartidas y grupos de compartimiento de colas.

IBM MQ proporciona dos modelos de releases distintos:

- El release de Long Term Support (LTS) es el más adecuado para los sistemas que requieren un despliegue a largo plazo y la máxima estabilidad.
- El release de Continuous Delivery (CD) está pensado para sistemas que necesitan aprovechar rápidamente las últimas mejoras funcionales para IBM MQ.

Ambos tipos de release se instalan de la misma manera, pero existen consideraciones en relación al soporte y a la migración que debe tener en cuenta. Para obtener más información, consulte Tipos de release y mantenimiento de versiones deIBM MQ.

Para obtener más información acerca de cómo planificar varias instalaciones, los requisitos de almacenamiento y rendimiento y el uso de clientes, consulte los otros subtemas.

Conceptos relacionados

Tipos de release y mantenimiento de versiones de IBM MQ

“Planning your IBM MQ environment on z/OS” en la página 147
When planning your IBM MQ environment, you must consider the resource requirements for data sets, page sets, Db2, Coupling Facilities, and the need for logging, and backup facilities. Use this topic to plan the environment where IBM MQ runs.

Disponibilidad, recuperación y reinicio

Tareas relacionadas

Comprobar requisitos

Asegurarse de que no se han perdido mensajes (registro cronológico)

Tipos de release de IBM MQ : consideraciones de planificación

Los dos tipos de release principales para IBM MQ son Long Term Support (LTS) y un Continuous Delivery (CD). Para cada plataforma soportada, el tipo de release que elija afecta a la ordenación, instalación, mantenimiento y migración.

Para obtener información detallada sobre los tipos de release, consulte Tipos de release y mantenimiento de versiones deIBM MQ.

Consideraciones sobre IBM MQ for Multiplatforms

Pedido

Dentro de Passport Advantage hay dos eAssemblies independientes para IBM MQ 9.4. Uno contiene imágenes de instalación para el release de IBM MQ 9.4.0 Long Term Support, y el otro contiene imágenes de instalación para el release de IBM MQ 9.4.x Continuous Delivery. Descargue las imágenes de instalación del eAssembly según el release que prefiera.

Todas las versiones de IBM MQ pertenecen al mismo ID de producto y, para IBM MQ 9.4, los releases de LTS y los releases CD.

La titularidad para utilizar IBM MQ se extiende por todo el producto (PID), sujeto a las restricciones de los componentes con licencia y las métricas de precios. Esto significa que puede elegir libremente entre las imágenes de instalación del release de LTS y del release de CD para IBM MQ 9.4.

Instalación

Después de descargar una imagen de instalación de Passport Advantage, debe seleccionar para la instalación sólo los componentes para los que ha adquirido la titularidad. Consulte Información de licencia de IBM MQ para obtener más información sobre qué componentes instalables se incluyen para cada componente de pago.

Puede instalar el release de IBM MQ 9.4.0 LTS y el release de IBM MQ 9.4.x CD en la misma imagen del sistema operativo. Si lo hace, los componentes aparecerán como instalaciones independientes, según establece el soporte multiversión de IBM MQ. Cada versión tiene conjuntos distintos de gestores de colas asociados.

Cada nuevo release de CD se proporciona como una imagen de instalación. El nuevo release de CD se puede instalar junto con un release existente, o el instalador puede actualizar un release anterior de CD al nuevo release.

Los releases de CD contienen mejoras funcionales, así como el conjunto más reciente de arreglos de defectos y actualizaciones de seguridad. Cada release de CD es acumulativo y sustituye por completo todos los anteriores para dicha versión de IBM MQ. Por lo tanto, puede omitir un release de CD específico si no contiene ninguna función que sea relevante para la empresa.

Mantenimiento

El release de LTS es atendido por la aplicación de fixpacks, que proporcionan arreglos de defectos, y actualizaciones de seguridad acumulativas (CSU), que proporcionan parches de seguridad. Los fixpacks y las CSU están disponibles periódicamente y son acumulativos.

Para CD, las CSU se generan sólo para el último release de CD , que puede estar en una versión posterior.

Es posible que el equipo de soporte de IBM le indique ocasionalmente que aplique un arreglo temporal. Los arreglos temporales también se conocen como arreglos de emergencia o de prueba, y se utilizan para aplicar actualizaciones urgentes que no pueden esperar a la siguiente entrega de mantenimiento.

Migración entre el release de LTS y el release de CD

Existen restricciones y limitaciones, pero, normalmente, un gestor de colas individual se puede migrar del uso de código de release de LTS a código de release de CD o del uso de código de release de CD a código de release de LTS, a condición de que el release de destino sea superior al utilizado antes de la migración.

Hay dos métodos posibles:

- Instalar el nuevo release de código in situ, de manera que se actualice la instalación existente de IBM MQ. Los gestores de colas asociados con la instalación utilizan el nuevo release de código al iniciarse.
- Instalar el nuevo release de código como una instalación nueva y, a continuación, mover las instancias de gestor de colas individuales a la nueva instalación mediante el mandato setmqm.

Cuando un gestor de colas empieza a ejecutar un release CD de código, el nivel de mandatos del gestor de colas se actualiza para indicar el nuevo nivel de release. Esto significa que las nuevas funciones proporcionadas en el release están habilitadas y que ya no puede reiniciar el gestor de colas utilizando un release de código con un número VRM inferior.

Consideraciones sobre IBM MQ for z/OS

z/OS

Pedido

Al solicitar IBM MQ for z/OS 9.4, se ofrecen dos características independientes en ShopZ. Las características corresponden al release de LTS y al release de CD. Ambas características se aplican al mismo ID de producto (PID). Se trata del ID de producto con licencia, por lo que, donde exista licencia para una característica, habrá titularidad para utilizar la característica alternativa en caso de ser necesario. Al realizar el pedido, seleccione la característica correspondiente con el release de LTS o el release de CD .

Si está seleccionando productos para su inclusión en un ServerPac, no puede elegir el release de LTS y el release de CD en el mismo orden ServerPac , porque SMP/E no puede instalar los productos en la misma zona de destino.

Instalación

Los releases de LTS y CD se proporcionan en conjuntos independientes de FMID. Tenga en cuenta que estos FMID no se pueden instalar en la misma zona de destino SMP/E. Si necesita los releases LTS y CD :

- Instale el release de LTS y el release de CD en zonas de destino separadas.
- Mantenga bibliotecas de destino y distribución separadas para los dos releases.

Si el gestor de colas está en un grupo de compartición de colas, al actualizar a la última versión de CD debe actualizar todos los gestores de colas del grupo.

El nivel de comando de un gestor de colas es el nivel VRM de tres dígitos. UnIBM MQ el programa puede llamar`MQINQ` , pasando el`MQIA_COMMAND_LEVEL` selector, para obtener el nivel de comando del administrador de colas al que está conectado.

Puesto que los releases utilizan distintos FMID, no puede actualizar un release de CD con mantenimiento para un release de LTS o al revés. De forma similar, no hay forma de conmutar una versión del código de producto de un release de LTS a un release de CD o viceversa. Sin embargo, puede conmutar un gestor de colas entre los modelos de release. Consulte Migración entre el release de LTS y el release de CD.

Nota:

Los releases de IBM MQ 9.0.x y IBM MQ 9.1.x CD tenían FMID dependientes de versión y release separados. Por lo tanto, pasar de 9.0.x CD a 9.1.x CD requería al menos una instalación de SMP/E completa.

A partir de IBM MQ for z/OS 9.2.0, el release de CD utiliza un conjunto de FMID que siguen siendo los mismos para todos los releases de IBM MQ for z/OS con un número de versión de 9. Puesto que cada nueva versión de IBM MQ está disponible como un release de CD y LTS , puede actualizar los releases de CD aplicando los PTF a una única instalación de SMP/E incluso cuando se cruza un límite de versión principal. Por ejemplo, puede ir de IBM MQ for z/OS 9.2.0 CD, a IBM MQ for z/OS 9.2.2 CD, a IBM MQ for z/OS 9.2.4 CD, a IBM MQ for z/OS 9.3.0 CD, simplemente aplicando los PTF.

Para distinguir entre un release de LTS y uno de CD con el mismo nivel de VRM, consulte el mensaje CSQY000I en el registro de trabajo del gestor de colas.

Mantenimiento

IBM MQ for z/OS utiliza PTF para mantenimiento.

LTS Los PTF son específicos de un conjunto determinado de bibliotecas correspondientes a un nivel de release específico. Para las características de UNIX System Services (es decir, JMS y WEB UI, Connector Pack y Managed File Transfer), los PTF de z/OS se alinean directamente con los fixpacks de Multiplatforms y las actualizaciones de seguridad acumulativas (CSU). Estos arreglos son acumulativos y están disponibles al mismo tiempo que el fixpack o CSU de Multiplatforms equivalente.

CD Las CSU de CD no suelen estar disponibles entre releases de CD, pero se incluyen en el siguiente release de IBM MQ for z/OS CD . También puede ponerse en contacto con el soporte para solicitar un + + USERMOD.

Otros arreglos en IBM MQ for z/OS son arreglos distintos en partes concretas. Estos arreglos resuelven problemas específicos, no son acumulativos y están disponibles a medida que se producen.

Migración entre el release de LTS y el release de CD

Existen restricciones y limitaciones, pero normalmente se puede migrar un único gestor de colas del uso de código de release de LTS a código de release de CD o del uso de código de release de CD a código de release de LTS a condición de que el release de destino sea más alto que el utilizado antes de la migración.

A partir de IBM MQ for z/OS 9.2.0, puede migrar de ida y vuelta entre los releases de CD y LTS con el mismo VRM tantas veces como sea necesario, y sin que ello afecte a la capacidad de migrar hacia atrás. Por ejemplo, un gestor de colas se puede iniciar en IBM MQ for z/OS 9.3.0 LTSy, a continuación, concluir e iniciar en IBM MQ for z/OS 9.3.0 CDy, a continuación, concluir e iniciar en IBM MQ for z/OS 9.3.0 LTS.

IBM MQ for z/OS tradicionalmente ha proporcionado una capacidad de reserva (migración a versiones anteriores) para que después de un periodo de ejecución posterior a una migración pueda volver al release anterior. Esta prestación se conserva para los releases de LTS y aquellos releases de CD con un modificador de `0` como 9.3.0 CD, pero no es posible cuando el origen o destino de una migración es un release de CD con un número de modificador distinto de cero, por ejemplo, 9.2.5 o 9.3.1.

A continuación se muestran escenarios de migración válidos e ilustran cómo funciona este principio:

Release de origen	Release de destino	Notas
9.1.0 LTS	9.4.0 LTS o 9.4.0 CD	La migración a versiones anteriores no está soportada ya que 9.1.0 LTS está fuera del soporte estándar.
9.2.0 LTS	9.4.0 LTS o 9.4.0 CD	Se admite la migración a versiones anteriores.
9.3.0 LTS	9.4.0 LTS o 9.4.0 CD	Se admite la migración a versiones anteriores.
9.3.5 CD	9.4.0 LTS o 9.4.0 CD	No se admite la migración a versiones anteriores ya que el release de origen es CD con un modificador distinto de cero.
9.4.0 LTS o 9.4.0 CD	9.4.1 CD	No se admite la migración a versiones anteriores ya que el release de destino es CD con un modificador distinto de cero. `Write to operator with reply` CSQY041D se emite para confirmar la migración.

Tareas relacionadas


Aplicación y eliminación de mantenimiento en z/OS

Información relacionada

DescargandoIBM MQ 9.4

Consideraciones de la instalación local de IBM MQ y IBM MQ Appliance para la preparación para el GDPR

Para los PID(s):

Distribuido

- IBM MQ/IBM MQ Advanced - 5724-H72
- IBM MQ for HPE NonStop - 5724-A39

z/OS

- IBM MQ for z/OS - 5655-MQ9
- IBM MQ for z/OS Value Unit Edition - 5655-VU9
- IBM MQ Advanced for z/OS - 5655-AV9
- IBM MQ Advanced for z/OS Value Unit Edition - 5655-AV1

IBM MQ Appliance

- IBM MQ Appliance M2003 -5900-ALJ
- IBM MQ Appliance M2002 - 5737-H47

Aviso:

Este documento se ha diseñado para ayudarle en los preparativos para la preparación para GDPR. Proporciona información sobre las características de IBM MQ que se pueden configurar, así como aspectos del uso del producto que se deben tener en cuenta para ayudar a la organización a prepararse para el GDPR. Esta información es una lista exhaustiva, debido a las muchas formas en las que los clientes pueden elegir y configurar características, y la gran variedad de formas en las que se puede utilizar el producto con el propio producto y con sistemas y aplicaciones de terceros.

Los clientes son responsables de garantizar su propio cumplimiento con distintas leyes y normativas, incluyendo el Reglamento General de Protección de Datos de la Unión Europea. Los clientes son los únicos responsables de obtener asesoramiento legal competente referente a la identificación y la interpretación de cualquier ley relevante que pudiera afectar a su negocio, así como cualquier medida que debieran tomar para cumplir con dichas leyes y normativas.

Los productos, servicios y otras funciones descritos en este documento no son adecuados para todas las situaciones de cliente y pueden tener la disponibilidad restringida. IBM no proporciona asesoramiento legal, contabilidad o de auditoría ni afirma ni garantiza que sus servicios o productos garantizarán que los clientes son compatibles con cualquier ley o normativa.

Tabla de contenido

1. RGPD
2. Configuración de producto para GDPR
3. Ciclo de vida de los datos
4. Recopilación de datos
5. Almacenamiento de datos
6. Acceso a los datos

7. Proceso de datos
8. Supresión de datos
9. Supervisión de datos
10. Prestación para restringir el uso de datos personales
11. Manejo de archivos

GDPR

El Reglamento General de Protección de Datos (GDPR) ha sido adoptado por la Unión Europea ("UE") y se aplica desde el 25 de mayo de 2018.

¿Por qué es importante el GDPR?

GDPR establece un marco regulatorio de protección de datos más potente para el proceso de datos personales individuales. GDPR ofrece:

- Nuevos y mejores derechos para las personas
- Una definición más amplia de datos personales
- Nuevas obligaciones para procesadores
- Posibles penalizaciones financieras significativas por el incumplimiento
- Notificación obligatoria de la filtración de datos

Obtenga más información sobre el GDPR:

- Portal de la información del GDPR UE
- Sitio web ibm.com/GDPR

Configuración del producto - Consideraciones para la preparación para GDPR

Las secciones siguientes proporcionan consideraciones para configurar IBM MQ para ayudar a su organización en la preparación para el GDPR.

Ciclo de vida de los datos

IBM MQ es un producto de middleware orientado al mensaje transaccional que permite que las aplicaciones intercambien de forma asíncrona los datos proporcionados de la aplicación. IBM MQ admite una gama de distintas API de mensajería, protocolos y puentes con la finalidad de conectar aplicaciones. A tal efecto, IBM MQ se puede utilizar para intercambiar muchas formas de datos, algunos de los cuales podrían estar potencialmente sujetos al GDPR. Existen varios productos de terceros con los cuales IBM MQ podría intercambiar datos. Algunos de estos son propiedad de IBM, pero muchos otros son proporcionados por otros proveedores de tecnología. El sitio web Informes de compatibilidad de producto de software proporciona listas del software asociado. Para obtener las consideraciones con respecto a la preparación para GDPR de un producto de terceros, deberá consultar la documentación de ese producto. Los administradores de IBM MQ controlan la forma en la que IBM MQ interactúa con los datos que se pasan a través de él, mediante la definición de colas, temas y suscripciones.

¿Qué tipos de datos fluyen a través de IBM MQ?

Puesto que IBM MQ proporciona servicio de mensajería asíncrona para datos de aplicación, no hay ninguna respuesta definitiva a esta pregunta porque los casos prácticos varían a través del despliegue de aplicaciones. Los datos de mensaje de aplicación se conservan en archivos de cola (conjuntos de páginas o el recurso de acoplamiento en z/OS), registros y archivados y el propio mensaje puede contener datos regidos por el GDPR. Los datos de mensaje proporcionados por la aplicación también pueden estar incluidos en archivos recopilados para fines de determinación de problemas como, por ejemplo, registros de error, archivos de rastreo y FFST. En z/OS, los datos de mensaje proporcionados por la aplicación también pueden estar incluidos en el espacio de direcciones o en volcados del recurso de acoplamiento.

A continuación, se muestran algunos ejemplos típicos de datos personales que se pueden intercambiar utilizando IBM MQ:

- Los empleados del cliente (por ejemplo; IBM MQ se podría utilizar para conectarse a los sistemas de nómina o RR. HH. del cliente).
- Los datos personales de los propios clientes del cliente (por ejemplo; IBM MQ podrían ser utilizados por un cliente para intercambiar datos entre aplicaciones que están relacionadas con sus clientes como, por ejemplo, tomar oportunidades de ventas y almacenar datos dentro de su sistema CRM).
- Los datos personales confidenciales de los propios clientes del cliente (por ejemplo, IBM MQ se podría tener que utilizar dentro de contextos del sector que requieren que se intercambien datos personales como, por ejemplo, registros de atención sanitaria basados en HL7 al integrar aplicaciones clínicas).

Además de los datos de mensaje proporcionados por la aplicación, IBM MQ procesa los tipos de datos siguientes:

- Credenciales de autenticación (como nombre de usuario y contraseña, claves de API, etc.)
- Información personal identificable técnicamente (como ID de dispositivo, identificadores basados en el uso, dirección IP, etc., cuando están vinculados a una persona)

Datos personales utilizados para el contacto en línea con IBM

Los clientes de IBM MQ pueden enviar comentarios/opiniones/solicitudes en línea para ponerse en contacto con IBM sobre temas de IBM MQ de muchas formas distintas, principalmente:

- Área de comentarios públicos en páginas del área de IBM MQ en IBM Developer
- Área de comentarios públicos en páginas de Información del producto IBM MQ en IBM Documentation
- Comentarios públicos en el foro de IBM Support
- Comentarios públicos en Ideas de integración de IBM

Normalmente, solo se utilizan el nombre de cliente y la dirección de correo electrónico para habilitar las respuestas personales para el asunto del contacto y el uso de datos personales se ajusta a Declaración de privacidad en línea de IBM.

Recopilación de datos

IBM MQ se puede utilizar para recopilar datos personales. Al evaluar el uso de IBM MQ y sus necesidades para satisfacer las demandas del GDPR, deberá tener en cuenta los tipos de datos personales que, en sus circunstancias, se están pasando a través de IBM MQ. Es posible que desee tener en cuenta aspectos como, por ejemplo:

- ¿Cómo llegan los datos a los gestores de colas? (¿Entre qué protocolos? ¿Los datos están cifrados? ¿Los datos están firmados?)
- ¿Cómo se envían los datos desde los gestores de colas? (¿Entre qué protocolos? ¿Los datos están cifrados? ¿Los datos están firmados?)
- ¿Cómo se almacenan los datos cuando pasan a través de un gestor de colas? (Cualquier aplicación de mensajería tiene la capacidad de escribir datos de mensaje en un soporte con estado, incluso aunque un mensaje no sea persistente. ¿Está informado de cómo las características de mensajería podrían exponer posiblemente aspectos de los datos de mensaje de aplicación que se pasan a través del producto?)
- ¿Cómo se recopilan y almacenan las credenciales cuando son necesarias para IBM MQ para acceder a aplicaciones de terceros?

Es posible que IBM MQ tenga que comunicarse con otros sistemas y servicios que requieren la autenticación, por ejemplo, LDAP. Cuando sea necesario, IBM MQ configura y almacena los datos de autenticación (ID de usuario, contraseñas) para ser utilizados en dichas comunicaciones. Siempre que sea posible, deberá evitar el uso de credenciales personales para la autenticación de IBM MQ. Considere la protección del almacenamiento utilizado para los datos de autenticación. (Consulte el Almacenamiento de datos más abajo.)

Almacenamiento de datos

Cuando los datos de mensaje viajan a través de gestores de colas, IBM MQ conservará (quizás varias de copias de) esos datos directamente en un soporte con estado. Es posible que los usuarios de IBM MQ deseen tener en cuenta proteger los datos de mensaje mientras están en reposo.

Los elementos siguientes destacan áreas donde IBM MQ conserva datos proporcionados por la aplicación, que los usuarios pueden desear tener en cuenta al garantizar la compatibilidad con el GDPR.

- Colas de mensajes de aplicación:

  IBM MQ proporciona colas de mensaje para permitir el intercambio de datos asíncrono entre aplicaciones. Los mensajes no persistentes y persistente almacenados en una cola se escriben en un soporte con estado.
- Colas de agente de transferencia de archivos:

  IBM MQ Managed File Transfer utiliza colas de mensaje para coordinar la transferencia fiable de datos de archivo, archivos que contienen datos personales y los registros de transferencias se almacenan en estas colas.
- Colas de transmisión:

  Para transferir mensajes con fiabilidad entre gestores de colas, los mensajes se almacenan temporalmente en colas de transmisión.
- Colas de mensajes no entregados:

  Existen algunas circunstancias bajo las cuales los mensajes no se pueden colocar en una cola de destino y se almacenan en una cola de mensajes no entregados, si se ha configurado una cola de ese tipo en el gestor de colas.
- Colas de retirada:

  Las interfaces de mensajería JMS y XMS proporcionan una función que permite mover los mensajes con formato incorrecto a una cola de retirada después de que se haya producido una serie de restituciones para permitir que se procesen otros mensajes válidos.
- Cola de error AMS:

  IBM MQ Advanced Message Security moverá los mensajes que no cumplan con una política de seguridad a SYSTEM.PROTECTION.ERROR.QUEUE de forma similar a la cola de mensajes no entregados.
- Publicaciones retenidas:

  IBM MQ proporciona una característica de publicación retenida para permitir a las aplicaciones de suscripción recuperar una publicación anterior.
- Entrega aplazada:

  IBM MQ da soporte a la característica de retardo de entrega JMS 2.0 y Jakarta Messaging 3.0 que permite que los mensajes se entreguen a su destino en un momento futuro. Los mensajes que todavía no se han entregado se almacenan en la cola SYSTEM.DDELAY.LOCAL.QUEUE.

Más información:

- Registro: Asegurarse de que no se han perdido mensajes
- Valores de cola del agente MFT
- Utilización de la cola de mensajes no entregados
- Manejo de mensajes con formato incorrecto en clases IBM MQ para JMS
- Manejo de errores AMS
- Publicaciones retenidas
- Retardo de entrega deJMS 2.0

Los elementos siguientes destacan áreas donde IBM MQ puede persistir indirectamente datos proporcionados por la aplicación que los usuarios también pueden desear tener en cuenta al garantizar la compatibilidad con el GDPR.

- Mensajería de ruta de rastreo:

  IBM MQ proporciona prestaciones de ruta de rastreo, que registran la ruta que sigue un mensaje entre aplicaciones. Los mensajes de suceso generados pueden incluir información personal identificable técnicamente como, por ejemplo, direcciones IP.

- Rastreo de actividad de aplicación:

  IBM MQ proporciona rastreo de actividad de aplicación, que registran las actividades de la API de mensajería de aplicaciones y canales, el rastreo de actividad de aplicación puede registrar el contenido de los datos de mensaje proporcionados por la aplicación en mensajes de suceso.

- Rastreo de servicio:

  IBM MQ proporciona características de rastreo de servicio, que registran las vías de acceso de código interno a través de las cuales fluyen los datos de mensaje. Como parte de estas características, IBM MQ puede registrar el contenido de datos de mensaje proporcionados por la aplicación en archivos de rastreo almacenados en disco.

- Sucesos de gestor de colas:

  IBM MQ puede generar mensajes de suceso que podrían incluir datos personales como, por ejemplo, sucesos de autorización, mandato y configuración.

Más información:

- Mensajería de ruta de rastreo
- Utilización del rastreo
- Supervisión de sucesos
- Sucesos de gestor de colas

Para proteger el acceso a copias de los datos de mensaje proporcionados por la aplicación, tenga en cuenta las acciones siguientes:

- Restrinja el acceso de usuario privilegiado a los datos de IBM MQ en el sistema de archivos, por ejemplo, restringiendo la pertenencia de usuario del grupo 'mqm' en plataformas UNIX and Linux®.
- Restrinja el acceso de aplicación a datos de IBM MQ a través de colas dedicadas y el control de accesos. Cuando sea adecuado evite el uso compartido innecesario de recursos como, por ejemplo, colas entre aplicaciones y proporcione un control de accesos granular a los recursos de cola y tema.
- Restrinja el acceso a copias replicadas de datos de IBM MQ en configuraciones de alta disponibilidad (HA) o recuperación tras desastre (DR), y proteja las conexiones utilizadas para la réplica.
- Utilice IBM MQ Advanced Message Security para proporcionar una firma y/o cifrado integral de datos de mensaje.
- Utilice el cifrado a nivel de archivo o volumen para proteger directorios o sistemas de archivos que pueden contener datos, rastreo o registros de IBM MQ .
- Después de cargar el rastreo del servicio en IBM, puede suprimir archivos de rastreo de servicio y datos FFST, si está preocupado sobre el contenido que posiblemente contiene datos personales.

Más información:

- Usuarios privilegiados
- Planificación del soporte del sistema de archivos en Multiplatforms
- Cifrado del sistema de archivos en IBM MQ Appliance

Un administrador de IBM MQ puede configurar un gestor de colas con credenciales (nombre de usuario y contraseña, claves de API, etc.) para servicios de terceros, como LDAP. Estos datos se almacenan generalmente en el directorio de datos del gestor de colas protegido mediante permisos del sistema de archivos.

Cuando se crea un gestor de colas IBM MQ, el directorio de datos se configura con un control de acceso basado en grupos, de forma que IBM MQ puede leer los archivos de configuración y utilizar las credenciales para conectarse a estos sistemas. Los administradores de IBM MQ se consideran usuarios privilegiados y son miembros de este grupo, de forma que tienen acceso de lectura a los archivos. Algunos archivos se enmascaran, pero no se cifran. Por este motivo, para proteger por completo el acceso a credenciales, deberá tener en cuenta las acciones siguientes:

- Restrinja el acceso del usuario privilegiado a los datos de IBM MQ, por ejemplo, restringiendo la pertenencia del grupo 'mqm' en plataformas UNIX and Linux.
- Utilice el cifrado de nivel de archivo o volumen para proteger el contenido del directorio de datos de gestor de colas.
- Cifre las copias de seguridad del directorio de configuración de producción y almacénelas con los controles de accesos apropiados.
- Considere proporcionar seguimientos de auditoría para el error de autenticación, el control de accesos y cambios de configuración con sucesos de seguridad, mandato y configuración.

Más información:

- Protección de IBM MQ

Acceso a datos

Se puede acceder a los datos del gestor de colas IBM MQ a través de las interfaces de producto siguientes, algunas de las cuales se han diseñado para acceder a través de una conexión remota, y otras para acceder a través de una conexión local.

- Consola IBM MQ [Solo remoto]
- API REST administrativa de IBM MQ [Solo remoto]
- API REST de mensajería de IBM MQ [Solo remoto]
- MQI [Local y remoto]
- JMS [Local y remoto]
- XMS [Local y remoto]
- IBM MQ Telemetry (MQTT) [Solo remoto]
- IBM MQ Light (AMQP) [Solo remoto]
- Puente IMS de IBM MQ [Solo local]
- Puente CICS de IBM MQ [Solo local]
- Puentes de protocolo MFT de IBM MQ [Solo remoto]
- Puentes Connect:Direct de IBM MQ [Solo remoto]
- IBM MQ MQAI [Local y remoto]
- Mandatos PCF de IBM MQ [Local y remoto]
- Mandatos MQSC de IBM MQ [Local y remoto]
- IBM MQ Explorer [Local y remoto]
- Salidas de usuario de IBM MQ [Solo local]
- IBM MQ Internet Pass-Thru [Solo remoto]
- Métricas de Red Hat® OpenShift® Monitoring (Prometheus) (las métricas son datos numéricos sobre las estadísticas del gestor de colas)
- Consola en serie de IBM MQ Appliance [Solo local]
- SSH de IBM MQ Appliance [Solo remoto]
- API REST IBM MQ Appliance [Solo remoto]
- Interfaz de usuario web de IBM MQ Appliance [Solo remoto]

- V 9.4.0 IBM MQ Kafka Connectors (Kafka Connect) [Local y remoto]

Las interfaces se han diseñado para permitir a los usuarios realizar cambios en un gestor de colas IBM MQ y en los mensajes almacenados en él. Las operaciones de administración y mensajería están protegidas de forma que haya tres etapas implicadas cuando se realiza una solicitud:

- Autenticación
- Correlación de roles
- Autorización

Autenticación

Si se ha solicitado el mensaje o la operación administrativa desde una conexión local, el origen de esta conexión es un proceso en ejecución en el mismo sistema. El usuario que ejecuta el proceso debe haber pasado los pasos de autenticación proporcionados por el sistema operativo. El nombre de usuario del propietario del proceso desde el cual se ha realizado la conexión se confirma como la identidad. Por ejemplo, esto podría ser el nombre del usuario que ejecuta el shell desde el cual se ha iniciado una aplicación. Las formas posibles de autenticación para las conexiones locales son:

1. Nombre de usuario certificado (SO local)
2. Nombre de usuario y contraseña opcionales (SO, LDAP o repositorios de terceros personalizados)
3. Sólo señal de seguridad (JWT) IBM MQ

Si la acción administrativa se ha solicitado desde una conexión remota, las comunicaciones con IBM MQ se realizan a través de una interfaz de red. Las formas de identidad siguientes se pueden presentar para la autenticación a través de conexiones de red:

1. Nombre de usuario certificado (de SO remoto)
2. Nombre de usuario y contraseña (SO, LDAP o repositorios de terceros personalizados)
3. Dirección de red de origen (como una dirección IP)
4. Certificado digital X.509 (autenticación SSL/TLS mutua)
5. Señales de seguridad (como la señal LTPA2 o la señal JWT)
6. Otra seguridad personalizada (prestación proporcionada por salidas de terceros)
7. Claves SSH

La integración de IBM MQcon IBM Cloud Pak for Integration añade un nuevo tipo de autenticación para IBM MQ Console: Single Sign-On con Cloud Pak. (solo CP4I)

Correlación de roles:

En la etapa de correlación de roles, las credenciales que se han proporcionado en la etapa de autenticación se pueden correlacionar con un identificador de usuario alternativo. Siempre que se permita continuar el identificador de usuario correlacionado (por ejemplo, los usuarios administrativos podrían estar bloqueados por reglas de autenticación de canal), el ID de usuario correlacionado se transporta hasta la etapa final cuando se autorizan actividades con respecto a recursos de IBM MQ.

Autorización:

IBM MQ proporciona la capacidad de que distintos usuarios tengan distintas autorizaciones con respecto a recursos de mensajería diferentes como, por ejemplo, colas, temas y otros objetos de gestor de colas.

Actividad de registro:

Es posible que algunos usuarios de IBM MQ necesiten crear un registro de auditoría de acceso a recursos MQ. Los ejemplos de registros de auditoría deseables podrían incluir cambios de configuración que contienen información sobre el cambio, además de quién lo ha solicitado.

Los orígenes de información siguientes están disponibles para implementar este requisito:

1. Se puede configurar un gestor de colas IBM MQ para generar sucesos de mandato cuando se ha ejecutado correctamente un mandato administrativo.

2. Se puede configurar un gestor de colas IBM MQ para generar sucesos de configuración cuando se crea, modifica o suprime un recurso de gestor de colas.
3. Se puede configurar un gestor de colas IBM MQ para generar un suceso de autoridad cuando una comprobación de autorización falla para un recurso.
4. Los mensajes de error que indican comprobaciones de autorización fallidas se escriben en los registros de error del gestor de colas.
5. La consola IBM MQ escribirá mensajes de auditoría en sus registros cuando fallan la autenticación, las comprobaciones de autorización o cuando se crean, inician, detienen o suprimen gestores de colas.
6. IBM MQ Appliance escribirá los mensajes de auditoría en sus registros para registrar los cambios del sistema y los inicios de sesión de usuario.

Al considerar estos tipos de soluciones, es posible que los usuarios de IBM MQ deseen prestar atención a los puntos siguientes:

- Los mensajes de suceso no son persistentes, por lo que cuando un gestor de colas se reinicia, se pierde la información. Los supervisores de sucesos se deben configurar para consumir de forma constante los mensajes disponibles y transferir el contenido a soporte persistente.
- Los usuarios privilegiados de IBM MQ tienen privilegios suficientes para inhabilitar sucesos, borrar registros o suprimir gestores de colas.

Si desea más información sobre cómo proteger el acceso a los datos de IBM MQ y proporcionar un seguimiento de auditoría, consulte los temas siguientes:

- Mecanismos de seguridad de IBM MQ
- Sucesos de configuración
- Sucesos de mandato
- Utilización de registros de errores

Proceso de datos

Cifrado mediante una infraestructura de claves públicas:

Puede proteger las conexiones de red a IBM MQ especificando que las conexiones utilizan TLS, que también puede proporcionar una autenticación mutua del extremo iniciador de la conexión.

Utilizar los recursos de seguridad de PKI proporcionados por mecanismos de transportes es el primer paso hacia la protección del proceso de datos con IBM MQ. Sin embargo, sin habilitar más características de seguridad, el comportamiento de una aplicación consumidora es procesar todos los mensajes entregados a la misma sin validar el origen del mensaje, o si se ha modificado durante el tránsito.

Los usuarios de IBM MQ que tienen licencia para utilizar las prestaciones Advanced Message Security (AMS) pueden controlar la forma en la que las aplicaciones procesan los datos personales contenidos en mensajes, a través de la definición y configuración de políticas de seguridad. Las políticas de seguridad permiten que se aplique la firma y/o cifrado a los datos de mensaje entre aplicaciones.

Es posible utilizar políticas de seguridad para requerir y validar una firma digital al consumir mensajes para asegurarse de que los mensajes son auténticos. El cifrado AMS proporciona un método a través del cual los datos de mensaje se convierten de un formato legible a una versión codificada que solo puede descodificarla otra aplicación, si esta es la destinataria prevista o el mensaje y si tiene acceso a la clave de descifrado correcta.

Si desea más información sobre cómo utilizar SSL y certificados para proteger las conexiones de red, consulte los temas siguientes en la documentación del producto IBM MQ:

- Configuración de la seguridad TLS para IBM MQ
- Descripción general de AMS

Supresión de datos

IBM MQ proporciona mandatos y acciones de interfaz de usuario para suprimir datos que se han proporcionado al producto. Esto significa que los usuarios de IBM MQ pueden suprimir datos que están relacionados con personas concretas, en caso de ser necesario.

- Áreas del comportamiento de IBM MQ para tener en cuenta con objeto de cumplir con la supresión de datos de cliente del GDPR.
  - Suprima datos de mensaje almacenados en una cola de aplicación:
    - Eliminando mensajes individuales mediante la API de mensajería o las herramientas o utilizando la caducidad de mensajería.
    - Especificando que los mensajes no son persistentes, que se incluyen en una cola donde la clase de mensaje no persistente es normal y reiniciando el gestor de colas.
    - Borrando la cola de forma administrativa.
    - Suprimiendo la cola.
  - Suprima los datos de publicación retenida almacenados en un tema:
    - Especificando que los mensajes no son persistente y reiniciando el gestor de colas.
    - Sustituyendo los datos retenidos con datos nuevos o utilizando la caducidad del mensaje.
    - Borrando la serie de tema de forma administrativa.
  - Suprima los datos almacenados en un gestor de colas suprimiendo todo el gestor de colas y las copias replicadas para alta disponibilidad o recuperación tras desastre.
  - Suprima los datos almacenados por los mandatos de rastreo de servicio suprimiendo los archivos del directorio de rastreo.
  - Suprima los datos FFST almacenados suprimiendo los archivos en el directorio de errores.
  - Suprima el espacio de direcciones y los volcados del recurso de acoplamiento (en z/OS).
  - Suprima el archivado, la copia de seguridad u otras copias de dichos datos.
- Áreas del comportamiento de IBM MQ para tener en cuenta con objeto de cumplir con la supresión de datos de cuenta del GDPR
  - Puede suprimir datos y preferencias de cuenta almacenados por IBM MQ para conectarse a gestores de colas y servicios de terceros suprimiendo (incluyendo archivado, copia de seguridad o copias duplicadas de cualquier otra forma de los mismos):
    - Objetos de información de autenticación de gestor de colas que almacenan credenciales.
    - Registros de autoridad de gestor de colas que hacen referencia a identificadores de usuario.
    - Reglas de autenticación de canal de gestor de colas que se correlacionan con o bloquean direcciones IP, identificadores de DN de certificado o identificadores de usuario específicos.
    - Archivos de credenciales utilizados por el agente, registrador y el plugin MFT de MQ Explorer de IBM MQ Managed File Transfer para la autenticación con el gestor de colas y servidores de archivos.
    - Certificados digitales X.509 que representan o contienen información sobre una persona de almacenes de claves que pueden utilizar conexiones SSL/TLS o IBM MQ Advanced Message Security (AMS).
    - Las cuentas de usuario individual de IBM MQ Appliance, que incluyen la referencia a estas cuentas en archivos de registro del sistema.
    - Valores de metadatos de espacio de trabajo IBM MQ Explorer y Eclipse
    - Almacén de contraseñas de IBM MQ Explorer tal como se especifica en Preferencias de contraseña.
    - Archivos de configuración de la consola IBM MQ y el servidor mqweb
    - Archivos de configuración y almacenes de claves de IBM MQ Internet Pass-Thru.

Más información:

- Autenticación de conexión de MFT e IBM MQ
- Correlación de credenciales para un servidor de archivos utilizando el archivo ProtocolBridgeCredentials.xml
- Configuración de usuarios y roles de IBM MQ Console

Supervisión de datos

IBM MQ proporciona un rango de características de supervisión que los usuarios pueden explotar para obtener una mejor comprensión de cómo se comportan las aplicaciones y los gestores de colas.

IBM MQ también proporciona una serie de características que ayudan a gestionar registros de errores del gestor de colas.

Más información:

- Supervisión de la red IBM MQ
- Servicios de mensajes de diagnóstico
- Servicio QMErrorLog
- Supervisión y creación de informes de IBM MQ Appliance

Capacidad para restringir el uso de datos personales

Utilizando los recursos resumidos en este documento, IBM MQ permite a un usuario final restringir el uso de sus datos personales.

Las colas de mensaje IBM MQ no se deben utilizar como un almacén de datos permanentes de la misma forma que una base de datos, lo que es especialmente cierto cuando se manejan datos de aplicación que están sujetos al GDPR.

A diferencia de una base de datos, donde los datos se pueden encontrar a través de una consulta de búsqueda, puede ser difícil encontrar datos de mensaje, a menos que conozca los identificadores de cola, mensaje y correlación de un mensaje.

Siempre que los mensajes que contienen los datos de una persona se puedan identificar y localizar fácilmente, es posible utilizar características de mensajería IBM MQ estándar para acceder o modificar datos de mensaje.

Manejo de archivos

1. IBM MQ Managed File Transfer no realiza exploración de programas maliciosos en los archivos transferidos. Los archivos se transfieren tal cual y se realiza una comprobación de integridad para garantizar que los datos de los archivos no se han modificado durante la transferencia. Las sumas de comprobación de origen y destino se publican como parte de la publicación del estado de transferencia. Se recomienda que los usuarios finales implementen la exploración de programas maliciosos según corresponda para su entorno antes de que MFT transfiera el archivo y después de que MFT entregue un archivo a un punto final remoto.
2. IBM MQ Managed File Transfer no realiza ninguna acción según el tipo MIME o la extensión de archivo. MFT lee los archivos y transfiere los bytes exactamente tal como se leen del archivo de entrada.

Arquitecturas basadas en un único gestor de colas

Las arquitecturas de IBM MQ más sencillas implican la configuración y el uso de un único gestor de colas.

Antes de planificar su arquitectura de IBM MQ, debe familiarizarse con los conceptos básicos de IBM MQ. Consulte Visión general técnica de IBM MQ.

En los siguientes apartados se describen varias arquitecturas posibles que utilizan un único gestor de colas:

- “Gestor de colas individual con aplicaciones locales que acceden a un servicio” en la página 19

- “Gestor de colas individual con aplicaciones remotas que acceden a un servicio como clientes” en la página 19
- “Gestor de colas individual con una configuración de publicación/suscripción” en la página 19

Gestor de colas individual con aplicaciones locales que acceden a un servicio

La primera arquitectura se basa en un solo gestor de colas donde las aplicaciones acceden a un servicio que se están ejecutando en el mismo sistema que las aplicaciones que proporciona el servicio. Un gestor de colas de IBM MQ proporciona intercomunicación asíncrona entre las aplicaciones que solicitan el servicio y las aplicaciones que suministran el servicio. Esto significa que la comunicación entre las aplicaciones pueden continuar incluso si una de las aplicaciones está fuera de línea durante un largo período de tiempo.

Gestor de colas individual con aplicaciones remotas que acceden a un servicio como clientes

La segunda arquitectura se basa en un único gestor de colas cuyas aplicaciones se ejecutan de forma remota desde las aplicaciones que suministran el servicio. Las aplicaciones remotas se ejecutan en distintos sistemas para los servicios. Las aplicaciones se conectan como clientes con el gestor de colas individual. Esto significa que puede proporcionarse acceso a un servicio a varios sistemas a través de un solo gestor de colas.

Una limitación de esta arquitectura es que debe estar disponible una conexión de red para que una aplicación funcione. La interacción entre la aplicación y el gestor de colas a través de la conexión de red es síncrona.

Gestor de colas individual con una configuración de publicación/suscripción

Una arquitectura alternativa que utiliza un único gestor de colas va a utilizar una configuración de publicación/suscripción. En la mensajería de publicación/suscripción, puede separar el proveedor de información de los clientes de esa información. Esto difiere de los estilos de mensajería de punto a punto en las arquitecturas descritas anteriormente, donde las aplicaciones deben tener información sobre la aplicación de destino, por ejemplo el nombre de la cola a la que transferir mensajes. Utilizando la publicación/suscripción de IBM MQ la aplicación emisora publica un mensaje con un tema especificado en función del asunto de la información. IBM MQ maneja la distribución del mensaje a aplicaciones que se mostrado un interés en ese asunto a través de una suscripción. Las aplicaciones receptoras tampoco necesitan tener información sobre el origen de los mensajes para recibirlos. Para obtener más información, consulte Mensajería de publicación/suscripción y Ejemplo de configuración de publicación/suscripción del gestor de colas único.

Conceptos relacionados

Introducción a IBM MQ

Tareas relacionadas

“Planificación de una arquitectura de IBM MQ” en la página 5
Cuando planifique su entorno de IBM MQ, tenga en cuenta el soporte que proporciona IBM MQ para las arquitecturas de uno o varios gestores de colas y para los estilos de mensajería de punto a punto y de publicación/suscripción. Además planifique los requisitos de recursos y su uso de los recursos de registro y copia de seguridad.

Creación y gestión de gestores de colas en Multiplatforms

Arquitecturas basadas en varios gestores de colas

Puede utilizar las técnicas de gestión de colas de mensajes distribuidos para crear una arquitectura de IBM MQ que implique la configuración y el uso de varios gestores de colas.

Antes de planificar su arquitectura de IBM MQ, debe familiarizarse con los conceptos básicos de IBM MQ. Consulte Visión general técnica de IBM MQ.

Una arquitectura de IBM MQ se puede modificar, sin alteraciones en las aplicaciones que proporcionan servicios, añadiendo gestores de colas adicionales.

Las aplicaciones pueden alojarse en la misma máquina que un gestor de colas, y luego establecer comunicación asíncrona con un servicio alojado en otro gestor de colas de otro sistema. Alternativamente, las aplicaciones que acceden a un servicio pueden conectarse como clientes a un gestor de colas que luego proporciona acceso asíncrono al servicio en otro gestor de colas.

Las rutas que conectan distintos gestores de colas y sus colas se definen utilizando las técnicas de gestión de colas distribuidas. Los gestores de colas dentro de la arquitectura se conectan mediante canales. Los canales se utilizan para mover mensajes automáticamente de un gestor de colas a otro en una dirección en función de la configuración de los gestores de colas.

Para obtener una visión general de alto nivel de la planificación de una red de IBM MQ, consulte "Diseño de redes de gestores de colas distribuidos" en la página 21.

Para obtener información sobre cómo planificar los canales para la arquitectura de IBM MQ, consulte Técnicas de gestión de colas distribuidas de IBM MQ.

La gestión de colas distribuidas permite crear y supervisar la comunicación entre gestores de colas. Para obtener más información sobre la gestión de colas distribuidas, consulte Introducción a la gestión de colas distribuidas.

Tareas relacionadas

"Planificación de una arquitectura de IBM MQ" en la página 5
Cuando planifique su entorno de IBM MQ, tenga en cuenta el soporte que proporciona IBM MQ para las arquitecturas de uno o varios gestores de colas y para los estilos de mensajería de punto a punto y de publicación/suscripción. Además planifique los requisitos de recursos y su uso de los recursos de registro y copia de seguridad.

Creación y gestión de gestores de colas en Multiplatforms

Planificación de sus gestores de colas y clústeres distribuidos

Puede conectar manualmente las colas alojadas en los gestores de colas distribuidos o puede crear un clúster de gestores de colas y dejar que el producto se conecte por sí solo a los gestores de colas. Para seleccionar una topología adecuada para su red de mensajería distribuida, debe tener en cuenta sus requisitos de control manual, tamaño de red, frecuencia de cambios, disponibilidad y escalabilidad.

Antes de empezar

Esta tarea presupone que comprende qué son las redes distribuidas y cómo funcionan. Para obtener una visión general técnica, consulte Colas y clústeres distribuidos.

Acerca de esta tarea

Para crear una red de mensajería distribuida, puede configurar manualmente los canales para que se conecten con las colas alojadas en diferentes gestores de colas o puede crear un clúster de gestores de colas. La agrupación en clúster permite que los gestores de colas se comuniquen entre sí sin necesidad de configurar definiciones de canales adicionales ni definiciones de colas remotas, lo cual simplifica su configuración y gestión.

Para elegir una topología adecuada para su red de publicación/suscripción distribuida, debe tener en cuenta las preguntas siguientes:

- ¿Cuánto control manual necesita sobre las conexiones de su red?
- ¿De qué tamaño será su red?
- ¿Cuál será el dinamismo de su sistema?
- ¿Cuáles son sus requisitos de disponibilidad y escalabilidad?

Procedimiento

- Considere cuánto control manual necesita sobre las conexiones de su red.

  Si solo necesita algunas conexiones o si es necesario definir con precisión conexiones individuales, probablemente deba crear manualmente la red.

  Si necesita varios gestores de colas que están relacionados lógicamente y necesitan compartir datos y aplicaciones, debe considerar agruparlos en un clúster de gestores de colas.
- Calcule de qué tamaño debe ser su red.
  a) Calcule cuántos gestores de colas necesita. Tenga en cuenta que las colas se pueden alojar en más de un gestor de colas.
  b) Si está considerando el uso de un clúster, añada dos gestores de colas adicionales para que actúen como repositorios completos.

  En el caso de las redes de gran tamaño, las tareas de configuración y mantenimiento de las conexiones pueden ocupar mucho tiempo y deberá considerar la posibilidad de utilizar un clúster.
- Considere si la actividad de red será muy dinámica o no.

  Planifique alojar las colas con mucha actividad en gestores de colas de alto rendimiento.

  Si espera que las colas se creen y supriman con frecuencia, considere la posibilidad de utilizar un clúster.
- Considere sus requisitos de disponibilidad y escalabilidad.
  a) Decida si es necesario garantizar la alta disponibilidad de los gestores de colas. Si es así, calcule a cuántos gestores de colas se aplica este requisito.
  b) Considere si algunos de los gestores de colas son menos capaces que otros.
  c) Considere si los enlaces de comunicaciones con algunos de los gestores de colas son más frágiles que otros.
  d) Considere la posibilidad de alojar las colas en varios gestores de colas.

  Las redes y clústeres configurados manualmente se pueden configurar para que sean altamente disponibles y escalables. Si utiliza un clúster, debe definir dos gestores de colas adicionales como repositorios completos. Si tiene dos repositorios completos se asegura de que el clúster continúe operativo en caso de que uno de los depósitos completo deje de estar disponible. Asegúrese de que los gestores de colas de depósito completo sean potentes y con alto rendimiento y tengan una buen conexión de red. NO planifique el uso de gestores de colas de depósito completo para cualquier otro trabajo.
- En función de estos cálculos, utilice los enlaces proporcionados como ayuda para decidir si configura manualmente las conexiones entre los gestores de colas o si utiliza un clúster.

Qué hacer a continuación

Ahora está preparado para configurar su red de publicación/suscripción distribuida.

Tareas relacionadas

Configuración de la gestión de colas distribuidas

Configuración de un clúster de gestores de colas

Diseño de redes de gestores de colas distribuidos

IBM MQ envía y recibe datos entre aplicaciones y a través de redes utilizando gestores de colas y canales. La planificación de redes supone la definición de requisitos para la creación de una infraestructura que permita conectar estos sistemas a través de una red.

Pueden crearse canales entre su sistema y cualquier otro sistema con el que necesite comunicarse. Pueden crearse canales de saltos múltiples para conectarse a sistemas con los que no tenga conexión directa. Las conexiones de canal de mensajes descritas en los escenarios se muestran como un diagrama de red en la Figura 1 en la página 22.

Si necesita crear canales entre sistemas en distintas redes físicas, o en canales que se comunican a través de un cortafuegos, el uso de IBM MQ Internet Pass-Thru podría simplificar la configuración. Para obtener más información, consulte IBM MQ Internet Pass-Thru.

Nombre de canal y de cola de transmisión

A las colas de transmisión se les puede dar cualquier nombre. Pero para evitar confusiones, puede darles los mismos nombres que los de los gestores de colas de destino o los de los alias de gestor de colas, según corresponda. Esto asocia la cola de transmisión con la ruta que utilizan, proporcionando una visión clara de las rutas paralelas creadas mediante los gestores de colas intermedios (saltos múltiples).

No está tan claro para los nombres de canal. Los nombres de canal de la Figura 1 en la página 22 para QM2, por ejemplo, deben ser diferentes para los canales de salida y para los de entrada. Todos los nombres de canal pueden contener los nombres de sus colas de transmisión, pero deben completarse para que sean exclusivos.

Por ejemplo, en QM2, existe un canal QM3 procedente de QM1 y un canal QM2 cuyo destino es QM3. Para que los nombres sean exclusivos, el primero puede denominarse QM3_desde_QM1 y el segundo QM3_desde_QM2. De este modo, los nombres de canal muestran el nombre de la cola de transmisión en la primera parte del nombre. El sentido y el nombre del gestor de colas adyacente se muestran en la segunda parte del nombre.

En la Tabla 1 en la página 22 se ofrece una tabla de nombres de canal propuestos para la Figura 1 en la página 22.

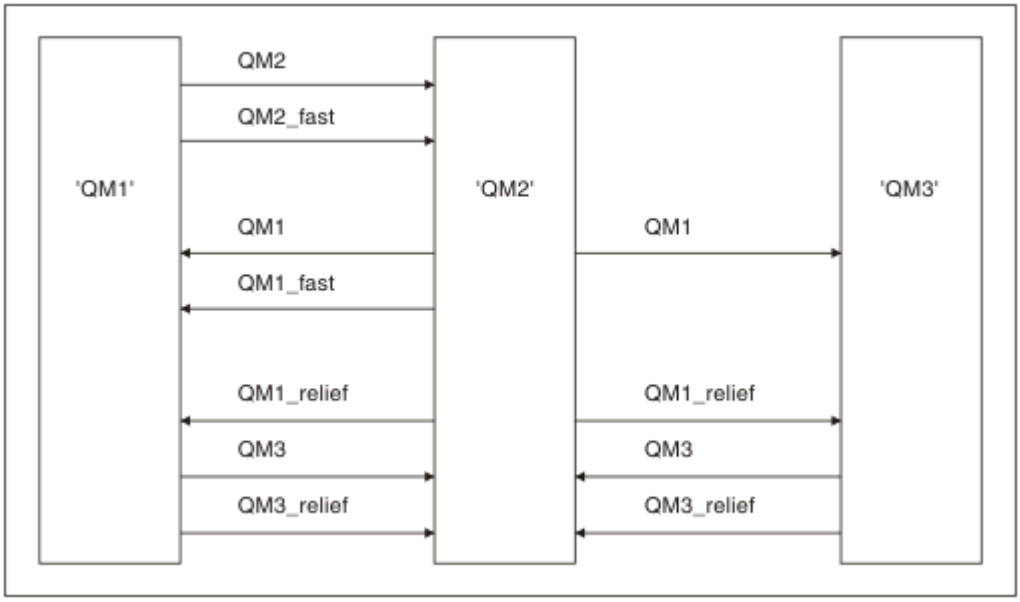


Figura 1. Diagrama de red que muestra todos los canales

Tabla 1. Ejemplo de nombres de canal

Nombre de la ruta	Canal que aloja los gestores de colas	Nombre de cola de transmisión	Nombre propuesto para el canal
QM1	QM1 & QM2	QM1 (en QM2)	QM1.from.QM2
QM1	QM2 & QM3	QM1 (en QM3)	QM1.from.QM3
QM1_fast	QM1 & QM2	QM1_rápido (en QM2)	QM1_fast.from.QM2
QM1_relief	QM1 & QM2	QM1_auxiliar (en QM2)	QM1_relief.from.QM2

Tabla 1. Ejemplo de nombres de canal (continuación)			
Nombre de la ruta	**Canal que aloja los gestores de colas**	**Nombre de cola de transmisión**	**Nombre propuesto para el canal**
QM1_relief	QM2 & QM3	QM1_auxiliar (en QM3)	QM1_relief.from.QM3
QM2	QM1 & QM2	QM2 (en QM1)	QM2.from.QM1
QM2_fast	QM1 & QM2	QM2_rápido (en QM1)	QM2_fast.from.QM1
QM3	QM1 & QM2	QM3 (en QM1)	QM3.from.QM1
QM3	QM2 & QM3	QM3 (en QM2)	QM3.from.QM2
QM3_relief	QM1 & QM2	QM3_auxiliar (en QM1)	QM3_relief.from.QM1
QM3_relief	QM2 & QM3	QM3_auxiliar (en QM2)	QM3_relief.from.QM2

Nota:

1. z/OS En IBM MQ for z/OS, los nombres del gestor de colas están limitados a cuatro caracteres.
2. Denomine todos los canales de la red de forma exclusiva. Como se muestra en la Tabla 1 en la página 22, una buena manera de hacerlo es incluyendo los nombres de los gestores de colas de origen y de destino en el nombre del canal.

Planificador de la red

La creación de una red presupone que existe otra función de nivel superior del *planificador de la red* cuyos planes los implementan otros miembros del equipo.

Para las aplicaciones que más se utilizan, es más económico pensar en términos de sitios de acceso local para la concentración del tráfico de mensajes, utilizando enlaces de banda ancha entre los sitios de acceso local, como se muestra en la Figura 2 en la página 24.

En este ejemplo hay dos sistemas principales y varios sistemas satélite. La configuración real dependería de consideraciones empresariales. Hay dos gestores de colas concentradores ubicados en centros adecuados. Cada QM concentrador tiene canales de mensajes con los gestores de colas locales:

- El QM concentrador 1 tiene canales de mensajes con cada uno de los tres gestores de colas locales, QM1, QM2 y QM3. Las aplicaciones que utilizan estos gestores de colas pueden comunicarse entre ellas mediante los QM concentradores.
- El QM concentrador 2 tiene canales de mensajes con cada uno de los tres gestores de colas locales, QM4, QM5 y QM6. Las aplicaciones que utilizan estos gestores de colas pueden comunicarse entre ellas mediante los QM concentradores.
- Los QM concentradores tienen canales de mensajes entre ellos que permiten que cualquier aplicación en un gestor de colas pueda intercambiar mensajes con cualquier otra aplicación en otro gestor de colas.

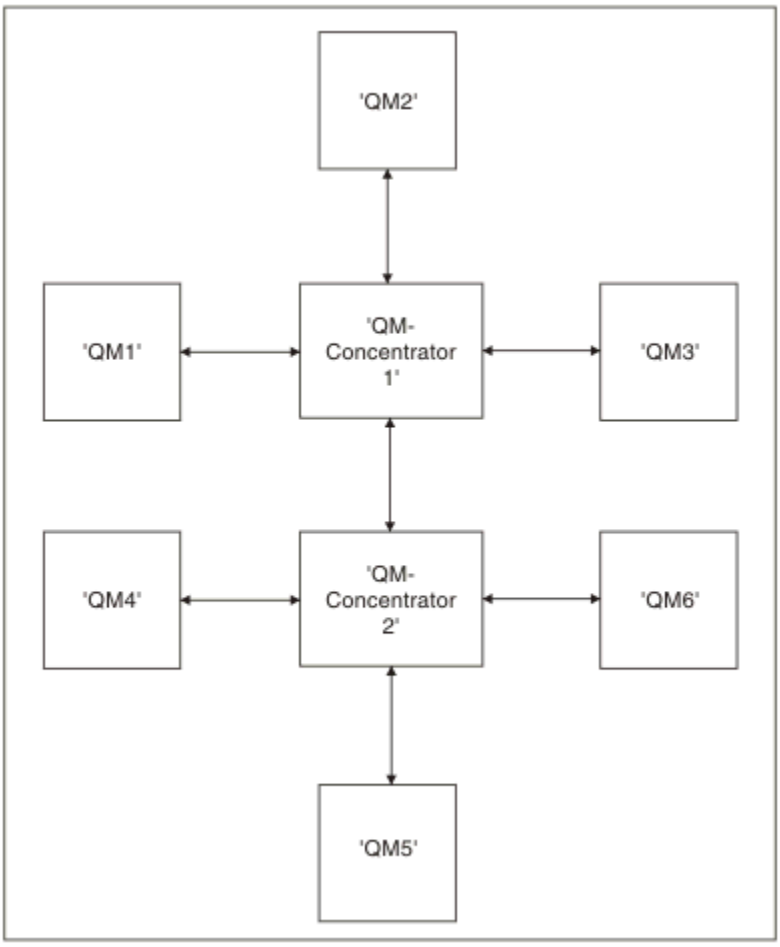


Figura 2. Diagrama de red que muestra los QM concentradores

Diseño de clústeres

Los clústeres proporcionan un mecanismo para interconectar gestores de colas de forma que simplifica la configuración inicial y la gestión continua. Los clústeres se deben diseñar con mucho cuidado para asegurarse de que funcionan correctamente y de que pueden alcanzar los niveles necesarios de disponibilidad y capacidad de respuesta.

Antes de empezar

Para obtener una introducción a los conceptos de clústeres, consulte los temas siguientes:

- Colas y clústeres distribuidos
- “Comparación de agrupación en clúster y gestión de colas distribuidas” en la página 30
- Componentes de un clúster

Cuando diseñe el clúster del gestor de colas, tendrá que tomar algunas decisiones. En primer lugar, debe decidir qué gestores de colas del clúster van a contener los repositorios completos de información de clúster. Cualquier gestor de colas que cree puede trabajar en un clúster. Puede seleccionar cualquier número de gestores de colas para esta finalidad, pero el número ideal es dos. Si desea más información sobre cómo seleccionar gestores de colas para contener los repositorios completos, consulte “Selección de gestores de colas para que contengan repositorios completos” en la página 33.

Consulte los temas siguientes si desea más información sobre cómo diseñar el clúster:

- “Clústeres de ejemplo” en la página 39
- “Organización de un clúster” en la página 34

- “Convenios de denominación de clústeres” en la página 34
- z/OS “Queue sharing groups and clusters” en la página 36
- “Solapamiento de clústeres” en la página 36

Qué hacer a continuación

Consulte los temas siguientes si desea más información sobre la configuración y el trabajo con clústeres:

- Establecimiento de la comunicación en un clúster
- Configuración de un clúster de gestores de colas
- Direccionamiento de mensajes a y desde clústeres
- Utilización de clústeres para la gestión de carga de trabajo

Si desea más información para ayudarle a configurar el clúster, consulte “Consejos para la agrupación en clúster” en la página 37.

Planificación de cómo utilizar varias colas de transmisión de clúster

Puede definir de forma explícita las colas de transmisión o hacer que el sistema genere las colas de transmisión. Si define por su cuenta las colas de transmisión, tendrá más control sobre las definiciones de cola. z/OS En z/OS, también tiene más control sobre el conjunto de páginas donde se conservan los mensajes.

Definición de las colas de transmisión

Existen dos métodos para definir las colas de transmisión:

- Automáticamente, utilizando el atributo de gestor de colas DEFCLXQ, como se indica a continuación:

  ```
  ALTER QMGR DEFCLXQ(SCTQ | CHANNEL)
  ```

  DEFCLXQ(SCTQ) indica que la cola de transmisión predeterminada para todos los canales de clúster emisor es SYSTEM.CLUSTER.TRANSMIT.QUEUE. Éste es el valor predeterminado.

  DEFCLXQ(CHANNEL) indica que, de forma predeterminada, cada canal de clúster emisor utiliza una cola de transmisión independiente denominada SYSTEM.CLUSTER.TRANSMIT.*nombre canal*. Cada cola de transmisión la define automáticamente por el gestor de colas. Consulte “Colas de transmisión del clúster definidas automáticamente” en la página 26 para obtener más información.
- Manualmente, definiendo una cola de transmisión con un valor especificado para el atributo CLCHNAME. El atributo CLCHNAME indica que los canales de clúster emisor deben utilizar la cola de transmisión. z/OS Si está definiendo manualmente una cola de transmisión en z/OS, consulte “Planificación de las colas de transmisión de clúster definidas manualmente” en la página 28 para obtener más información.

¿Qué seguridad necesito?

Para iniciar un conmutador, ya sea de forma automática o manual, necesita autorización para iniciar un canal.

Para definir la cola que se utiliza como una cola de transmisión, necesita autorización de IBM MQ estándar.

¿Cuál es el momento adecuado para implementar el cambio?

Al cambiar la cola de transmisión utilizada por los canales de clúster emisor, debe asignar un tiempo en el que realizar la actualización, teniendo en cuenta los siguientes puntos:

- El tiempo necesario para que un canal conmute la cola de transmisión depende del número total de mensajes en la cola de transmisión antigua, del número de mensajes que necesitan moverse y del tamaño de los mensajes.
- Las aplicaciones pueden continuar colocando mensajes en la cola de transmisión mientras se está produciendo el cambio. Esto puede llevar a un aumento en la hora de transición.
- Puede cambiar el parámetro CLCHNAME de cualquier cola de transmisión o DEFCLXQ en cualquier momento, preferiblemente cuando la carga de trabajo sea baja.

  Tenga en cuenta que nada sucede de forma inmediata.
- Los cambios solo se producen cuando el canal se inicia o reinicia. Cuando se inicia un canal, comprueba la configuración actual y conmuta a una nueva cola de transmisión si es necesario.
- Existen varios cambios que pueden alterar la asociación de un canal de clúster emisor con una cola de transmisión:
  - Modificar el valor del atributo CLCHNAME de una cola de transmisión, haciendo que CLCHNAME sea menos específico o esté en blanco.
  - Modificar el valor del atributo CLCHNAME de una cola de transmisión, haciendo que CLCHNAME sea más específico.
  - Suprimir una cola con CLCHNAME especificado.
  - Modificar el atributo de gestor de colas DEFCLXQ.

¿Cuánto tiempo tardará la conmutación?

Durante el período de transición, cualquier mensaje para el canal se moverá de una cola de transmisión a otra. El tiempo necesario para que un canal conmute la cola de transmisión depende del número total de mensajes en la cola de transmisión antigua y de cuántos mensajes es necesario mover.

En el caso de las colas que contienen miles de mensajes, deberá tardar menos de un segundo en mover los mensajes. El tiempo real depende del número y el tamaño de los mensajes. El gestor de colas debe poder mover los mensajes a muchos megabytes por segundo.

Las aplicaciones pueden continuar colocando mensajes en la cola de transmisión mientras se está produciendo el cambio. Esto puede llevar a un aumento en la hora de transición.

Cada canal de clúster emisor afectado debe reiniciarse para que el cambio entre en vigor. Por lo tanto, es mejor cambiar la configuración de cola de transmisión cuando el gestor de colas no está ocupado y hay pocos mensajes almacenados en las colas de transmisión de clúster.

El Mandato **runswchl** z/OS o Mandato SWITCH CHANNEL (*) STATUS en CSQUTIL en z/OS se puede utilizar para consultar el estado de los canales de clúster emisor y qué cambios pendientes están pendientes en su configuración de cola de transmisión.

Cómo implementar el cambio

Consulte Implementación del sistema utilizando varias colas de transmisión de clúster para obtener más detalles sobre cómo realizar el cambio en varias colas de transmisión de clúster, ya sea forma manual o automática.

Deshacer el cambio

z/OS

Consulte Deshacer un cambio en una cola de transmisión en z/OS para obtener detalles sobre cómo restituir los cambios si encuentra problemas.

Colas de transmisión del clúster definidas automáticamente
Puede hacer que el sistema genere las colas de transmisión.

Antes de empezar

z/OS Para configurar las colas de transmisión de clúster manualmente en z/OS, consulte “Planificación de las colas de transmisión de clúster definidas manualmente” en la página 28.

Acerca de esta tarea

Si un canal no tiene asociada una cola de transmisión de clúster definida manualmente y especifica DEFCLXQ(CHANNEL), cuando se inicia el canal, automáticamente el gestor de colas define una cola dinámica permanente para el canal emisor de clúster. La cola modelo SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE se utiliza para definir automáticamente la cola de transmisión de clúster dinámico permanente con el nombre SYSTEM.CLUSTER.TRANSMIST.ChannelName.

Importante: z/OS En IBM MQ 8.0, el gestor de colas no tiene SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE. No puede migrar directamente de IBM MQ 8.0 a esta versión. Para obtener información sobre cómo añadir SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE a un gestor de colas que se migra desde IBM MQ 8.0, consulte este tema en la documentación de la versión provisional que ha utilizado para migrar el gestor de colas.

Procedimiento

1. Utilice el atributo del gestor de colas *DEFCLXQ*.

   Si desea más información sobre este atributo, consulte ALTER QMGR.

   Hay dos opciones:

   SCTQ
   Esta opción es el valor predeterminado y significa que utiliza la cola individual SYSTEM.CLUSTER.TRANSMIT.QUEUE.

   CHANNEL
   Significa que utiliza varias colas de transmisión del clúster.
2. Para conmutar a la nueva asociación:
   - Detiene y reinicia el canal.
   - El canal utiliza la nueva definición de la cola de transmisión.
   - Los mensajes se transfieren mediante un proceso de conmutación transicional desde la cola antigua a la nueva cola de transmisión.

   Tenga en cuenta que cualquier mensaje de aplicación se coloca en la definición antigua.

   Cuando el número de mensajes de la cola antigua sea cero, los nuevos mensajes se colocan directamente en la nueva cola de transmisión.
3. Para supervisar cuando finaliza el proceso de conmutación:
   a) Una conmutación de la cola de transmisión que ha sido iniciada por un canal se ejecuta como programa de fondo y el administrador puede supervisar el registro de trabajos del gestor de colas para determinar si se ha completado.
   b) Supervise los mensajes en el registro de trabajos para mostrar el progreso de la conmutación.
   c) Para asegurarse de que sólo los canales que desea están utilizando esta cola de transmisión, emita el mandato DIS CLUSQMGR (*) donde, por ejemplo, la propiedad de cola de transmisión que define la cola de transmisión es APPQMGR.CLUSTER1.XMITQ.
   d) z/OS
   Utilice el mandato SWITCH CHANNEL (*) STATUS bajo CSQUTIL.

   Esta opción le indica qué cambios están pendientes y el número de mensajes que se han de mover entre las colas de transmisión.

Resultados

Ha configura su cola o colas de transmisión del clúster.

Tareas relacionadas

“Planificación de las colas de transmisión de clúster definidas manualmente” en la página 28
En IBM MQ for z/OS, si define usted mismo las colas de transmisión, tiene más control sobre las definiciones y el conjunto de páginas en el que se retienen los mensajes.

Referencia relacionada

ALTER QMGR

DISPLAY CLUSQMGR

z/OS *Planificación de las colas de transmisión de clúster definidas manualmente*

En IBM MQ for z/OS, si define usted mismo las colas de transmisión, tiene más control sobre las definiciones y el conjunto de páginas en el que se retienen los mensajes.

Antes de empezar

Para configurar automáticamente las colas de transmisión del clúster, consulte “Colas de transmisión del clúster definidas automáticamente” en la página 26.

Acerca de esta tarea

El administrador define manualmente una cola de transmisión y utiliza el atributo de cola CLCHNAME para definir qué canal emisor de clúster, o canales, utilizarán esta cola como cola de transmisión.

Tenga en cuenta que CLCHNAME puede incluir un comodín al principio o al final para permitir que se utilice una sola cola en varios canales.

Procedimiento

1. Por ejemplo, escriba lo siguiente:

```
DEFINE QLOCAL(APPQMGR.CLUSTER1.XMITQ)
CLCHNAME(CLUSTER1.TO.APPQMGR)
USAGE(XMITQ) STGCLASS(STG1)
INDXTYPE( CORRELID ) SHARE

DEFINE STGCLASS(STG1) PSID(3)
DEFINE PSID(3) BUFFERPOOL(4)
```

   Consejo: Debe planificar qué conjunto de páginas (y agrupación de almacenamiento intermedio) utilizará para sus colas de transmisión. Puede tener conjuntos de páginas diferentes para colas diferentes y proporcionar aislamiento entre ellas, de modo que un conjunto de páginas que se llene no afecte a las colas de transmisión de otros conjuntos de páginas.

   Consulte la sección Trabajo con las colas de transmisión del clúster y los canales de clúster emisor para obtener información sobre cómo cada canal selecciona la cola adecuada.

   Cuando el canal se inicia, cambia su asociación a la nueva cola de transmisión. Para asegurarse de que no se pierde ningún mensaje, el gestor de colas transfiere automáticamente los mensajes de la cola de transmisión de clúster antigua a la cola de transmisión nueva en orden.

2. Utilice la función CSQUTIL SWITCH para cambiar a la nueva asociación.

   Para obtener más información, consulte Conmutar la cola de transmisión asociada a los canales de clúster emisor (SWITCH).

   a) DETENGA el o los canales, cuya cola de transmisión se haya de modificar para que su estado sea STOPPED.

      Por ejemplo:

```
STOP CHANNEL(CLUSTER1.TO.APPQMGR)
```

b) Cambie el atributo `CLCHNAME(XXXX)` de la cola de transmisión.

c) Utilice la función SWITCH para conmutar los mensajes o supervisar lo que está sucediendo.
Utilice el mandato

```
SWITCH CHANNEL(*) MOVEMSGS(YES)
```

para mover los mensajes sin iniciar el canal.

d) Inicie el o los canales y compruebe si el canal está utilizando las colas correctas.
Por ejemplo:

```
DIS CHS(CLUSTER1.TO.APPQMGR)
DIS CHS(*) where(XMITQ eq APPQMGR.CLUSTER1.XMITQ)
```

Consejo: El proceso siguiente utiliza la función CSQUTIL SWITCH. Para obtener más información, consulte Conmutar la cola de transmisión asociada con los canales de clúster emisor (SWITCH).

No es necesario que utilice esta función pero si la utiliza tendrá más opciones:

- El uso de SWITCH CHANNEL (*) STATUS proporciona un modo fácil de identificar el estado de conmutación de los canales de clúster emisor. Esto permite al administrador ver qué canales se están conmutando actualmente y qué canales están pendientes de que se haga efectiva una conmutación cuando se inicien a continuación dichos canales.

  Sin esta función, el administrador debe utilizar los mandatos DISPLAY y, a continuación, procesar la salida resultante para confirmar esta información. El administrador también puede confirmar que un cambio de configuración ha tenido el resultado necesario.
- Si se utiliza CSQUTIL para iniciar la conmutación, CSQUTIL continúa supervisando el progreso de esta operación y solo finaliza cuando se ha completado la conmutación.

  Esto facilita mucho la ejecución de estas operaciones en procesos por lotes. Asimismo, si se ejecuta CSQUTIL para conmutar varios canales, CSQUTIL realiza estas acciones de forma secuencial. Esto puede tener menos impacto en su empresa que ejecutar en paralelo varias conmutaciones.

Resultados

Ha configurado la cola o colas de transmisión de clúster en z/OS.

Control de accesos y varias colas de transmisión de clúster

Elija entre tres modalidades de comprobación cuando una aplicación transfiere mensajes a las colas de clúster remoto. Las modalidades se están comprobando de forma remota en la cola de clúster, se están comprobando localmente en `SYSTEM.CLUSTER.TRANSMIT.QUEUE`, o se están comprobando los perfiles locales para la cola de clúster o el gestor de colas de clúster.

IBM MQ le ofrece la opción de comprobar localmente, o local y remotamente, si un usuario tiene permiso para transferir un mensaje a una cola remota. Una aplicación IBM MQ típica utiliza sólo la comprobación local y confía en el gestor de colas remoto, confiando en las comprobaciones de acceso realizadas en el gestor de colas local. Si no se utiliza la comprobación remota, el mensaje se transfiere a la cola de destino con la autoridad del proceso de canal de mensajes remoto. Para utilizar la comprobación remota debe establecer la autorización de transferencia del canal receptor en la seguridad de contexto.

Las comprobaciones locales se realizan en la cola que la aplicación abre. En las colas distribuidas, la aplicación normalmente abre una definición de cola remota y las comprobaciones de acceso se realizan sobre la definición de cola remota. Si el mensaje se transfiere con una cabecera de direccionamiento completa, las comprobaciones se realizan en la cola de transmisión. Si una aplicación abre una cola de clúster que no está en el gestor de colas local, no hay ningún objeto local que comprobar. Las comprobaciones de control de acceso se realizan en la cola de transmisión

de clúster, SYSTEM.CLUSTER.TRANSMIT.QUEUE. Incluso con varias colas de transmisión de clúster, las comprobaciones de control de acceso local para las colas de clúster remoto se realizan en SYSTEM.CLUSTER.TRANSMIT.QUEUE.

La elección entre la comprobación local o remota es una elección entre dos extremos. La comprobación remota es precisa. Cada usuario debe tener un perfil de control de acceso en cada gestor de colas del clúster para transferirlo a cualquier cola de clúster. La comprobación local es general. Cada usuario necesita sólo un perfil de control de acceso para la cola de transmisión de clúster en el gestor de colas al que está conectado. Con este perfil, pueden transferir un mensaje a cualquier cola de clúster en cualquier gestor de colas de cualquier clúster.

Los administradores tienen otra forma de configurar el control de acceso para las colas de clúster. Puede crear un perfil de seguridad para una cola de clúster en cualquier gestor de colas del clúster mediante el mandato **setmqaut**. El perfil entra en vigor si abre una cola de clúster remoto localmente, especificando únicamente el nombre de cola. También puede configurar un perfil para un gestor de colas remoto. Si lo hace, el gestor de colas puede comprobar el perfil de un usuario que abre una cola de clúster proporcionando un nombre completo.

Los nuevos perfiles sólo funcionan si cambia la stanza del gestor de colas, **ClusterQueueAccessControl** a RQMName. El valor predeterminado es Xmitq. Debe crear perfiles para todas las aplicaciones existentes de colas de clúster que utilizan colas de clúster. Si cambia la stanza por RQMName sin crear perfiles es posible que las aplicaciones fallen.

Consejo: La comprobación de acceso a la cola de clúster no se aplica a la cola remota. Las comprobaciones de accesos se siguen realizando sobre las definiciones locales. Los cambios significan que puede seguir el mismo procedimiento para configurar la comprobación de accesos en colas de clúster y temas de clúster. z/OS Los cambios también alinean el procedimiento de comprobación de accesos para las colas de clúster más estrechamente con z/OS. Los mandatos para configurar la comprobación de accesos en z/OS son diferentes, pero ambos comprueban el acceso sobre un perfil en lugar de hacerlo sobre el propio objeto.

Conceptos relacionados

“Agrupación en clúster: Aislamiento de aplicaciones utilizando varias colas de transmisión de clúster” en la página 49
Puede aislar los flujos de mensajes entre los gestores de colas de un clúster. Puede colocar mensajes transportados por diferentes canales de clúster emisor en diferentes colas de transmisión de clúster. Puede utilizar el enfoque en un solo clúster o con clústeres solapados. El tema proporciona ejemplos y algunas prácticas recomendadas que le guiarán para elegir un procedimiento para utilizarlo.

Tareas relacionadas

Establecer ClusterQueueAccessControl

Comparación de agrupación en clúster y gestión de colas distribuidas

Compare los componentes que deben definirse para conectar gestores de colas utilizando la gestión de colas distribuidas y la agrupación en clúster

Si no utiliza clústeres, los gestores de colas son independientes y se comunican mediante la gestión de colas distribuidas. Si un gestor de colas necesita enviar mensajes a otro, se debe definir:

- Una cola de transmisión
- Un canal para el gestor de colas remoto

La Figura 3 en la página 31 muestra los componentes necesarios para la gestión de colas distribuidas.

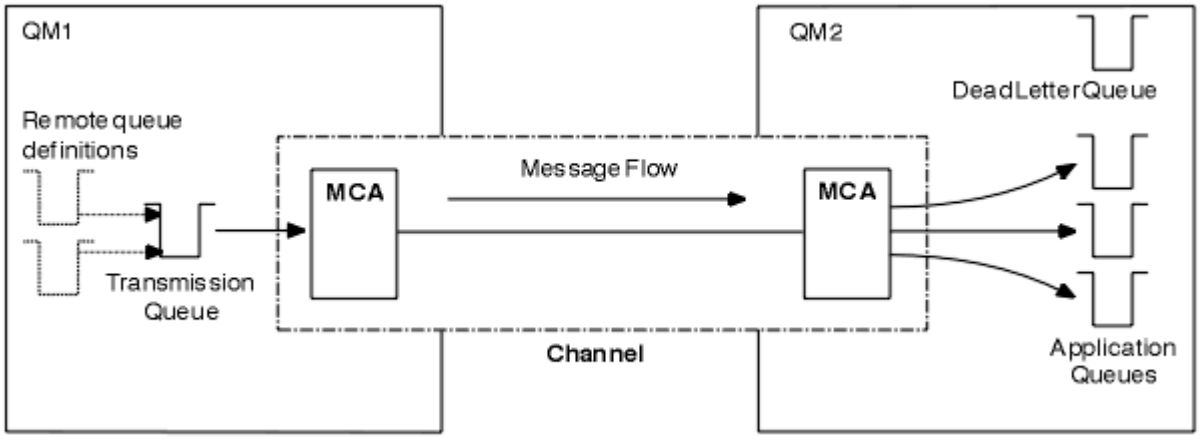


Figura 3. Gestión de colas distribuidas

Si agrupa los gestores de colas en un clúster, las colas de cualquier gestor de colas están disponibles para cualquier otro gestor de colas del clúster. Cualquier gestor de colas puede enviar un mensaje a cualquier otro gestor de colas en el mismo clúster sin definiciones explícitas. No se proporcionan definiciones de canal, definiciones de cola remota o colas de transmisión para cada destino. Cada gestor de colas de un clúster tiene una sola cola de transmisión desde la que puede transmitir mensajes a cualquier otro gestor de colas del clúster. Cada gestor de colas de un clúster tiene que definir sólo:

- Un canal de clúster receptor en el que se recibirán los mensajes
- Un canal de clúster emisor con el que se presenta y se informa sobre el clúster

Definiciones para configurar un clúster en comparación con la gestión de colas distribuidas

Observe la Figura 4 en la página 31, que muestra cuatro gestores de colas, cada uno de ellos con dos colas. Considere cuántas definiciones son necesarias para conectar estos gestores de colas utilizando la gestión de colas distribuidas. Compare cuántas definiciones son necesarias para configurar la misma red como un clúster.

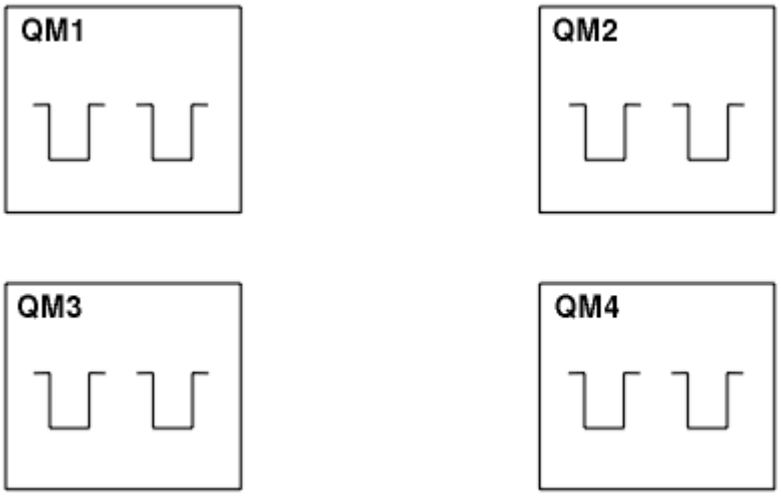


Figura 4. Una red de cuatro gestores de colas

Definiciones para configurar una red utilizando la gestión de colas distribuidas

Para configurar la red que se muestra en la Figura 3 en la página 31 utilizando la gestión de colas distribuidas, puede tener las siguientes definiciones:

Tabla 2. Definiciones para gestión de colas distribuidas		
Descripción	**Número por gestor de colas**	**Número total**
Una definición de canal emisor para un canal en el que enviar mensajes a cualquier otro gestor de colas	3	12
Una definición de canal receptor para un canal en el que recibir mensajes de cualquier otro gestor de colas	3	12
Una definición de cola de transmisión para una cola de transmisión a cualquier otro gestor de colas	3	12
Una definición de cola local para cada cola local	2	8
Una definición de cola remota para cada cola remota a la que este gestor de colas desea transferir mensajes	6	24

Puede reducir este número de definiciones utilizando definiciones de canal receptor genéricas. El número máximo de definiciones podría ser de hasta 17 en cada gestor de colas, que es un total de 68 para esta red.

Definiciones para configurar una red utilizando clústeres

Para configurar la red que se muestra en la Figura 3 en la página 31 utilizando clústeres necesita las definiciones siguientes:

Tabla 3. Definiciones para agrupación en clúster		
Descripción	**Número por gestor de colas**	**Número total**
Una definición de canal de clúster emisor para un canal en el que enviar mensajes a un gestor de colas de repositorio	1	4
Una definición de canal de clúster receptor para un canal en el que recibir mensajes de otros gestores de colas en el clúster	1	4
Una definición de cola local para cada cola local	2	8

Para configurar este clúster de gestores de colas (con dos repositorios completos), se necesitan cuatro definiciones en cada gestor de colas, un total de dieciséis definiciones en conjunto. También es necesario modificar las definiciones del gestor de colas para dos de los gestores de colas, para convertirlos en gestores de colas de repositorio completo para el clúster.

Sólo se necesita una definición de canal CLUSSDR y CLUSRCVR. Cuando el clúster está definido, puede añadir o eliminar gestores de colas (excepto los gestores de colas de repositorio) sin ninguna interrupción en los otros gestores de colas.

La utilización de un clúster reduce el número de definiciones necesarias para configurar una red que contenga muchos gestores de colas.

Con menos definiciones que hacer hay menos riesgo de error:

- Los nombres de objeto siempre coinciden, por ejemplo, el nombre de canal en un par emisor-receptor.
- El nombre de cola de transmisión especificado en una definición de canal siempre coincide con la definición de cola de transmisión correcta o el nombre de cola de transmisión especificado en una definición de cola remota.
- Una definición QREMOTE siempre apunta a la cola correcta en el gestor de colas remoto.

Una vez que se ha configurado un clúster, puede mover colas de clúster de un gestor de colas a otro dentro del clúster sin tener que realizar ningún trabajo de gestión del sistema en cualquier otro gestor

de colas. No hay ninguna posibilidad de olvidarse de suprimir o modificar definiciones de canal, de cola-remota o de cola de transmisión. Puede añadir nuevos gestores de colas a un clúster sin ninguna interrupción en la red existente.

Selección de gestores de colas para que contengan repositorios completos

En cada clúster debe seleccionar al menos uno, y preferiblemente dos gestores de colas para que contengan repositorios completos. Dos repositorios completos son suficientes para todas las circunstancias excepto las más excepcionales. Si es posible, elija gestores de colas que se alojen en plataformas robustas y con conexión permanente, que no tengan interrupciones coincidentes y que estén en una posición central geográficamente hablando. Considere también el uso de sistemas dedicados como hosts de repositorios completos y no utilice estos sistemas para otras tareas.

Los *repositorios completos* son gestores de colas que contienen una imagen completa del estado del clúster. Para compartir esta información, cada repositorio completo está conectado mediante los canales CLUSSDR (y sus correspondientes definiciones CLUSRCVR) a cada dos repositorios completos en el clúster. Debe definir manualmente estos canales.

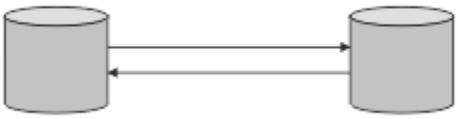


Figura 5. Dos repositorios completos conectados.

Cada dos gestores de colas del clúster conserva una imagen de lo que se sabe sobre el estado del clúster en un *repositorio parcial*. Estos gestores de colas publican información sobre sí mismos, y solicitan información sobre otros gestores de colas, utilizando cualquiera de los dos repositorios completos disponibles. Si el depósito completo elegido no está disponible, se utiliza otro. Cuando el repositorio completo elegido vuelve a estar disponible, recopila la información nueva y cambiada más reciente de los otros para que se mantengan sincronizados. Si todos los repositorios completos se quedan fuera de servicio, los otros gestores de colas utilizan la información que tienen en sus repositorios parciales. Sin embargo, están limitados a utilizar la información que tienen; la información nueva y las solicitudes de actualizaciones no se pueden procesar. Cuando los repositorios completos vuelven a conectarse a la red, se intercambian mensajes para actualizar todos los repositorios (tanto los completos como los parciales).

Al planificar la asignación de los repositorios completos, tenga en cuenta las siguientes consideraciones:

- Los gestores de colas elegidos para contener repositorios completos tienen que ser fiables y gestionados. Elija gestores de colas que estén alojados en una plataforma robusta y permanentemente conectados.
- Tenga en cuenta las interrupciones planificadas de los sistemas que alojan los repositorios completos y asegúrese de que no tengan interrupciones coincidentes.
- Tenga en cuenta el rendimiento de la red: elija gestores de colas que estén en una posición central geográficamente, o que compartan el mismo sistema que otros gestores de colas del clúster.
- Tenga en cuenta si un gestor de colas es miembro de más de un clúster. Puede ser administrativamente conveniente utilizar el mismo gestor de colas para alojar los repositorios completos para varios clústeres, siempre que esta ventaja esté equilibrada con el grado de ocupación que espera que tenga el gestor de colas.
- Puede dedicar algunos sistemas de modo que contengan sólo repositorios completos, pero no debe utilizar estos sistemas para realizar otras tareas. De este modo, estos sistemas sólo requieren mantenimiento para la configuración del gestor de colas, y no se retiran de servicio durante el mantenimiento de otras aplicaciones de negocio. También garantiza que la tarea de mantener el repositorio no compita con las aplicaciones de los recursos del sistema. Esto puede resultar especialmente beneficioso en clústeres de gran tamaño (es decir, los clústeres con más de un millar de gestores de colas), donde los repositorios completos tienen una carga de trabajo mucho mayor a la hora de mantener el estado del clúster.

Tener más de dos repositorios completos es posible, pero no se recomienda, salvo en circunstancias especiales. Aunque las definiciones de objeto (es decir, colas, temas y canales) fluyen a todos los repositorios completos disponibles, las peticiones sólo fluyen desde un repositorio parcial a un máximo de dos repositorios completos. Esto significa que, cuando se han definido más de dos repositorios completos, y cualquiera de los dos repositorios completos quedan no disponibles, puede que algunos repositorios parciales no reciban las actualizaciones que esperarían. Consulte Clústeres MQ: ¿Por qué sólo dos repositorios completos?

Una situación en la que podría resultar de utilidad definir más de dos repositorios completos es cuando se migran repositorios completos existentes a un hardware nuevo o a gestores de colas nuevos. En un caso así, podría realizar una sustitución de repositorios completos, pero debe confirmar que se hayan completado del todo antes de retirar los repositorios completos anteriores. Siempre que añada un repositorio completo, recuerde que debe conectarlo directamente a cada dos repositorios completos con los canales CLUSSDR.

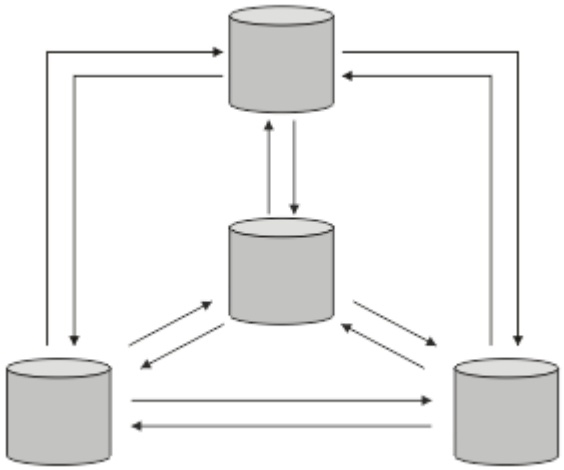


Figura 6. Más de dos repositorios completos conectados

Información relacionada

Clústeres MQ: ¿Por qué sólo dos repositorios completos?

¿Qué tamaño puede tener un clúster MQ?

Organización de un clúster

Seleccione qué gestores de colas desea enlazar a qué repositorio completo. Tenga en cuenta el efecto en el rendimiento, la versión del gestor de colas y si varios canales CLUSSDR son convenientes.

Una vez que ha seleccionado los gestores de colas que van a contener repositorios completos, debe decidir qué gestores de colas desea enlazar a qué repositorio completo. La definición de canal CLUSSDR enlaza un gestor de colas a un repositorio completo en el que se informa sobre los otros repositorios completos del clúster. A partir de entonces, el gestor de colas envía mensajes a cualquiera de los dos repositorios completos. Siempre intenta utilizar primero aquel para el que tiene una definición de canal CLUSSDR. Puede decidir enlazar un gestor de colas a cualquiera de los dos repositorios completos. En la elección, tenga en cuenta la topología de la configuración, y la ubicación física o geográfica de los gestores de colas.

Puesto que toda la información del clúster se envía a dos repositorios completos, puede haber situaciones en las que desee hacer una segunda definición de canal CLUSSDR. Puede definir un segundo canal CLUSSDR en un clúster que tenga muchos repositorios completos repartidos en un área amplia. Podrá entonces controlar a qué dos repositorios completos se envía la información.

Convenios de denominación de clústeres

Considere la posibilidad de denominar gestores de colas en el mismo clúster utilizando un convenio de denominación que identifique el clúster al que pertenece el gestor de colas. Utilice un convenio de denominación similar para los nombres de canal y amplíelo para describir las características del canal.

Procedimientos recomendados al denominar clústeres de MQ

Aunque los nombres de clúster pueden tener hasta 48 caracteres, los nombres de clúster relativamente cortos son útiles al aplicar convenios de denominación a otros objetos. Consulte “Prácticas recomendadas al elegir nombres de canal de clúster” en la página 35.

Al elegir un nombre de clúster, normalmente es útil representar el 'propósito' del clúster (que es probable que sea de larga duración) en lugar del 'contenido'. Por ejemplo, 'B2BPROD' o 'ACTTEST' en lugar de 'QM1_QM2_QM3_CLUS'.

Prácticas recomendadas al elegir nombres de gestor de colas de clúster

Si está creando un nuevo clúster y sus miembros desde cero, tenga en cuenta un convenio de denominación para los gestores de colas que refleje su uso del clúster. Cada gestor de colas debe tener un nombre diferente. Sin embargo, puede proporcionar a los gestores de colas de un clúster un conjunto de nombres similares, para ayudarle a identificar y recordar agrupaciones lógicas (por ejemplo, 'ACTTQM1, ACTTQM2).

Los nombres de gestor de colas relativamente cortos (por ejemplo, menos de 8 caracteres) ayudan si elige utilizar el convenio descrito en la sección siguiente, o algo similar, para los nombres de canal.

Prácticas recomendadas al elegir nombres de canal de clúster

Puesto que los gestores de colas y los clústeres pueden tener nombres de hasta 48 caracteres, y un nombre de canal está limitado a 20 caracteres, tenga cuidado al nombrar por primera vez los objetos para evitar tener que cambiar el convenio de denominación a mitad de un proyecto (consulte la sección anterior).

Al definir canales, recuerde que los canales de clúster emisor creados automáticamente en cualquier gestor de colas del clúster toman su nombre del canal de clúster receptor correspondiente configurado en el gestor de colas receptor del clúster y, por lo tanto, deben ser exclusivos y tener sentido *en los gestores de colas remotos del clúster.*

Un enfoque común es utilizar el nombre del gestor de colas precedido por el nombre del clúster. Por ejemplo, si el nombre de clúster es CLUSTER1 y los gestores de colas son QM1, QM2, los canales de clúster receptor son CLUSTER1.QM1, CLUSTER1.QM2.

Puede ampliar este convenio si los canales tienen prioridades diferentes o utilizan protocolos diferentes. Por ejemplo:

- CLUSTER1.QM1.S1
- CLUSTER1.QM1.N3
- CLUSTER1.QM1.T4

En este ejemplo, S1 podría ser el primer canal SNA, N3 podría ser el canal NetBIOS con una prioridad de red de tres y T4 podría ser TCP IP utilizando una red IPV4 .

Denominación de definiciones de canal compartido

Una única definición de canal se puede compartir entre varios clústeres, en cuyo caso los convenios de denominación sugeridos aquí necesitarían modificación. Sin embargo, como se describe en Gestión de definiciones de canal , normalmente es preferible definir canales discretos para cada clúster en cualquier caso.

Convenios de denominación de canales más antiguos

Fuera de los entornos de clúster, históricamente ha sido común utilizar un convenio de denominación 'FROMQM.TO.TARGETQM', por lo que es posible que encuentre que los clústeres existentes han utilizado algo similar (como CLUSTER.TO.TARGET). Esto no se recomienda como parte de un nuevo esquema de denominación de clúster porque reduce aún más los caracteres disponibles para transmitir información 'útil' dentro del nombre de canal.

z/OS Nombres de canal en IBM MQ for z/OS

Puede definir recursos genéricos VTAM o nombres genéricos *Dynamic Domain Name Server* (DDNS). Puede definir nombres de conexión utilizando nombres genéricos. Sin embargo, cuando cree una definición de clúster receptor, no utilice un nombre de conexión genérico.

El problema con la utilización de nombres de conexión genéricos para definiciones de clúster receptor es el siguiente: si define un CLUSRCVR con un CONNAME genérico, no hay garantía de que los canales de CLUSSDR apunten a los gestores de colas que desea. Su CLUSSDR inicial podría terminar apuntando a cualquier gestor de colas del grupo de compartición de colas, no necesariamente uno que aloje un repositorio completo. Si un canal vuelve a intentar una conexión, puede volver a conectarse a un gestor de colas diferente con el mismo nombre genérico, interrumpiendo el flujo de mensajes.

z/OS *Queue sharing groups and clusters*

Shared queues can be cluster queues and queue managers in a queue sharing group can also be cluster queue managers.

On IBM MQ for z/OS you can group queue managers into queue sharing groups. A queue manager in a queue sharing group can define a local queue that is to be shared by up to 32 queue managers.

Shared queues can also be cluster queues. Furthermore, the queue managers in a queue sharing group can also be in one or more clusters.

Puede definir recursos genéricos VTAM o nombres genéricos *Dynamic Domain Name Server* (DDNS). Puede definir nombres de conexión utilizando nombres genéricos. Sin embargo, cuando cree una definición de clúster receptor, no utilice un nombre de conexión genérico.

El problema con la utilización de nombres de conexión genéricos para definiciones de clúster receptor es el siguiente: si define un CLUSRCVR con un CONNAME genérico, no hay garantía de que los canales de CLUSSDR apunten a los gestores de colas que desea. Su CLUSSDR inicial podría terminar apuntando a cualquier gestor de colas del grupo de compartición de colas, no necesariamente uno que aloje un repositorio completo. Si un canal vuelve a intentar una conexión, puede volver a conectarse a un gestor de colas diferente con el mismo nombre genérico, interrumpiendo el flujo de mensajes.

A CLUSRCVR channel that uses the group listener port can not be started because, if this were the case, it would not be possible to tell which queue manager the CLUSRCVR would connect to each time. The cluster system queues on which information is kept about the cluster are not shared. Each queue manager has its own.

Cluster channels are used not only to transfer application messages but internal system messages about the setup of the cluster. Each queue manager in the cluster must receive these internal system messages to participate properly in clustering, so needs its own unique CLUSRCVR channel on which to receive them.

A shared CLUSRCVR could start on any queue manager in the queue sharing group (QSG) and so lead to an inconsistent supply of the internal system messages to the QSG queue managers, meaning none can properly participate in the cluster. To ensure no shared CLUSRCVR channels can be used, any attempt fails with the CSQX502E message.

Solapamiento de clústeres

El solapamiento de clústeres proporciona funciones administrativas adicionales. Utilice listas de nombres para reducir el número de mandatos necesarios para administrar clústeres que se solapan.

Puede crear clústeres que se solapen. Hay varias razones por las que puede definir clústeres que se solapan; por ejemplo:

- Para permitir que organizaciones diferentes tengan su propia administración.
- Para permitir que aplicaciones independientes se administren por separado.
- Para crear clases de servicio.

En Figura 7 en la página 37, el gestor de colas STF2 es miembro de ambos clústeres. Cuando un gestor de colas es miembro de más de un clúster, se pueden utilizar listas de nombres para reducir el número de definiciones que se necesitan. Las listas de nombres contienen una lista de nombres, por ejemplo, nombres de clúster. Puede crear una lista de nombres con los nombres de los clústeres. Especifique la lista de nombres en el mandato `ALTER QMGR` para STF2 con objeto de que sea un gestor de colas de repositorio completo para ambos clústeres.

Si tiene más de un clúster en la red, debe asignarles nombres diferentes. Si se fusionan dos clústeres con el mismo nombre, no es posible separarlos de nuevo. También es una buena idea asignar nombres diferentes a los clústeres y canales. Se distinguen más fácilmente cuando se mira la salida de los mandatos `DISPLAY`. Los nombres de gestor de colas deben ser exclusivos dentro de un clúster para que éste funcione correctamente.

Definir clases de servicio

Imagine una universidad que tiene un gestor de colas para cada miembro del personal y cada estudiante. Los mensajes entre los miembros del personal tienen que desplazarse por canales con una prioridad alta y un gran ancho de banda. Los mensajes entre los estudiantes tienen que desplazarse por canales más lentos y económicos. Puede configurar esta red utilizando técnicas de gestión de colas distribuidas tradicionales. IBM MQ selecciona los canales que va a utilizar mirando el nombre de la cola de destino y el nombre del gestor de colas.

Para distinguir claramente entre el personal y los estudiantes, puede agrupar sus gestores de colas en dos clústeres, tal como se muestra en la Figura 7 en la página 37. IBM MQ traslada mensajes a la cola de reuniones en el clúster de personal sólo a través de canales que están definidos en dicho clúster. Los mensajes para la cola de comentarios en el clúster de estudiantes pasan por canales definidos en dicho clúster y reciben la clase de servicio adecuada.

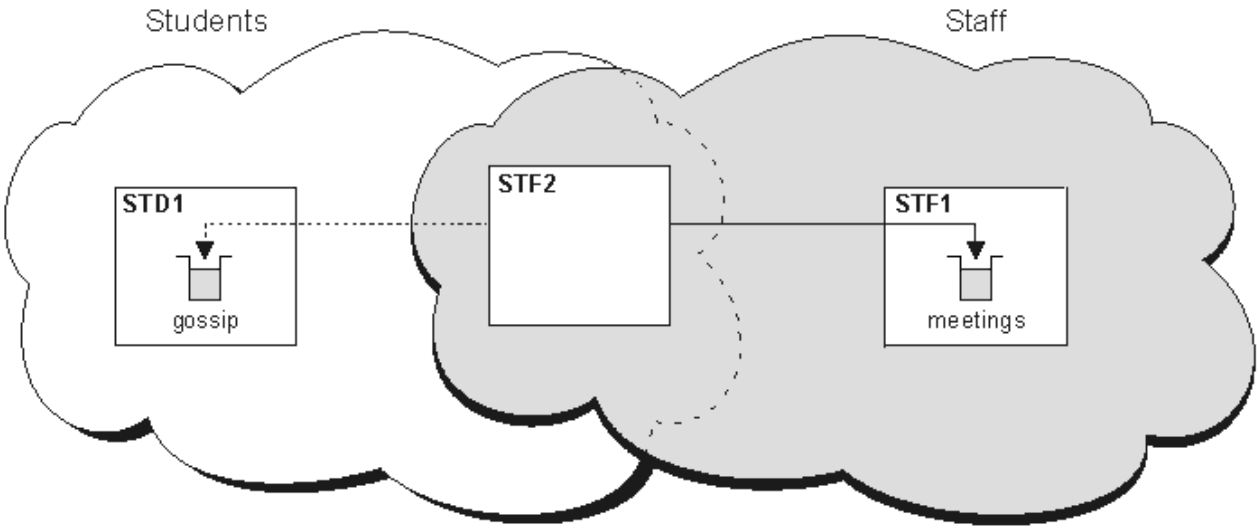


Figura 7. Clases de servicio

Consejos para la agrupación en clúster

Es posible que tenga que realizar algunos cambios en sus sistemas o aplicaciones antes de utilizar la agrupación en clúster. Hay tanto similitudes como diferencias con respecto al comportamiento de la gestión de colas distribuidas.

- Debe añadir definiciones de configuración manuales a los gestores de colas fuera de un clúster para que puedan acceder a las colas de clúster.
- Si fusiona dos clústeres con el mismo nombre, no puede separarlos de nuevo. Por lo tanto, es aconsejable asignar a todos los clústeres un nombre exclusivo.
- Si un mensaje llega a un gestor de colas, pero no hay ninguna cola allí para recibirlo, el mensaje se coloca en la cola de mensajes no entregados. Si no hay ninguna cola de mensajes no entregados, el

canal falla y vuelve a intentarlo. El uso de la cola de mensajes no entregados es el mismo que con la gestión de colas distribuidas.

- La integridad de los mensajes persistentes se mantiene. No se duplican ni se pierden mensajes como resultado de utilizar clústeres.
- La utilización de clústeres reduce la administración del sistema. Los clústeres facilitan la conexión de redes más grandes con muchos más gestores de colas de lo que sería capaz de contemplar utilizando la gestión de colas distribuidas. Existe el riesgo de que pueda consumir demasiados recursos de red si intenta habilitar la comunicación entre todos los gestores de colas de un clúster.
- Si utiliza IBM MQ Explorer, que presenta los gestores de colas en una estructura de árbol, la vista para clústeres grandes puede ser difícil de manejar.
- Multi El propósito de las listas de distribución es utilizar un único mandato MQPUT para enviar el mismo mensaje a varios destinos. Las listas de distribución solo están soportadas en IBM MQ for Multiplatforms. Puede utilizar listas de distribución con clústeres de gestores de colas. En un clúster, todos los mensajes se expanden en el momento de la llamada MQPUT. La ventaja, en términos de tráfico de la red, no es tan grande como en un entorno no de clúster. La ventaja de las listas de distribución es que no es necesario definir manualmente los numerosos canales y colas de transmisión.
- Si va a utilizar clústeres para equilibrar la carga de trabajo, examine sus aplicaciones. Vea si estas requieren que los mensajes sean procesados por un gestor de colas específico o en una secuencia determinada. Se dice que estas aplicaciones tienen afinidades de mensajes. Es posible que tenga que modificar las aplicaciones antes de poder utilizarlas en clústeres complejos.
- Puede decidir utilizar la opción MQOO_BIND_ON_OPEN en una llamada MQOPEN para forzar el envío de los mensajes a un destino específico. Si el gestor de colas de destino no está disponible, los mensajes no se entregan hasta que el gestor de colas vuelve a estar disponible. Los mensajes no se direccionan a otro gestor de colas debido al riesgo de duplicación.
- Si un gestor de colas va a alojar un repositorio de clúster, debe saber su nombre de host o dirección IP. Tiene que especificar esta información en el parámetro CONNAME cuando realice la definición CLUSSDR en otros gestores de colas que se unan al clúster. Si utiliza DHCP, la dirección IP está sujeta a cambios, ya que DHCP puede asignar una nueva dirección IP cada vez que reinicie un sistema. Por lo tanto, no debe especificar la dirección IP en las definiciones CLUSSDR. Aunque todas las definiciones CLUSSDR especificaran el nombre de host en lugar de la dirección IP, las definiciones seguirían sin ser fiables. DHCP no necesariamente actualiza la entrada de directorio DNS del host con la nueva dirección. Si debe designar gestores de colas como repositorios completos en sistemas que utilizan DHCP, instale software que garantice que el directorio DNS se mantiene actualizado.
- No utilice nombres genéricos, por ejemplo recursos genéricos VTAM o nombres genéricos DDNS (Dynamic Domain Name Server), como los nombres de conexión de los canales. Si lo hace, los canales podrían conectarse a un gestor de colas diferente del esperado.
- Sólo puede obtener un mensaje de una cola de clúster local, pero puede transferir un mensaje a cualquier cola de un clúster. Si abre una cola para utilizar el mandato MQGET, el gestor de colas abre la cola local.
- No necesita modificar ninguna de sus aplicaciones si configura un clúster de IBM MQ simple. La aplicación puede nombrar la cola de destino en la llamada MQOPEN y no necesita saber la ubicación del gestor de colas. Si configura un clúster para la gestión de carga de trabajo, debe revisar sus aplicaciones y modificarlas según sea necesario.
- Puede ver datos de supervisión y de estado actuales para un canal o cola mediante los mandatos **runmqsc** DISPLAY CHSTATUS y DISPLAY QSTATUS. La información de supervisión se puede utilizar para ayudar a medir el rendimiento y el estado del sistema. La supervisión se controla mediante atributos de gestor de colas, de cola y de canal. La supervisión de canales de clúster emisor definidos automáticamente es posible con el atributo de gestor de colas MONACLS.

Conceptos relacionados

Clústeres

“Comparación de agrupación en clúster y gestión de colas distribuidas” en la página 30

Compare los componentes que deben definirse para conectar gestores de colas utilizando la gestión de colas distribuidas y la agrupación en clúster

Componentes de un clúster

Tareas relacionadas

Configuración de un clúster de gestores de colas

Configurar un nuevo clúster

¿Cuánto tiempo conservan los depósitos de gestor de colas la información?

Los repositorios de gestor de colas conservan la información durante 30 días. Un proceso automático renueva eficientemente la información que se está utilizando.

Cuando un gestor de colas envía información sobre sí mismo, los gestores de colas de repositorio completo y parcial almacenan la información durante 30 días. La información se envía, por ejemplo, cuando un gestor de colas anuncia la creación de una nueva cola. Para evitar que esta información caduque, los gestores de colas vuelven a enviar automáticamente toda la información sobre sí mismos al cabo de 27 días. Si un repositorio parcial envía una nueva solicitud de información una vez iniciado el periodo de tiempo de 30 días, el tiempo de caducidad sigue siendo los 30 días originales.

Cuando la información caduca, no se elimina inmediatamente del repositorio. En su lugar, se conserva durante un período de gracia de 60 días. Si no se recibe ninguna actualización dentro del período de gracia, la información se elimina. El periodo de gracia tiene en cuenta el hecho de que un gestor de colas puede haber estado temporalmente fuera de servicio en la fecha de caducidad. Si un gestor de colas se desconecta de un clúster durante más de 90 días, deja de formar parte del clúster. Sin embargo, si se vuelve a conectar a la red, vuelve a formar parte del clúster. Los repositorios completos no utilizan la información que ha caducado para atender nuevas solicitudes de otros gestores de colas.

De forma similar, cuando un gestor de colas envía una solicitud para obtener información actualizada de un repositorio completo, la solicitud tiene una duración de 30 días. Después de 27 días IBM MQ comprueba la solicitud. Si se ha hecho referencia a ella durante los 27 días, se renueva automáticamente. Si no, se deja que caduque y el gestor de colas la renueva si se necesita de nuevo. La caducidad de las solicitudes evita la acumulación de solicitudes de información de los gestores de colas latentes.

Nota: Debe descargar e instalar el PTF para APAR PH43191, que corrige los errores del sistema al calcular la hora de caducidad de una suscripción. Estos errores pueden hacer que la suscripción caduque antes (lo que da como resultado que se emita el mensaje CSQX456I) o que caduque después de que el objeto haya caducado (lo que da como resultado errores MQRC 2085 (MQRC_UNKNOWN_OBJECT) erróneos).

Para los clústeres de gran tamaño, puede producirse una interrupción si muchos gestores de colas reenvían automáticamente toda la información sobre sí mismos al mismo tiempo. Consulte La renovación en un clúster grande puede afectar el rendimiento y la disponibilidad del clúster.

Conceptos relacionados

“Agrupación en clúster: utilización de las recomendaciones de REFRESH CLUSTER” en la página 70

Puede utilizar el mandato **REFRESH CLUSTER** para descartar toda la información retenida localmente sobre un clúster y reconstruir esa información a partir de los repositorios completos en el clúster. Es poco probable que necesite utilizar este mandato, excepto en circunstancias excepcionales. Si necesitara utilizar este mandato, existen algunas consideraciones especiales sobre cómo se utiliza. Esta información es una guía basada en las pruebas y los comentarios de los clientes.

Clústeres de ejemplo

El primer ejemplo muestra el clúster más pequeño posible de dos gestores de colas. Los ejemplos segundo y tercero muestran dos versiones de un clúster con tres gestores de colas.

El clúster más pequeño posible contiene sólo dos gestores de colas. En este caso, ambos gestores de colas contienen repositorios completos. Sólo necesita unas pocas definiciones para configurar el clúster y todavía hay un alto grado de autonomía en cada gestor de colas.

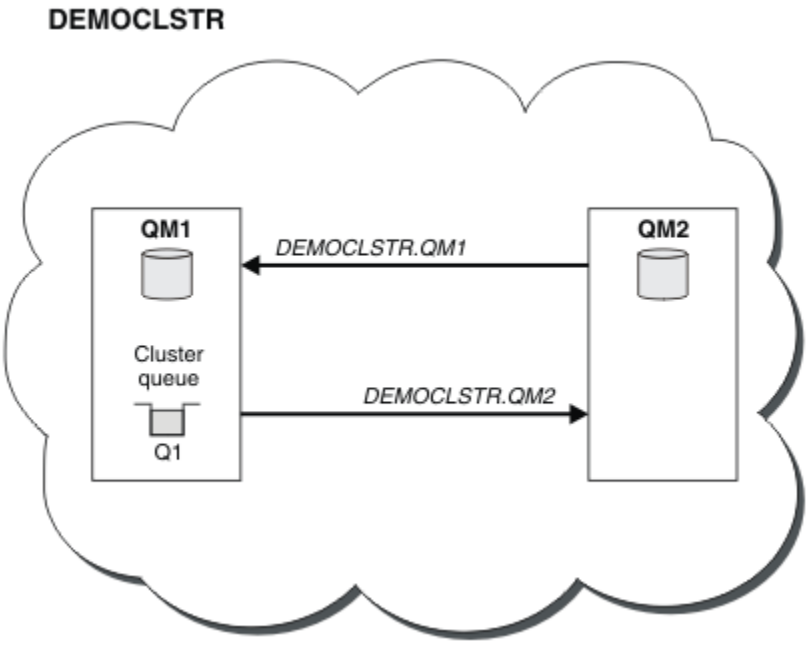


Figura 8. Un clúster pequeño con dos gestores de colas

- Los gestores de colas pueden tener nombres largos como LONDON y NEWYORK. z/OS En IBM MQ for z/OS, los nombres del gestor de colas están limitados a cuatro caracteres.
- Cada gestor de colas se configura normalmente en una máquina distinta. Sin embargo, puede tener varios gestores de colas en la misma máquina.

Si desea ver instrucciones sobre cómo configurar un clúster de ejemplo similar, consulte Configurar un nuevo clúster.

Figura 9 en la página 41 muestra los componentes de un clúster denominado CLSTR1.

- En este clúster, hay tres gestores de colas, QM1, QM2 y QM3.
- Los repositorios de host QM1 y QM2 de información sobre todos los gestores de colas y los objetos relacionados con el clúster en el clúster. Se denominan como *gestores de colas de repositorio completo*. Los repositorios están representados en el diagrama por cilindros sombreados.
- QM2 y QM3 alojan algunas colas que son accesibles para cualquier otro gestor de colas del clúster. Las colas a las que puede acceder cualquier otro gestor de colas en el clúster se denominan *colas de clúster*. Las colas de clúster están representadas en el diagrama por colas sombreadas. A las colas de clúster se pueden acceder desde cualquier lugar del clúster. El código de agrupación en clúster de IBM MQ asegura que las definiciones de cola remota se crean en cualquier gestor de colas que hace referencia a las mismas.

  Al igual que con la agrupación en colas distribuidas, una aplicación utiliza la llamada MQPUT para colocar un mensaje en una cola de clúster en cualquier gestor de colas del clúster. Una aplicación utiliza la llamada MQGET para recuperar mensajes de una cola de clúster sólo en el gestor de colas donde reside la cola.
- Cada gestor de colas tiene una definición creada manualmente para el extremo receptor de un canal denominado *cluster_name. queue_manager_name* en el que puede recibir mensajes. En el gestor de colas receptor, *cluster_name. queue_manager_name* es un canal de clúster receptor. Un canal de clúster receptor es como un canal receptor utilizado en la agrupación de colas distribuidas; recibe mensajes para el gestor de colas. Además, también recibe información sobre el clúster.

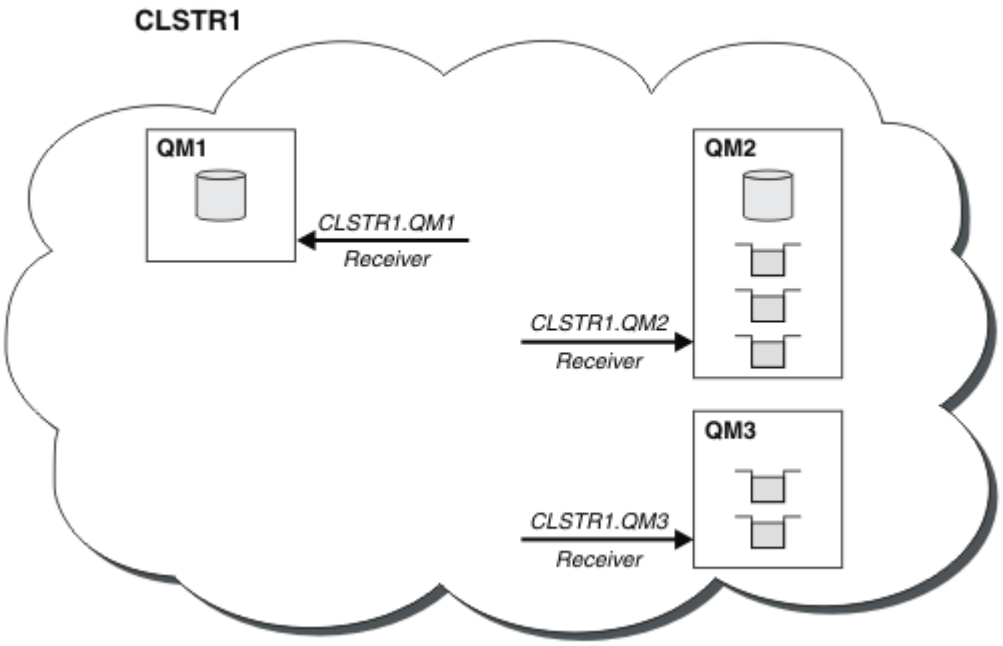
- 

Figura 9. Un clúster de gestores de colas

- En la Figura 10 en la página 42 cada gestor de colas también tiene una definición para el extremo emisor de un canal. Se conecta al canal de clúster receptor de uno de los gestores de colas de repositorio completo. En el gestor de colas emisor, *cluster_name. queue_manager_name* es un canal de clúster emisor. QM1 y QM3 tienen canales de clúster emisor que se conectan a CLSTR1.QM2, consulte la línea de puntos "2".

  QM2 tiene un canal de clúster emisor que se conecta a CLSTR1.QM1, consulte la línea de puntos "3". Un canal de clúster emisor es como un canal emisor utilizado en la agrupación de colas distribuidas; envía mensajes al gestor de colas receptor. Además, también envía información sobre el clúster.

  Una vez que se definen tanto el extremo del clúster receptor, como el extremo del clúster emisor, el canal se inicia automáticamente.

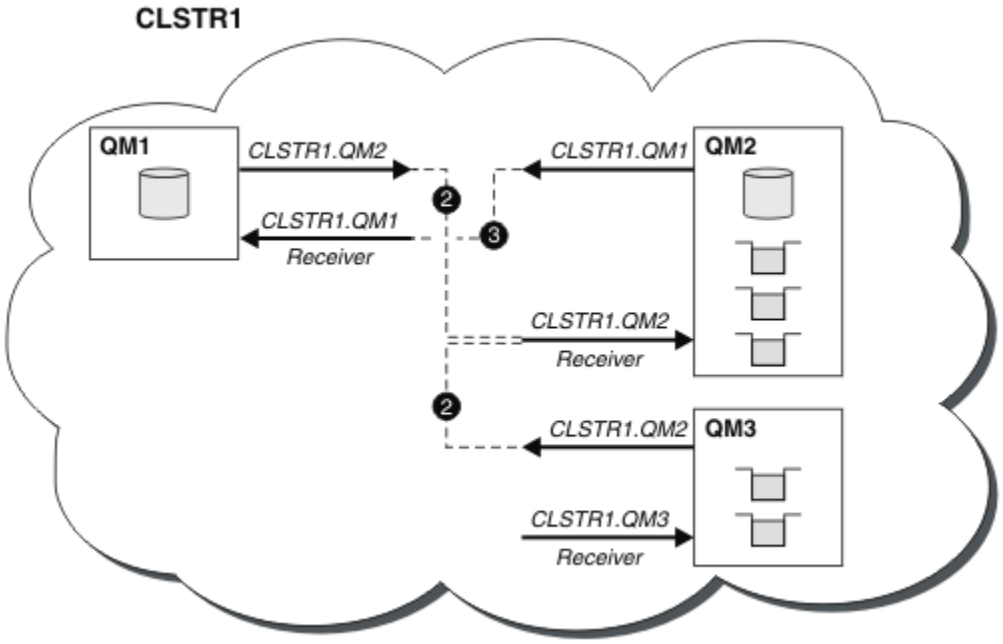


Figura 10. Un clúster de gestores de colas con canales emisor

Definir un canal de clúster emisor en el gestor de colas local presenta dicho gestor de colas a uno de los gestores de colas de repositorio completo. El gestor de colas de repositorio completo actualiza la información en su repositorio completo de forma consecuente. Automáticamente, vuelve a crear un canal de clúster emisor en el gestor de colas original y envía la información de dicho gestor de colas sobre el clúster. De esta forma un gestor de colas obtiene información sobre un clúster y un clúster sobre un gestor de colas.

Busque de nuevo en la Figura 9 en la página 41. Suponga que una aplicación conectada al gestor de colas QM3 desea enviar algunos mensajes a las colas en QM2. La primera vez que QM3 debe acceder a esas colas, las descubre consultando un repositorio completo. El repositorio completo en este caso es QM2, al que se accede utilizando el canal emisor CLSTR1.QM2. Con la información del repositorio, puede crear automáticamente definiciones remotas para dichas colas. Si las colas están en QM1, este mecanismo todavía funciona, porque QM2 es un repositorio completo. Un repositorio completo tiene un registro completo de todos los objetos del clúster. En este último caso, QM3 también crearía automáticamente un canal de clúster emisor correspondiente al canal de clúster receptor en QM1, lo que permite la comunicación directa entre los dos.

La Figura 11 en la página 43 muestra el mismo clúster, con dos canales de clúster emisor que se crearon automáticamente. Los canales de clúster emisor están representados por las dos líneas de guiones que se unen al canal de clúster receptor CLSTR1.QM3. También muestra la cola de transmisión de clúster, SYSTEM.CLUSTER.TRANSMIT.QUEUE, que QM1 utiliza para enviar sus mensajes. Todos los gestores de colas del clúster tienen una cola de transmisión de clúster, desde la cual pueden enviar mensajes a cualquier otro gestor de colas del mismo clúster.

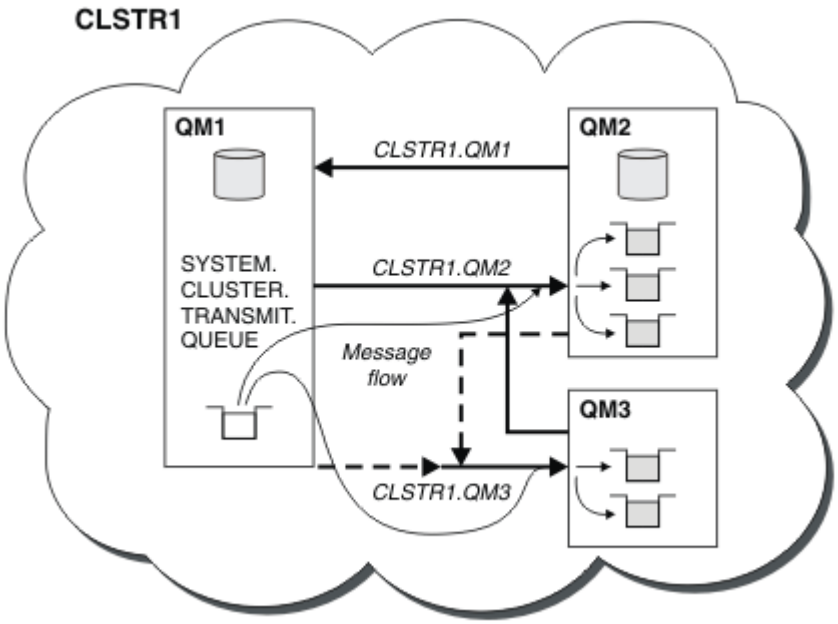


Figura 11. Un clúster de gestores de colas, que muestra canales definidos automáticamente

Nota: Otros diagramas muestran sólo los extremos de recepción de los canales para los cuales realiza definiciones manuales. Los extremos de emisor se omiten porque en su mayoría se definen automáticamente cuando sea necesario. La definición automática de la mayoría de canales de clúster emisor es crucial para el funcionamiento y la eficacia de los clústeres.

Conceptos relacionados

"Comparación de agrupación en clúster y gestión de colas distribuidas" en la página 30
Compare los componentes que deben definirse para conectar gestores de colas utilizando la gestión de colas distribuidas y la agrupación en clúster

Componentes de un clúster

Tareas relacionadas

Configuración de un clúster de gestores de colas

Configurar un nuevo clúster

Agrupación en clúster: procedimientos recomendados

Los clústeres proporcionan un mecanismo para interconectar gestores de colas. Las mejores prácticas descritas en esta sección se basan en pruebas y comentarios de clientes.

Una configuración de clúster correcta depende de una buena planificación y unos amplios conocimientos de los conceptos básicos de IBM MQ como, por ejemplo, una buena gestión de aplicaciones y un buen diseño de red. Asegúrese de que está familiarizado con la información de los temas relacionados antes de continuar.

Conceptos relacionados

Gestión de colas distribuidas y clústeres

Clústeres

Tareas relacionadas

"Diseño de clústeres" en la página 24
Los clústeres proporcionan un mecanismo para interconectar gestores de colas de forma que simplifica la configuración inicial y la gestión continua. Los clústeres se deben diseñar con mucho cuidado para asegurarse de que funcionan correctamente y de que pueden alcanzar los niveles necesarios de disponibilidad y capacidad de respuesta.

Supervisión de clústeres

Agrupación en clúster: consideraciones especiales para los clústeres que se solapan
En este tema se proporcionan instrucciones para planificar y administrar clústeres de IBM MQ. Esta información es una guía basada en las pruebas y los comentarios de los clientes.

Propiedad de clúster

Familiarícese con el solapamiento de clústeres antes de leer la información siguiente. Consulte “Solapamiento de clústeres” en la página 36 and Configurar vías de acceso de mensajes entre clústeres para obtener la información necesaria.

Al configurar y gestionar un sistema que consta de clústeres que se solapan, es mejor ajustarse a lo siguiente:

- Aunque los clústeres de IBM MQ están 'débilmente acoplados' tal como se ha descrito anteriormente, resulta útil considerar un clúster como una sola unidad de administración. Este concepto se utiliza porque la interacción entre definiciones en los gestores de colas individuales es muy importante para el buen funcionamiento del clúster. Por ejemplo: Al utilizar colas de clúster de carga equilibrada es importante que un solo administrador o equipo conozca el conjunto completo de posibles destinos para los mensajes, que depende de las definiciones distribuidas por todo el clúster. Más trivialmente, los pares de canales de clúster emisor/receptor deben ser totalmente compatibles.
- Teniendo en cuenta este concepto anterior; cuando se encuentran varios clústeres (que deben ser administrados por equipos / personas diferentes), es importante disponer de políticas claras que controlen la administración de los gestores de colas de pasarela.
- Resulta útil tratar los clústeres que se solapan como un único espacio de nombres: los nombres de canal y los nombres de gestor de colas deben ser exclusivos en todo un solo clúster. La administración es mucho más fácil cuando son exclusivos en toda la topología. Es mejor seguir un convenio de denominación adecuado; los posibles convenios se describen en “Convenios de denominación de clústeres” en la página 34.
- A veces es esencial la cooperación administrativa y de gestión de sistemas. Por ejemplo, la cooperación entre organizaciones que poseen diferentes clústeres que deben solaparse. Una clara comprensión de quién es propietario de qué y las reglas y convenios aplicables ayudan a que la agrupación en clúster se ejecute sin problemas cuando se solapan los clústeres.

Clústeres que se solapan: pasarelas

En general, un solo clúster es más fácil de administrar que varios clústeres. Por lo tanto, la creación de grandes cantidades de pequeños clústeres (uno para cada aplicación, por ejemplo) es algo que se debe evitar en general.

Sin embargo, para proporcionar clases de servicio, puede implementar clústeres solapados. Por ejemplo:

- Si tiene clústeres concéntricos donde el más pequeño es para Publicación/Suscripción. Consulte Cómo dimensionar sistemas si desea más información.
- Si algunos gestores de colas van a ser administrados por equipos diferentes. Consulte la sección anterior “Propiedad de clúster” en la página 44para obtener más información.
- Si tiene sentido desde le punto de vista geográfico o de la organización.
- Si los clústeres equivalentes funcionan con resolución de nombres, por ejemplo, al implementar TLS en un clúster existente.

No hay ninguna ventaja de seguridad de los clústeres solapados; permitiendo que los clústeres administrados por dos equipos diferentes se solapen, se une de forma efectiva a los equipos, así como a la topología:

- Cualquier nombre anunciado en un clúster de este tipo es accesible para el otro clúster.
- Cualquier nombre anunciado en un clúster se puede anunciar en el otro para extraer mensajes elegibles.

- Cualquier objeto no anunciado en un gestor de colas adyacente a la pasarela se puede resolver desde cualquier clúster del que la pasarela sea miembro.

El espacio de nombres es la unión de ambos clústeres y debe ser tratado como un único espacio de nombres. Por lo tanto, la propiedad de un clúster que se solapa es compartida entre todos los administradores de ambos clústeres.

Cuando un sistema contiene varios clústeres, puede haber un requisito de direccionar mensajes desde los gestores de colas de un clúster a las colas en los gestores de colas de otro clúster. En esta situación, los múltiples clústeres deben estar interconectados de alguna manera: un buen patrón a seguir es utilizar gestores de colas de pasarela entre los clústeres. Esta organización evita la creación de una malla de canales de punto a punto difícil de gestionar y proporciona un buen lugar para gestionar cuestiones como las políticas de seguridad. Hay dos modos distintos de conseguir esta organización:

1. Coloque uno o varios gestores de colas en ambos clústeres utilizando una segunda definición de clúster receptor. Esta organización implica menos definiciones administrativas, pero, tal como se dijo anteriormente, significa que la propiedad de un clúster que se solapa es compartida entre todos los administradores de ambos clústeres.
2. Empareje un gestor de colas en el clúster 1 con un gestor de colas en el clúster 2 utilizando canales punto a punto tradicionales.

En cualquiera de estos casos, se pueden utilizar varias herramientas para direccionar el tráfico de forma adecuada. En concreto, los alias de cola o de gestor de colas se pueden utilizar para direccionar al otro clúster, y un alias de gestor de colas con la propiedad **RQMNAME** en blanco vuelve a generar el equilibrio de la carga de trabajo donde se desea.

Conceptos relacionados

"Convenios de denominación de clústeres" en la página 34
Considere la posibilidad de denominar gestores de colas en el mismo clúster utilizando un convenio de denominación que identifique el clúster al que pertenece el gestor de colas. Utilice un convenio de denominación similar para los nombres de canal y amplíelo para describir las características del canal.

Agrupación en clúster: consideraciones sobre el diseño de topologías
En este tema se proporcionan instrucciones para planificar y administrar clústeres de IBM MQ. Esta información es una guía basada en las pruebas y los comentarios de los clientes.

Al reflexionar de antemano sobre dónde se ubicarán las aplicaciones de usuario y los procesos administrativos internos, se pueden evitar muchos problemas o minimizar en un futuro. Este tema contiene información sobre decisiones de diseño que pueden mejorar el rendimiento y simplificar las tareas de mantenimiento a medida que se amplía el clúster.

- "Rendimiento de la infraestructura de clústeres" en la página 45
- "Depósitos completos" en la página 46
- "¿Las aplicaciones deben utilizar colas en repositorios completos?" en la página 47
- "Gestión de definiciones de canal" en la página 48
- "Equilibro de carga de trabajo a través de varios canales" en la página 48

Rendimiento de la infraestructura de clústeres

Cuando una aplicación intenta abrir una cola en un gestor de colas en un clúster, el gestor de colas registra su interés con los repositorios completos para dicha cola, para que pueda aprender dónde existe la cola en el clúster. Los repositorios completos envían automáticamente las actualizaciones en la ubicación de la cola o la configuración al gestor de colas interesados. Este registro de interés se conoce internamente como una suscripción (estas suscripciones no son las mismas que las suscripciones de IBM MQ utilizadas para la mensajería de publicación/suscripción en IBM MQ)

Toda la información sobre un clúster pasa por cada repositorio completo. Por lo tanto, los repositorios completos siempre se están utilizado en un clúster para el tráfico de mensajes administrativos. El elevado uso de los recursos del sistema cuando se gestionan estas suscripciones, y la transmisión de los mismos y los mensajes de configuración resultantes, puede provocar una considerable carga en la infraestructura

de clústeres. Se deben tener en cuenta varios factores para garantizar que esta carga se reconozca y minimice siempre que sea posible:

- Cuantos más gestores de colas individuales haya utilizando una cola de clúster, más suscripciones hay en el sistema y, por lo tanto, mayor será la carga administrativa cuando se produzcan cambios y se deban notificar a los suscriptores interesados, especialmente en los gestores de colas de repositorio completo. Una forma de minimizar el tráfico innecesario y la carga del repositorio completo es conectando las aplicaciones similares (es decir, las aplicaciones que trabajan con las mismas colas) a un número menor de gestores de colas.
- Además del número de suscripciones en el sistema que afectan al rendimiento, la tasa de cambio en la configuración de objetos en clúster puede afectar al rendimiento, por ejemplo, el cambio frecuente de una configuración de colas en clúster.
- Cuando un gestor de colas es miembro de varios clústeres (es decir, forma parte de un sistema de clústeres que se solapan), cualquier interés creado en una cola da como resultado una suscripción para cada clúster del que es miembro, aunque los mismos gestores de colas sean los repositorios completos de más de uno de los clústeres. Esta disposición aumenta la carga en el sistema y es una de las razones para considerar si son necesarios varios clústeres solapados, en lugar de un solo clúster.
- El tráfico de mensajes de aplicaciones (es decir, los mensajes que envían las aplicaciones de IBM MQ a las colas de clúster) no pasan por los repositorios completos para llegar a los gestores de colas de destino. Este tráfico de mensajes se envía directamente entre el gestor de colas donde el mensaje entra en el clúster y el gestor de colas donde existe la cola de clúster. Por lo tanto, no es necesario permitir tasas elevadas de tráfico de mensajes de aplicación respecto a los gestores de colas de repositorio completo, a menos que los gestores de colas de repositorio completo sean uno de esos dos gestores de colas citados. Por este motivo, se recomienda no utilizar los gestores de colas de repositorio completo para el tráfico de mensajes de aplicación en los clústeres donde la carga de la infraestructura de clústeres sea significativa.

Depósitos completos

Un repositorio es una recopilación de información sobre los gestores de colas que son miembros de un clúster. Un gestor de colas que aloja un conjunto completo de información sobre todos los gestores de colas del clúster tiene un repositorio completo. Para obtener más información sobre repositorios completos y repositorios parciales, consulte Repositorio de clúster.

Los repositorios completos deben mantenerse en servidores que sean fiables y tengan la máxima disponibilidad posible, y deben evitarse los puntos únicos de anomalía. El diseño del clúster siempre debe tener dos repositorios. Si se produce un error en un repositorio completo, el clúster puede seguir funcionando.

Detalles de las actualizaciones en los recursos de clúster realizadas por un gestor de colas en un clúster; por ejemplo, las colas en clúster se envían desde ese gestor de colas a dos repositorios completos como máximo en ese clúster (o a uno si sólo hay un gestor de colas de repositorio completo en el clúster). Estos repositorios completos contienen la información y la propagan a los gestores de colas del clúster que muestran un interés en ella (es decir, que se suscriben a ella). Para garantizar que cada miembro del clúster tenga una vista actualizada de los recursos del clúster, cada gestor de colas debe poder comunicarse con al menos un gestor de colas de repositorio completo en cualquier momento.

Si por cualquier motivo un gestor de colas no puede comunicarse con ningún repositorio completo, puede continuar funcionando en el clúster según su nivel de información almacenado en la memoria caché durante un período de tiempo, pero no hay disponibles nuevas actualizaciones ni el acceso a los recursos de clúster no utilizados anteriormente.

Por este motivo, debe intentar mantener los dos repositorios completos disponibles en todo momento. No obstante, esta disposición no significa que deban tomarse precauciones extremas, porque el clúster funciona adecuadamente durante un breve periodo de tiempo sin un repositorio completo.

Hay otro motivo por el que un clúster debe tener dos gestores de colas de repositorio completo, aparte de la disponibilidad de la información del clúster: garantizar que la información de clúster contenida en la memoria caché del repositorio completo exista en dos lugares a efectos de recuperación. Si sólo hay un repositorio completo y éste pierde su información sobre el clúster, se requiere intervención manual

en todos los gestores de colas del clúster para que el clúster pueda funcionar de nuevo. En cambio, si hay dos repositorios completos, dado que la información siempre se publica y se suscribe desde dos repositorios completos, el repositorio completo que falla puede recuperarse con un esfuerzo mínimo.

- Es posible realizar tareas de mantenimiento en gestores de colas de repositorio completo en un diseño de clúster de dos repositorios completos sin afectar a los usuarios del clúster: el clúster sigue funcionando con sólo uno de los repositorios, por lo que, siempre que sea posible, debe dejar fuera de servicio los repositorios, aplicar el mantenimiento y volver a ponerlos en servicio uno a uno. Aun cuando se produzca un corte de alimentación en el segundo repositorio completo, la ejecución de las aplicaciones no se ve afectada como mínimo durante tres días.
- A menos que haya un buen motivo para utilizar un tercer repositorio como, por ejemplo, el uso de un repositorio completo local geográficamente por razones geográficas, utilice el diseño de dos repositorios. Tener tres repositorios completos significa que nunca sabe cuáles son los dos repositorios en uso, y es posible que surjan problemas administrativos provocados por las interacciones entre varios parámetros de gestión de la carga de trabajo. No es recomendable tener más de dos repositorios completos.
- Si todavía necesita una mayor disponibilidad, considere la posibilidad de alojar los gestores de colas de repositorios completos como gestores de colas multiinstancia o utilizar el soporte de alta disponibilidad específico de la plataforma para mejorar su disponibilidad.
- Debe conectar entre sí y por completo todos los gestores de colas de repositorio completo con los canales emisor de clúster definidos manualmente. Tenga especial cuidado cuando el clúster no tenga, por alguna razón justificable, más de dos repositorios completos. En este caso, se pueden perder uno o varios canales y que no sea perceptible inmediatamente. Cuando no se produce la interconexión completa, a menudo pueden surgir problemas difícil de diagnosticar. Son difíciles de diagnosticar porque algunos repositorios completos no contienen todos los datos del repositorio y, por lo tanto, los gestores de colas en el clúster tienen vistas diferentes del clúster, según los repositorios completos a los que se conecten.

¿Las aplicaciones deben utilizar colas en repositorios completos?

Un repositorio completo en gran parte es igual a cualquier otro gestor de colas, por lo que es posible alojar colas de aplicación en el repositorio completo y conectar las aplicaciones directamente a estos gestores de colas. ¿Las aplicaciones deben utilizar colas en repositorios completos?

La respuesta comúnmente aceptada es "No?". Aunque esta configuración es posible, muchos clientes prefieren mantener estos gestores de colas dedicados a mantener la memoria caché del clúster de repositorio completo. A continuación, se describen los puntos a tener en cuenta a la hora de decidir qué opción desea utilizar, pero en última instancia la arquitectura del clúster debe ser adecuada para las demandas concretas del entorno.

- Actualizaciones: normalmente, para poder utilizar las nuevas características de clúster en los nuevos releases de IBM MQ, los gestores de colas de repositorio completo de ese clúster deben actualizarse primero. Cuando una aplicación del clúster necesita utilizar características nuevas, puede ser útil poder actualizar los repositorios completos (y algún subconjunto de repositorios parciales) sin probar una serie de aplicaciones coubicadas.
- Mantenimiento: de forma similar, si debe aplicar un mantenimiento urgente a los repositorios completos, éstos pueden reiniciarse o actualizarse con el mandato **REFRESH** sin tocar las aplicaciones.
- Rendimiento: a medida que crecen los clústeres y que la demanda de mantenimiento de la memoria caché de clúster de repositorio completo es mayor, mantener las aplicaciones separadas reduce el riesgo de que esto afecte al rendimiento de la aplicación mediante la contienda de recursos del sistema.
- Requisitos de hardware: normalmente, no es necesario que los repositorios completos sean potentes; por ejemplo, un servidor UNIX con buenas expectativas de disponibilidad es suficiente. De manera alternativa, para los clústeres muy grandes o en constante cambio, debe tenerse en cuenta el rendimiento del sistema de repositorio completo.

- Requisitos de software: los requisitos son generalmente la razón principal para elegir alojar las colas de aplicación en un repositorio completo. En un clúster pequeño, la asignación puede significar un requisito de menos servidores/gestores de colas en general.

Gestión de definiciones de canal

Incluso en un solo clúster, pueden existir varias definiciones de canal que proporcionen distintas rutas entre dos gestores de colas.

A veces supone una ventaja tener canales paralelos dentro de un clúster individual, pero esta decisión de diseño tiene que analizarse con atención; aparte de añadir complejidad, este diseño puede dar como resultado que se infrautilicen los canales, lo que da lugar a una caída del rendimiento. Esta situación se produce porque generalmente las pruebas implican el envío de muchos mensajes a un ritmo constante, por lo que los canales paralelos se utilizan por completo. En cambio, en condiciones reales de un flujo no constante de mensajes, el algoritmo de equilibrado de carga hace que el rendimiento baje a medida que el flujo de mensajes conmuta de canal a canal.

Cuando un gestor de colas es miembro de varios clústeres, existe la opción de utilizar una definición de canal única con una lista de nombres de clúster, en lugar de definir un canal CLUSRCVR por separado para cada clúster. Sin embargo, esta configuración puede dar problemas de administración más adelante; por ejemplo, en el caso de que TLS se aplique a un clúster, pero no a un segundo. Por tanto, es preferible crear definiciones separadas, y el convenio de denominación que se recomienda en "Convenios de denominación de clústeres" en la página 34 da soporte a esta opción.

Equilibro de carga de trabajo a través de varios canales

Esta información está concebida como conocimientos avanzados del tema. Para obtener la descripción básica de este tema (que debe comprenderse antes de utilizar la información que se indica aquí), consulte Utilización de clústeres para la gestión de carga de trabajo, Equilibrio de carga de trabajo en clústeres y Algoritmo de gestión de la carga de trabajo del clúster.

El algoritmo de gestión de carga de trabajo del clúster proporciona un amplio conjunto de herramientas, pero no deben ser utilizadas todas entre sí sin entender completamente cómo funcionan e interactúan. Es posible que no sea inmediatamente evidente la importancia de los canales para el proceso de equilibrio de carga de trabajo: el algoritmo de rotación de carga de trabajo de gestión de carga de trabajo se comporta como si varios canales de clúster a un gestor de colas que es propietario de una cola en clúster se tratasen como varias instancias de esa cola. Este proceso se explica de forma más detallada en el siguiente ejemplo:

1. Hay dos gestores de colas que alojan una cola en un clúster: QM1 y QM2.
2. Hay cinco canales receptores de clúster en QM1.
3. Sólo hay un canal receptor de clúster en QM2.
4. Cuando **MQPUT** o **MQOPEN** en QM3 selecciona una instancia, es cinco veces más probable que el algoritmo envíe el mensaje a QM1 que a QM2.
5. La situación en el paso 4 se produce porque el algoritmo ve seis opciones para elegir entre (5 + 1) e iteraciones cíclicas en los cinco canales a QM1 y el único canal a QM2.

Otro comportamiento sutil es que incluso al colocar mensajes en una cola de clúster que tiene una instancia configurada en el gestor de colas local, IBM MQ utiliza el estado del canal receptor de clúster local para decidir si los mensajes deben colocarse en la instancia local de la cola o en instancias remotas de la cola. En este escenario:

1. Al colocar los mensajes, el algoritmo de gestión de carga de trabajo no examina las colas de clúster individuales, examina los canales de clúster que pueden llegar a esos destinos.
2. Para llegar a los destinos locales, los canales de receptor local se incluyen en esta lista (aunque no se utilizan para enviar el mensaje).
3. Cuando se detiene un canal de receptor local, el algoritmo de gestión de carga de trabajo prefiere una instancia alternativa de forma predeterminada, si su CLUSRCVR no está detenido. Si hay varias

instancias CLUSRCVR locales para el destino y al menos una no se detiene, la instancia local sigue siendo elegible.

Agrupación en clúster: Aislamiento de aplicaciones utilizando varias colas de transmisión de clúster
Puede aislar los flujos de mensajes entre los gestores de colas de un clúster. Puede colocar mensajes transportados por diferentes canales de clúster emisor en diferentes colas de transmisión de clúster. Puede utilizar el enfoque en un solo clúster o con clústeres solapados. El tema proporciona ejemplos y algunas prácticas recomendadas que le guiarán para elegir un procedimiento para utilizarlo.

Cuando despliega una aplicación, puede elegir qué recursos de IBM MQ comparte con otras aplicaciones y qué recursos no comparte. Existe una serie de tipos de recursos que se pueden compartir. Los principales son el propio servidor, el gestor de colas, las canales y las colas. Puede optar por configurar aplicaciones con menos recursos compartidos; asignar diferentes colas, canales, gestores de colas o incluso servidores a aplicaciones individuales. Si lo hace, la configuración global del sistema resultará mayor y más compleja. La utilización de clústeres de IBM MQ reduce la complejidad de gestionar más servidores, gestores de colas, colas y canales, pero introduce otro recurso compartido, la cola de transmisión de clúster, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.

Figura 12 en la página 50 s parte de un gran despliegue de IBM MQ que ilustra la importancia de compartir `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. En el diagrama, la aplicación, `Client App`, está conectada al gestor de colas `QM2` del clúster `CL1`. La aplicación, `Server App` procesa un mensaje de `Client App`. `Server App` recupera el mensaje de la cola de clúster `Q1` en el gestor de colas `QM3` en `CLUSTER2`. Dado que las aplicaciones cliente y servidor no están en el mismo clúster, el gestor de colas de pasarela `QM1` transfiere el mensaje.

La forma normal de configurar un clúster de pasarela es convertir el gestor de colas de pasarela en miembro de todos los clústeres. En el gestor de colas de pasarela están definidas colas alias de clúster para colas de clúster en todos los clústeres. Los alias de cola de clúster están disponibles en todos los clústeres. Los mensajes transferidos a los alias de cola de clúster se direccionan a través del gestor de colas de pasarela a su destino correcto. El gestor de colas de pasarela coloca los mensajes enviados a las colas de alias en clúster en `SYSTEM.CLUSTER.TRANSMIT.QUEUE` común en `QM1`.

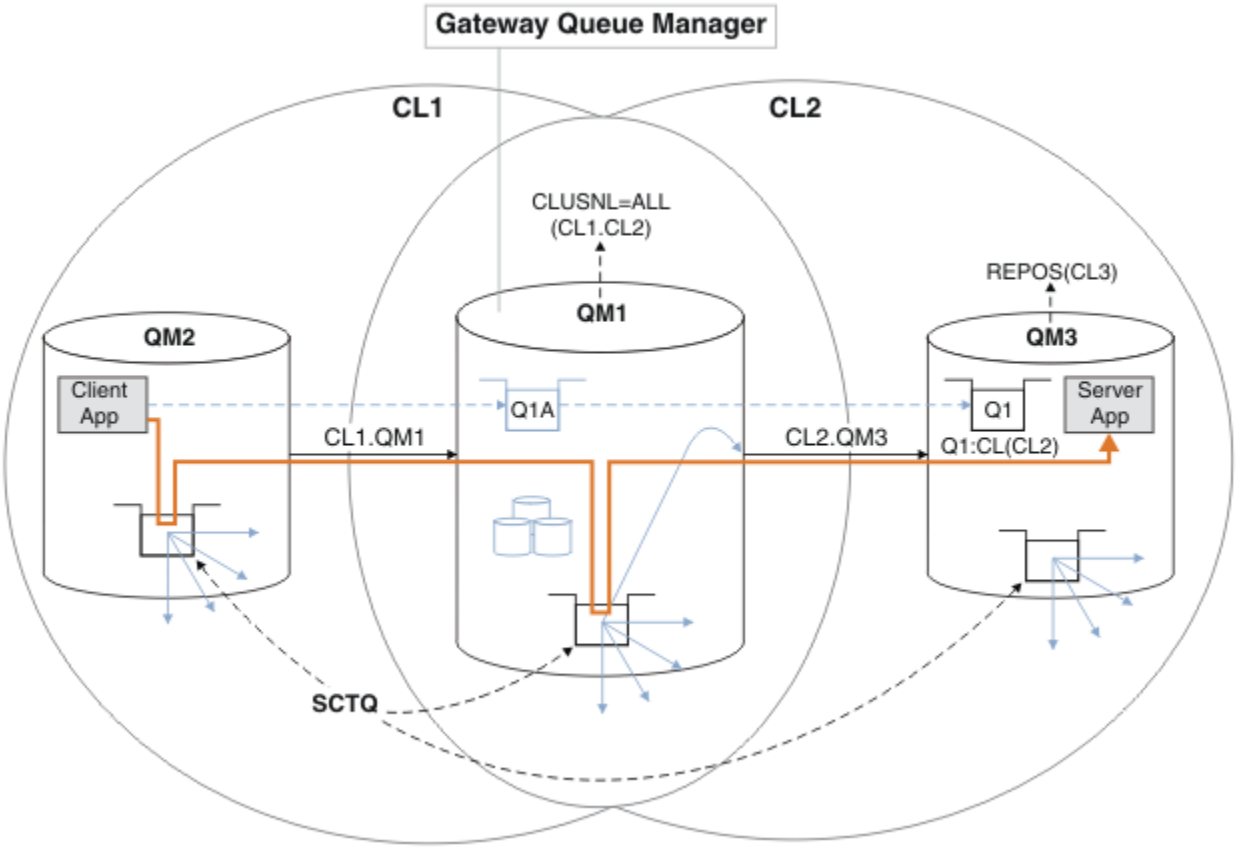
La arquitectura en estrella requiere que todos los mensajes entre clústeres pasen a través del gestor de cola de pasarela. El resultado es que todos los mensajes fluyen a través de la cola de transmisión de clúster individual en `QM1`, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.

Desde una perspectiva de rendimiento, una sola cola no es un problema. Una cola de transmisión común no suele representar un cuello de botella de rendimiento. El rendimiento de los mensajes en la pasarela viene determinado en gran medida por el rendimiento de los canales que conectados a ella. El rendimiento no suele verse afectado por el número de colas, o el número de mensajes de las colas que utilizan los canales.

Desde alguna otra perspectiva, el uso de una sola cola de transmisión para varias aplicaciones tiene inconvenientes:

- No se puede aislar el flujo de mensajes a un destino del flujo de mensajes a otro destino. No se puede separar el almacenamiento de mensajes antes de que se reenvíen, incluso si los destinos se encuentran en distintos clústeres en distintos gestores de colas.

  Si un destino de clúster deja de estar disponible, los mensajes para dicho destino se acumulan en la cola de transmisión única y finalmente los mensajes la acaban llenando. Cuando la cola de transmisión está llena, impide que los mensajes se coloquen en la cola de transmisión para cualquier destino de clúster.
- No es fácil supervisar la transferencia de mensajes a diferentes destinos de clúster. Todos los mensajes están en la cola de transmisión única. La visualización de la profundidad de la cola de transmisión le ofrece una pequeña indicación de si los mensajes se transfieren a todos los destinos.



Nota: Las flechas en la Figura 12 en la página 50 y en las figuras siguientes son de distintos tipos. Las flechas continuas representan flujos de mensajes. Las etiquetas de las flechas continuas son nombres de canales de mensajes. Las flechas sólidas grises son flujos de mensajes potenciales desde `SYSTEM.CLUSTER.TRANSMIT.QUEUE` a los canales de clúster emisor. Las líneas negras discontinuas conectan las etiquetas con sus destinos. Las flechas grises discontinuas son referencias; por ejemplo de una llamada `MQOPEN` por `Client App` a la definición de cola de alias de clúster `Q1A`.

Figura 12. Aplicación cliente-servidor desplegada en una arquitectura en estrella utilizando clústeres de IBM MQ

En Figura 12 en la página 50, los clientes de `Server App` abren la cola `Q1A`. Los mensajes se transfieren a `SYSTEM.CLUSTER.TRANSMIT.QUEUE` en `QM2`, se transfieren a `SYSTEM.CLUSTER.TRANSMIT.QUEUE` en `QM1`y luego se transfieren a `Q1` en `QM3`, donde son recibidos por la aplicación `Server App`..

El mensaje de `Client App` pasa a través de las colas de transmisión del clúster del sistema en `QM2` y `QM1`. En Figura 12 en la página 50, el objetivo es aislar el flujo de mensajes en el gestor de colas de pasarela desde la aplicación cliente, de modo que sus mensajes no se almacenen en `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. Puede aislar los flujos en cualquiera de los otros gestores de colas en clúster. También puede aislar los flujos en la otra dirección, de vuelta al cliente. Para que las descripciones de las soluciones sean breves, las descripciones sólo tienen en cuenta un solo flujo desde la aplicación cliente.

Soluciones para aislar el tráfico de mensajes de clúster en un gestor de colas de pasarela de clúster

Una manera de resolver el problema es utilizar alias de gestor de colas o definiciones de colas remotas para crear un puente entre clústeres. Cree una definición de cola remota de clúster, una cola de transmisión y un canal, para separar cada flujo de mensajes en el gestor de colas de pasarela; consulte el apartado Añadir una definición de cola remota para aislar los mensajes enviados desde un gestor de colas de pasarela.

A partir de la IBM WebSphere MQ 7.5, los gestores de colas de clúster no están limitadas a una sola cola de transmisión de clúster. Tiene dos opciones:

1. Definir colas de transmisión de clúster adicionales manualmente y definir qué canales de clúster emisor transfieren mensajes de cada cola de transmisión; consulte Añadir una cola de transmisión de clúster para aislar el tráfico de mensajes de clúster enviados desde un gestor de colas de pasarela.
2. Permitir que el gestor de colas cree y gestione colas de transmisión de clúster adicionales automáticamente. Define una cola de transmisión de clúster diferente para cada canal de clúster emisor; consulte el apartado Modificar el valor predeterminado para separar colas de transmisión de clúster para aislar el tráfico de mensajes.

Puede combinar manualmente colas de transmisión de clúster definidas para algunos canales de clúster emisor con el gestor de colas que gestiona el resto. La combinación de colas de transmisión es el enfoque que se ha utilizado en Añadir una cola de transmisión de clúster para aislar el tráfico de mensajes de clúster enviados desde un gestor de colas de pasarela. En esa solución, la mayoría de los mensajes entre clústeres utilizan `SYSTEM.CLUSTER.TRANSMIT.QUEUE` común. Una aplicación es crítica, y todos sus flujos de mensajes se aíslan de otros flujos utilizando una cola de transmisión de clúster definida manualmente.

La configuración en Añadir una cola de transmisión de clúster para aislar el tráfico de mensajes de clúster enviados desde un gestor de colas de pasarela es una configuración limitada. No separa el tráfico de mensajes dirigido a una cola de clúster en el mismo gestor de colas en el mismo clúster que otra cola de clúster. Puede separar el tráfico de mensajes a colas individuales utilizando definiciones de colas remotas que formen parte de colas distribuidas. Con clústeres, utilizando varias colas de transmisión de clúster, puede separar el tráfico de mensajes dirigido a diferentes canales de clúster emisor. Varias colas de clúster en el mismo clúster, en el mismo gestor de colas, comparten un canal de clúster emisor. Los mensajes para dichas colas se almacenan en la misma cola de transmisión antes de ser reenviados desde el gestor de colas de pasarela. En la configuración del apartado Añadir un clúster y una cola de transmisión de clúster para aislar el tráfico de mensajes de clúster enviados desde un gestor de colas de pasarela, la limitación se esquiva añadiendo otro clúster y convirtiendo al gestor de colas y la cola de clúster en miembros del nuevo clúster. El nuevo gestor de colas podría ser el único gestor de colas del clúster. Puede añadir más gestores de colas al clúster y utilizar el mismo clúster para aislar colas de clúster de dichos gestores de colas también.

Conceptos relacionados

“Control de accesos y varias colas de transmisión de clúster” en la página 29
Elija entre tres modalidades de comprobación cuando una aplicación transfiere mensajes a las colas de clúster remoto. Las modalidades se están comprobando de forma remota en la cola de clúster, se están comprobando localmente en `SYSTEM.CLUSTER.TRANSMIT.QUEUE`, o se están comprobando los perfiles locales para la cola de clúster o el gestor de colas de clúster.

Cómo trabajar con colas de transmisión de clúster y canales de clúster emisor

“Solapamiento de clústeres” en la página 36
El solapamiento de clústeres proporciona funciones administrativas adicionales. Utilice listas de nombres para reducir el número de mandatos necesarios para administrar clústeres que se solapan.

Tareas relacionadas

Autorización de transferencia de mensajes a colas de clústeres remotos

Añadir una definición de cola remota para aislar los mensajes enviados desde un gestor de colas de pasarela

Añadir una cola de transmisión de clúster para aislar el tráfico de mensajes de clúster enviados desde un gestor de colas de pasarela

Añadir un clúster y una cola de transmisión de clúster para aislar el tráfico de mensajes de clúster enviados desde un gestor de colas de pasarela

Modificar el valor predeterminado para separar colas de transmisión de clúster para aislar el tráfico de mensajes

Crear dos clústeres solapados con un gestor de cola de pasarela

Configurar vías de acceso de mensajes entre clústeres

Seguridad

Referencia relacionada

setmqaut

Agrupación en clúster: Planificación de cómo configurar las colas de transmisión de clúster

Este apartado le guiará a través de las opciones de las colas de transmisión de clúster. Puede configurar una cola predeterminada común, colas predeterminadas distintas o colas definidas manualmente.

Antes de empezar

Revise el apartado "Cómo seleccionar qué tipo de cola de transmisión de clúster se debe utilizar" en la página 55.

Acerca de esta tarea

Dispone de varias opciones que puede realizar cuando planifique cómo configurar un gestor de colas para seleccionar una cola de transmisión de clúster.

1. ¿Cuál es la cola de transmisión de clúster predeterminada para las transferencias de mensajes de clúster?
   a. Una cola de transmisión de clúster común, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.
   b. Colas de transmisión de clúster distintas. El gestor de colas gestiona las colas de transmisión de clúster distintas. Los crea como colas dinámicas permanentes de la cola modelo, `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE`. Crea una cola de transmisión de clúster para cada canal de clúster emisor que utiliza.
2. Para las colas de transmisión de clúster que decida crear manualmente, dispone de dos opciones más:
   a. Definir una cola de transmisión para cada canal de clúster emisor que decida configurar manualmente. En este caso, establezca el atributo de cola **CLCHNAME** de la cola de transmisión en el nombre de un canal de clúster emisor. Seleccione el canal de clúster emisor que debe transferir mensajes desde esta cola de transmisión.
   b. Combinar el tráfico de mensajes para un grupo de canales de clúster emisor en la misma cola de transmisión de clúster; consulte la Figura 13 en la página 53. En este caso, establezca el atributo de cola **CLCHNAME** de cada cola de transmisión común en un nombre de canal de clúster emisor genérico. Un nombre de canal de clúster emisor genérico es un filtro para agrupar nombres de canal de clúster emisor. Por ejemplo, `SALES.*` agrupa todos los canales de clúster emisor que tienen nombres que empiezan por `SALES.`. Puede colocar varios caracteres comodín en cualquier parte de la serie de filtro. El carácter comodín es un asterisco, "`*`". Representa de cero a cualquier número de caracteres.

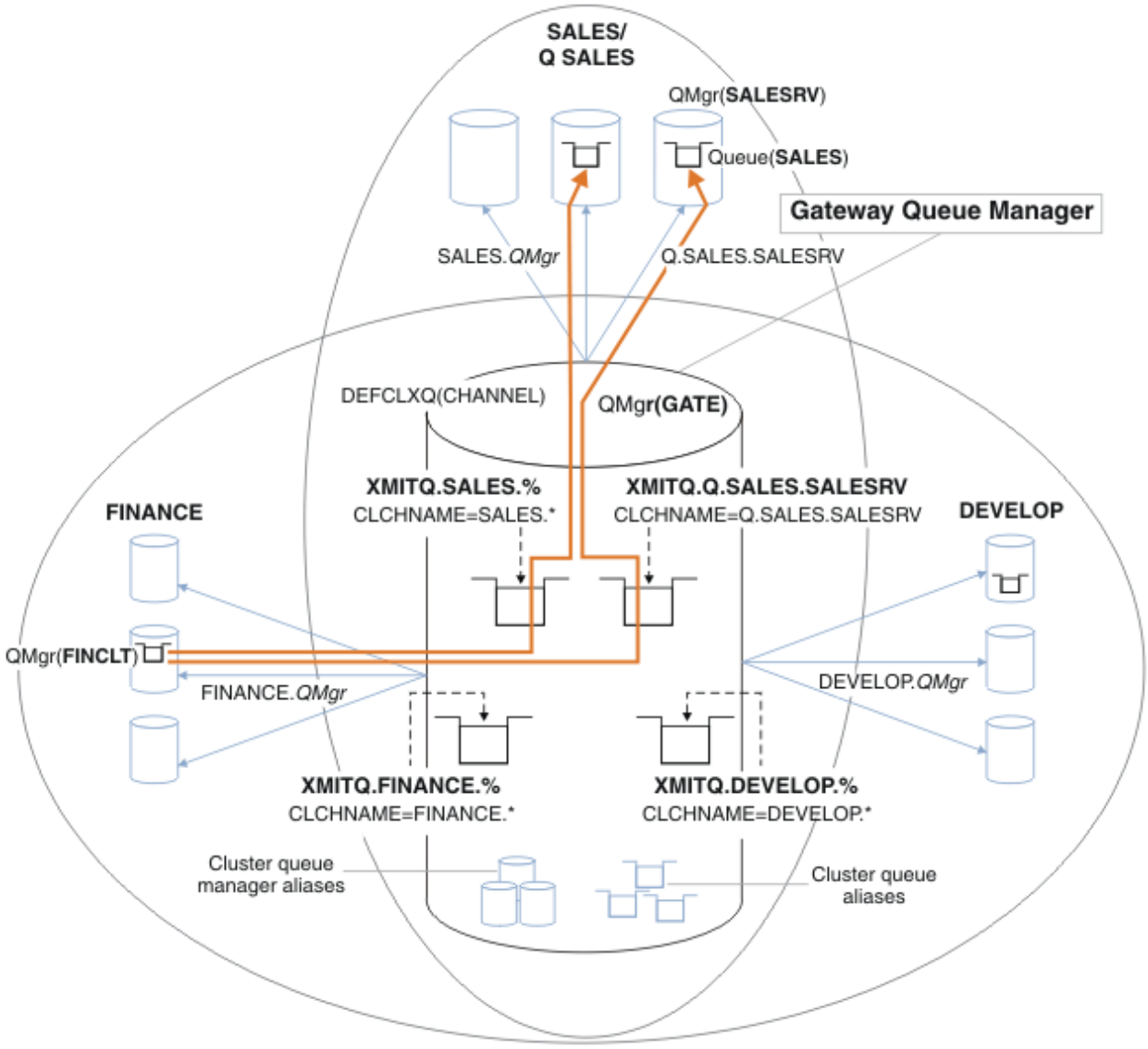


Figura 13. Ejemplo de colas de transmisión específicas para diferentes clústeres departamentales de IBM MQ

Procedimiento

1. Seleccione el tipo de cola de transmisión de clúster predeterminada que se debe utilizar.
   - Elija una sola cola de transmisión de clúster, o colas distintas para cada conexión de clúster.

   Deje el valor predeterminado o ejecute el mandato **MQSC**:

   ```
   ALTER QMGR DEFCLXQ(CHANNEL)
   ```

2. Aísle cualquier flujo de mensajes que no deba compartir una cola de transmisión de clúster con otros flujos.
   - Consulte “Agrupación en clúster: Ejemplo de configuración de varias colas de transmisión de clúster” en la página 56. En el ejemplo, la cola SALES, que debe estar aislada, es miembro del clúster de SALES , en SALESRV. Para aislar la cola SALES , cree un nuevo clúster `Q.SALES`, convierta el gestor de colas SALESRV en miembro y modifique la cola SALES para que pertenezca a `Q.SALES`.
   - Los gestores de colas que envían mensajes a SALES también deben ser miembros del nuevo clúster. Si utiliza un alias de cola de clúster y un gestor de colas de pasarela, como en el ejemplo, en muchos

casos puede limitar los cambios para convertir al gestor de colas de pasarela en miembro del nuevo clúster.

- No obstante, al separar flujos de la pasarela al destino no se separan los flujos a la pasarela del gestor de colas de origen. Pero a veces resulta ser suficiente para separar los flujos de la pasarela y no los flujos a la pasarela. Si no es suficiente, añada el gestor de colas de origen al nuevo clúster. Si desea que los mensajes viajen a través de la pasarela, mueva el alias de clúster al nuevo clúster y siga enviando mensajes al alias de clúster en la pasarela, y no directamente al gestor de colas de destino.

Siga estos pasos para aislar los flujos de mensajes:

a) Configure los destinos de los flujos de modo que cada cola de destino sea la única cola de un clúster específico, en ese gestor de colas.

b) Cree los canales de clúster emisor y de clúster receptor para los nuevos clústeres que haya creado siguiendo un convenio de denominación sistemático.

- Consulte "Agrupación en clúster: consideraciones especiales para los clústeres que se solapan" en la página 44.

c) Defina una cola de transmisión de clúster para cada destino aislado en cada gestor de colas que envía mensajes a la cola de destino.

- Un convenio de denominación para las colas de transmisión de clúster es utilizar el valor del atributo de nombre de canal de clúster, `CLCHNAME`, con el prefijo `XMITQ.`

3. Cree colas de transmisión de clúster para cumplir con los requisitos de supervisión o administración.

- Los requisitos de administración y supervisión típicos dan como resultado una cola de transmisión por clúster o una cola de transmisión por gestor de colas. Si sigue el convenio de denominación para los canales de clúster, *`ClusterName. QueueManagerName`*, es fácil crear nombres de canal genéricos que seleccionan un clúster de gestores de colas, o todos los clústeres de los que un gestor de colas es miembro; consulte "Agrupación en clúster: Ejemplo de configuración de varias colas de transmisión de clúster" en la página 56.
- Amplíe el convenio de denominación para colas de transmisión de clúster para permitir nombres de canal genéricos, sustituyendo el símbolo de asterisco por un signo de porcentaje. Por ejemplo,

```
DEFINE QLOCAL(XMITQ.SALES.%) USAGE(XMITQ) CLCHNAME(SALES.*)
```

Conceptos relacionados

Cómo trabajar con colas de transmisión de clúster y canales de clúster emisor

"Control de accesos y varias colas de transmisión de clúster" en la página 29

Elija entre tres modalidades de comprobación cuando una aplicación transfiere mensajes a las colas de clúster remoto. Las modalidades se están comprobando de forma remota en la cola de clúster, se están comprobando localmente en `SYSTEM.CLUSTER.TRANSMIT.QUEUE`, o se están comprobando los perfiles locales para la cola de clúster o el gestor de colas de clúster.

"Solapamiento de clústeres" en la página 36

El solapamiento de clústeres proporciona funciones administrativas adicionales. Utilice listas de nombres para reducir el número de mandatos necesarios para administrar clústeres que se solapan.

Tareas relacionadas

Añadir una definición de cola remota para aislar los mensajes enviados desde un gestor de colas de pasarela

Añadir una cola de transmisión de clúster para aislar el tráfico de mensajes de clúster enviados desde un gestor de colas de pasarela

Añadir un clúster y una cola de transmisión de clúster para aislar el tráfico de mensajes de clúster enviados desde un gestor de colas de pasarela

Modificar el valor predeterminado para separar colas de transmisión de clúster para aislar el tráfico de mensajes

Crear dos clústeres solapados con un gestor de cola de pasarela

Configurar vías de acceso de mensajes entre clústeres

Cómo seleccionar qué tipo de cola de transmisión de clúster se debe utilizar
Cómo elegir entre diferentes opciones de configuración de cola de transmisión de clúster.

Puede elegir qué cola de transmisión de clúster está asociada con un canal de clúster emisor.

1. Puede tener todos los canales de clúster emisor asociados con la única cola de transmisión de clúster predeterminada, SYSTEM.CLUSTER.TRANSMIT.QUEUE; esta opción es la predeterminada.
2. Puede establecer que todos los canales de clúster emisor se asocien automáticamente con una cola de transmisión de clúster distinta. Las colas las crea el gestor de colas de la cola modelo SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE y se denomina SYSTEM.CLUSTER.TRANSMIT.*ChannelName*. Los canales utilizarán su cola de transmisión de clúster con nombre exclusivo si el atributo de gestor de colas **DEFCLXQ** se establece en CANAL.
3. Puede establecer que una sola cola de transmisión de clúster preste servicio a canales de clúster emisor específicos. Seleccione esta opción creando una cola de transmisión y estableciendo su atributo **CLCHNAME** en el nombre del canal de clúster emisor.
4. Puede seleccionar grupos de canales de clúster emisor para que les preste servicio una sola cola de transmisión de clúster. Seleccione esta opción creando una cola de transmisión y estableciendo su atributo **CLCHNAME** en un nombre de canal genérico como, por ejemplo, *ClusterName*.*. Si asigna un nombre a los canales de clúster siguiendo los convenios de denominación de "Agrupación en clúster: consideraciones especiales para los clústeres que se solapan" en la página 44, este nombre selecciona todos los canales de clúster conectados a los gestores de colas del clúster *ClusterName*.

Puede combinar cualquiera de las opciones de cola de transmisión de clúster predeterminada para algunos canales de clúster emisor con cualquier número de configuraciones de cola de transmisión de clúster específicas y genéricas.

Procedimientos recomendados

En la mayoría de los casos, para las instalaciones de IBM MQ existentes, la configuración predeterminada es la mejor opción. Un gestor de colas de clúster almacena mensajes de clúster en una sola cola de transmisión de clúster, SYSTEM.CLUSTER.TRANSMIT.QUEUE. Tiene la opción de cambiar el valor predeterminado para almacenar mensajes para gestores de colas diferentes y clústeres diferentes en distintas colas de transmisión, o de definir sus propias colas de transmisión.

En la mayoría de los casos, para las instalaciones de IBM MQ nuevas, la configuración predeterminada es la mejor opción. El proceso de cambio de la configuración predeterminada a la alternativa predeterminada de tener una cola de transmisión para cada canal de clúster emisor es automático. El cambio de vuelta a la configuración predeterminada también es automático. La elección de una u otra no es crucial, se puede invertir.

La razón para elegir una configuración diferente tiene que ver más con la administración y la gestión que con la funcionalidad o el rendimiento. Con un par de excepciones, configurar varias colas de transmisión de clúster no beneficia al comportamiento del gestor de colas. Tiene como resultado más colas y requiere que se modifiquen los procedimientos de la supervisión y gestión que ya se han configurado y que hacen referencia a la cola de transmisión única. Es por ello que, en conjunto, dejar la configuración predeterminada es la mejor opción, a menos que haya firmes razones de administración o gestión para elegir una opción diferente.

Las excepciones se refieren a lo que sucede si aumenta el número de mensajes almacenados en SYSTEM.CLUSTER.TRANSMIT.QUEUE. Si toma todas las medidas para separar los mensajes para un destino de los mensajes para otro destino, entonces los problemas de canal y entrega de un destino no deberían afectar a la entrega a otro destino. Sin embargo, el número de mensajes almacenados en SYSTEM.CLUSTER.TRANSMIT.QUEUE puede aumentar debido a que no entrega mensajes lo suficientemente rápido como para un destino. El número de mensajes en SYSTEM.CLUSTER.TRANSMIT.QUEUE para un destino puede afectar a la entrega de mensajes a otros destinos.

Para evitar los problemas debidos al llenado de una cola de transmisión individual, procure crear suficiente capacidad en la configuración. A continuación, si un destino falla y se empieza a acumular un retardo de mensajes, tendrá tiempo para arreglar el problema.

Si los mensajes se direccionan a través de un gestor de colas concentrador, como por ejemplo una pasarela de clúster, comparten una cola de transmisión común, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. Si el número de mensajes almacenados en `SYSTEM.CLUSTER.TRANSMIT.QUEUE` en el gestor de colas de pasarela alcanza su profundidad máxima, el gestor de colas empieza a rechazar mensajes nuevos para la cola de transmisión hasta que se reduce la profundidad. La congestión afecta a los mensajes para todos los destinos que se direccionan a través de la pasarela. Los mensajes de las colas de transmisión de otros gestores de colas que envían mensajes a la pasarela. El problema se manifiesta en mensajes grabados en los registros de errores del gestor de colas, una reducción del rendimiento de los mensajes e intervalos de tiempo superiores entre el envío de un mensaje y la hora de llegada del mensaje a su destino.

El efecto de la congestión en una sola cola de transmisión puede ser evidente, incluso antes de que esté llena. Si tiene un tráfico de mensajes mixto, con algunos mensajes no permanente de gran tamaño y algunos mensajes pequeños, el tiempo para entregar mensajes pequeños aumenta a medida que la cola de transmisión se llena. El retraso se debe a la grabación en disco de mensajes no permanentes de gran tamaño que normalmente no se graban en el disco. Si tiene flujos de mensajes para los que el tiempo resulte crucial, que comparten una cola de transmisión de clúster con otros flujos de mensajes mixtos, puede ser recomendable configurar una ruta de mensajes especial para aislarla de otros flujos de mensajes; consulte Añadir un clúster y una cola de transmisión de clúster para aislar el tráfico de mensajes de clúster enviados desde un gestor de colas de pasarela.

Las otras razones para configurar colas de transmisión de clúster separadas son cumplir los requisitos de administración o simplificar la supervisión de mensajes que se envían a destinos de clúster diferentes. Por ejemplo, es posible que tenga que demostrar que los mensajes para un destino nunca comparten una cola de transmisión con mensajes para otro destino.

Cambie el atributo de gestor de colas **DEFCLXQ** que controla la cola de transmisión de clúster predeterminada, para crear diferentes colas de transmisión de clúster para cada canal de clúster emisor. Varios destinos pueden compartir un canal de clúster emisor, por lo que debe planificar sus clústeres para cumplir este objetivo completamente. Aplique el método Añadir un clúster y una cola de transmisión de clúster para aislar el tráfico de mensajes de clúster enviados desde un gestor de colas de pasarela de forma sistemática a todas las colas del clúster. El resultado que se intenta conseguir es que ningún destino de clúster comparta un canal de clúster emisor con otro destino de clúster. Como consecuencia, ningún mensaje para un destino de clúster comparte su cola de transmisión de clúster con un mensaje para otro destino.

La creación de una cola de transmisión de clúster distinta para algún flujo de mensajes específico facilita la supervisión del flujo de mensajes a dicho destino. Para utilizar una nueva cola de transmisión de clúster, defina la cola, asóciela con un canal de clúster emisor y detenga e inicie el canal. El cambio no tiene que ser necesariamente permanente. Puede aislar un flujo de mensajes durante un tiempo, para supervisar la cola de transmisión, y luego volver a utilizar la cola de transmisión predeterminada.

Tareas relacionadas

Agrupación en clúster: Ejemplo de configuración de varias colas de transmisión de clúster
En esta tarea seguirá los pasos para asignar varias colas de transmisión de clúster a tres clústeres solapados. Los requisitos son separar los flujos de mensajes destinados a una cola de clúster de todos los demás flujos de mensajes y almacenar los mensajes destinados a clústeres diferentes en colas de transmisión de clúster diferentes.

Agrupación en clúster: conmutación de colas de transmisión de clúster
Planifique cómo entrarán en vigor los cambios de las colas de transmisión de clúster de un gestor de colas de producción existente.

Agrupación en clúster: Ejemplo de configuración de varias colas de transmisión de clúster
En esta tarea seguirá los pasos para asignar varias colas de transmisión de clúster a tres clústeres solapados. Los requisitos son separar los flujos de mensajes destinados a una cola de clúster de todos los demás flujos de mensajes y almacenar los mensajes destinados a clústeres diferentes en colas de transmisión de clúster diferentes.

Acerca de esta tarea

Los pasos de esta tarea muestran cómo aplicar el procedimiento descrito en "Agrupación en clúster: Planificación de cómo configurar las colas de transmisión de clúster" en la página 52 y llegar a la configuración mostrada en Figura 14 en la página 57. El ejemplo incluye tres clústeres solapados, con un gestor de colas de pasarela, que está configurado con colas de transmisión de clúster separadas. Los mandatos MQSC para definir los clústeres se describen en "Creación de clústeres de ejemplo" en la página 59.

Para el ejemplo, existen dos requisitos. Un requisito es separar el flujo de mensajes que va desde el gestor de colas de pasarela a la aplicación de ventas que registra las ventas. El segundo requisito es consultar cuántos mensajes están a la espera de ser enviados a los diferentes departamentos en un momento determinado cualquiera. Los clústeres SALES, FINANCE y DEVELOP ya están definidos. Los mensajes de clúster se reenvían actualmente desde SYSTEM.CLUSTER.TRANSMIT.QUEUE.

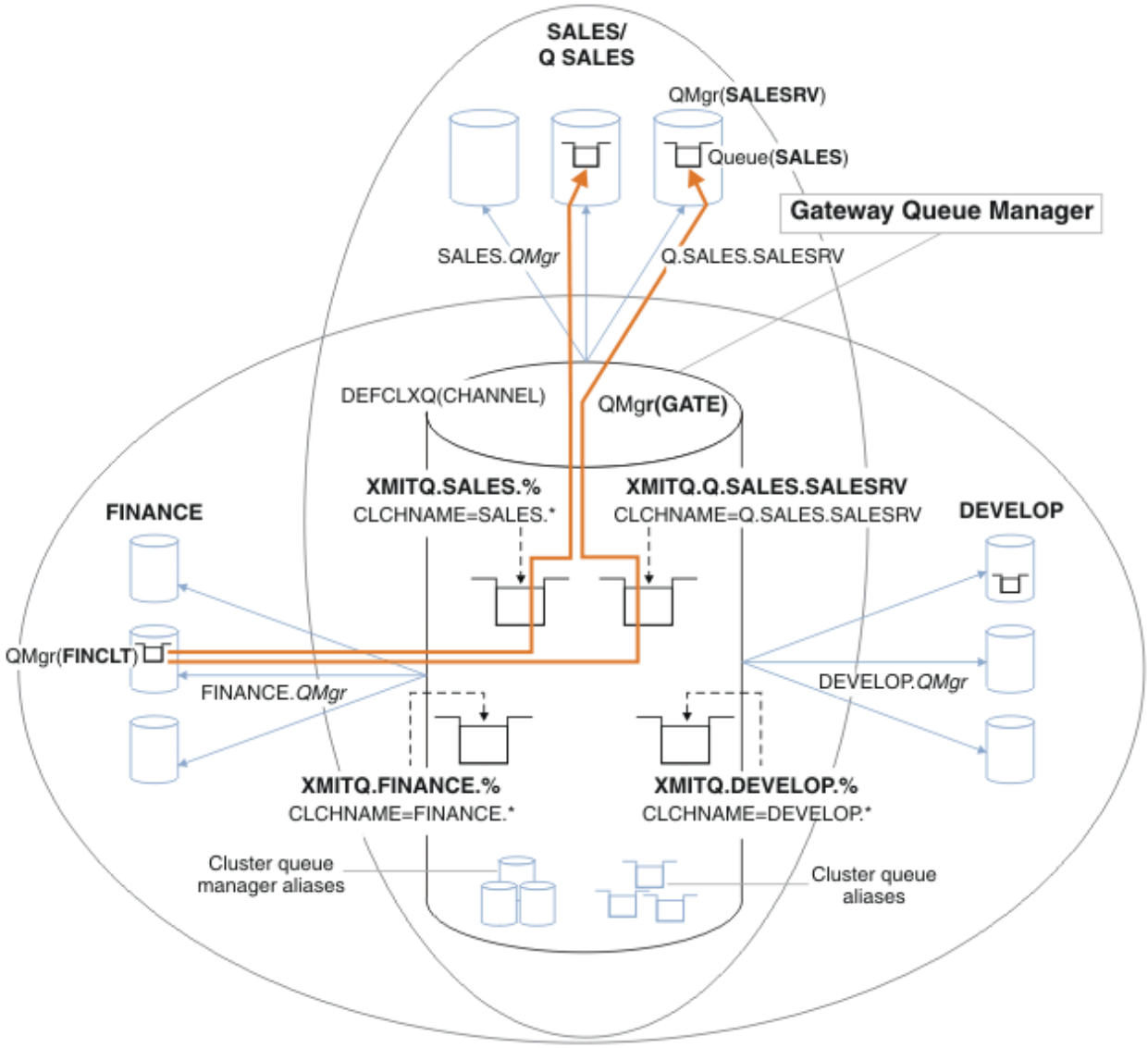


Figura 14. Ejemplo de colas de transmisión específicas para diferentes clústeres departamentales de IBM MQ

Los pasos para modificar los clústeres son los siguientes. Para ver las definiciones, consulte Cambios para aislar la cola de ventas en un nuevo clúster y separar las colas de transmisión de clúster de pasarela.

Procedimiento

1. El primer paso de configuración es " Seleccione el tipo de cola de transmisión de clúster predeterminada que se debe utilizar ".

   La decisión es crear colas de transmisión de clúster predeterminadas diferentes ejecutando el siguiente mandato **MQSC** en el gestor de colas GATE.

   ```
   ALTER QMGR DEFCLXQ(CHANNEL)
   ```

   No existe ninguna razón importante para elegir este valor predeterminado, puesto que la intención es definir manualmente colas de transmisión de clúster. La elección tiene un valor de diagnóstico débil. Si una definición manual se realiza incorrectamente y un mensaje fluye por una cola de transmisión de clúster predeterminada, se pone de manifiesto en la creación de una cola de transmisión de clúster dinámica permanente.
2. El segundo paso de configuración es " Aísle cualquier flujo de mensajes que no deba compartir una cola de transmisión de clúster con otros flujos ".

   En este caso, la aplicación de ventas que recibe mensajes de la cola SALES en SALESRV requiere aislamiento. Sólo es necesario el aislamiento de los mensajes procedentes del gestor de colas de pasarela. Los tres subpasos son:

   a) " Configure los destinos de los flujos de modo que cada cola de destino sea la única cola de un clúster específico, en ese gestor de colas ".

      El ejemplo requiere añadir el gestor de colas SALESRV a un nuevo clúster dentro del departamento de ventas. Si tiene pocas colas que requieren aislamiento, puede decidir la creación de un clúster específico para la cola SALES. Un posible convenio de denominación para el nombre de clúster es el nombre de dichos clústeres, `Q.` *QueueName*, por ejemplo `Q.SALES`. Un método alternativo, que puede ser más práctico si tiene un gran número de colas para aislar, es crear clústeres de colas aisladas donde y cuando sea necesario. Los nombres de clústeres pueden ser `QUEUES.` *n*.

      En el ejemplo, el nuevo clúster se llama `Q.SALES`. Para añadir el nuevo clúster, consulte las definiciones en Cambios para aislar la cola de ventas en un nuevo clúster y separar las colas de transmisión de clúster de pasarela. El resumen de los cambios de definición es el siguiente:

      i) Añada `Q.SALES` a la lista de nombres de clústeres en los gestores de colas de repositorio. La lista de nombres es referida en el parámetro del gestor de colas **REPOSNL**.

      ii) Añada `Q.SALES` a la lista de nombres de clústeres en el gestor de colas de pasarela. La lista de nombres es utilizada en todas las definiciones de alias de cola de clúster y definiciones de alias de gestor de colas de clúster en el gestor de colas de pasarela.

      iii) Cree una lista de nombres en el gestor de colas SALESRV, para los clústeres en los que es miembro y cambie la pertenencia al clúster de la cola SALES:

      ```
      DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES) REPLACE
      ALTER QLOCAL(SALES) CLUSTER(' ') CLUSNL(SALESRV.CLUSTERS)
      ```

      La cola SALES es miembro de ambos clústeres, sólo para la transición. Una vez que se está ejecutando la nueva configuración, se elimina la cola SALES del clúster SALES; consulte Figura 15 en la página 62.

   b) " Cree los canales de clúster emisor y de clúster receptor para los nuevos clústeres que haya creado siguiendo un convenio de denominación sistemático ".

      i) Añadir el canal de clúster receptor `Q.SALES.` *RepositoryQMgr* a cada uno de los gestores de colas del repositorio

      ii) Añada el canal de clúster emisor `Q.SALES.` *OtherRepositoryQMgr* a cada uno de los gestores de colas de repositorio para conectarse al otro gestor de repositorios. Inicie estos canales.

      iii) Añada los canales de clúster receptor `Q.SALES.SALESRV` y `Q.SALES.GATE` a cualquiera de los gestores de colas de repositorio que se están ejecutando.

iv) Añada los canales de clúster emisor Q.SALES.SALESRV y Q.SALES.GATE a los gestores de colas SALESRV y GATE. Conecte el canal de clúster emisor al gestor de colas de repositorio en el que ha creado los canales de clúster receptor.

c) " Defina una cola de transmisión de clúster para cada destino aislado en cada gestor de colas que envía mensajes a la cola de destino ".

En el gestor de colas de pasarela, defina la cola de transmisión de clúster XMITQ.Q.SALES.SALESRV para el canal de clúster emisor de Q.SALES.SALESRV:

```
DEFINE QLOCAL(XMITQ.Q.SALES.SALESRV) USAGE(XMITQ) CLCHNAME(Q.SALES.SALESRV) REPLACE
```

3. El tercer paso de configuración es " Cree colas de transmisión de clúster para cumplir con los requisitos de supervisión o administración ".

   En el gestor de colas de pasarela defina las colas de transmisión de clúster:

```
DEFINE QLOCAL(XMITQ.SALES)  USAGE(XMITQ) CLCHNAME(SALES.*)  REPLACE
DEFINE QLOCAL(XMITQ.DEVELOP) USAGE(XMITQ) CLCHNAME(DEVELOP.*) REPLACE
DEFINE QLOCAL(XMITQ.FINANCE) USAGE(XMITQ) CLCHNAME(SALES.*)  REPLACE
```

Qué hacer a continuación

Conmute a la nueva configuración en el gestor de colas de pasarela.

La conmutación se desencadena al iniciar los nuevos canales y reiniciar los canales que ahora están asociados a colas de transmisión diferentes. Como alternativa, puede detener e iniciar el gestor de colas de pasarela.

1. Detenga los canales siguientes en el gestor de colas de pasarela:

```
SALES. Qmgr
DEVELOP. Qmgr
FINANCE. Qmgr
```

2. Inicie los canales siguientes en el gestor de colas de pasarela:

```
SALES. Qmgr
DEVELOP. Qmgr
FINANCE. Qmgr
Q.SALES.SAVESRV
```

Cuando el conmutador esté completo, elimine la cola SALES del clúster SALES; consulte Figura 15 en la página 62.

Conceptos relacionados

Cómo seleccionar qué tipo de cola de transmisión de clúster se debe utilizar
Cómo elegir entre diferentes opciones de configuración de cola de transmisión de clúster.

Tareas relacionadas

Agrupación en clúster: conmutación de colas de transmisión de clúster
Planifique cómo entrarán en vigor los cambios de las colas de transmisión de clúster de un gestor de colas de producción existente.

Creación de clústeres de ejemplo
Las definiciones e instrucciones para crear el clúster de ejemplo y modificarlo para aislar la cola de SALES y separar los mensajes en el gestor de colas de pasarela.

Acerca de esta tarea

Los mandatos **MQSC** completos para crear los clústeres FINANCE, SALESy Q.SALES se proporcionan en Definiciones para los clústeres básicos, Cambios para aislar la cola de ventas en un nuevo clúster y

separar las colas de transmisión de clúster de pasarelay Eliminar la cola de ventas en el gestor de colas SALESRV del clúster de ventas. El clúster DEVELOP se omite de las definiciones para hacerlas más cortas.

Procedimiento

1. Cree los clústeres SALES y FINANCE y el gestor de colas de pasarela.
   a) Cree los gestores de colas.

      Ejecute el mandato: `crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE` *QmgrName* para cada uno de los nombres de gestor de colas en Tabla 4 en la página 60.

Tabla 4. Nombres de gestores de colas y números de puerto		
Descripción	**Nombre del gestor de colas**	**Número de puerto**
Repositorio Finance	FINR1	1414
Repositorio Finance	FINR2	1415
Cliente de Finance	FINCLT	1418
Repositorio Sales	SALER1	1416
Repositorio Sales	SALER2	1417
Servidor de Sales	SALESRV	1419
Pasarela	GATE	1420

   b) Inicie todos los gestores de colas.

      Ejecute el mandato: `strmqm` *QmgrName* para cada uno de los nombres de gestor de colas en Tabla 4 en la página 60.

   c) Cree las definiciones para cada gestor de colas

      Ejecute el mandato: `runmqsc` *QmgrName* < *filename* donde los archivos se listan en Definiciones para los clústeres básicos, y el nombre de archivo coincide con el nombre del gestor de colas.

      Definiciones para los clústeres básicos
      finr1.txt

```
DEFINE LISTENER(1414) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1414) REPLACE
START LISTENER(1414)
ALTER QMGR REPOS(FINANCE)
DEFINE CHANNEL(FINANCE.FINR2) CHLTYPE(CLUSSDR) CONNAME('localhost(1415)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
```

      finr2.txt

```
DEFINE LISTENER(1415) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1415) REPLACE
START LISTENER(1415)
ALTER QMGR REPOS(FINANCE)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINR2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1415)')
CLUSTER(FINANCE) REPLACE
```

      finclt.txt

```
DEFINE LISTENER(1418) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1418) REPLACE
START LISTENER(1418)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR)  CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINCLT) CHLTYPE(CLUSRCVR) CONNAME('localhost(1418)')
```

```
CLUSTER(FINANCE) REPLACE
DEFINE QMODEL(SYSTEM.SAMPLE.REPLY) REPLACE
```

saler1.txt

```
DEFINE LISTENER(1416) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1416) REPLACE
START LISTENER(1416)
ALTER QMGR REPOS(SALES)
DEFINE CHANNEL(SALES.SALER2) CHLTYPE(CLUSSDR) CONNAME('localhost(1417)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
```

saler2.txt

```
DEFINE LISTENER(1417) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1417) REPLACE
START LISTENER(1417)
ALTER QMGR REPOS(SALES)
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALER2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1417)')
CLUSTER(SALES) REPLACE
```

salesrv.txt

```
DEFINE LISTENER(1419) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1419) REPLACE
START LISTENER(1419)
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALESRV) CHLTYPE(CLUSRCVR) CONNAME('localhost(1419)')
CLUSTER(SALES) REPLACE
DEFINE QLOCAL(SALES) CLUSTER(SALES) TRIGGER INITQ(SYSTEM.DEFAULT.INITIATION.QUEUE)
PROCESS(ECHO) REPLACE
DEFINE PROCESS(ECHO) APPLICID(AMQSECH) REPLACE
```

gate.txt

```
DEFINE LISTENER(1420) TRPTYPE(TCP) IPADDR(LOCALHOST) CONTROL(QMGR) PORT(1420) REPLACE
START LISTENER(1420)
DEFINE NAMELIST(ALL) NAMES(SALES, FINANCE)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('LOCALHOST(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.GATE) CHLTYPE(CLUSRCVR) CONNAME('LOCALHOST(1420)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('LOCALHOST(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.GATE)  CHLTYPE(CLUSRCVR) CONNAME('LOCALHOST(1420)')
CLUSTER(SALES) REPLACE
DEFINE QALIAS(A.SALES) CLUSNL(ALL) TARGET(SALES) TARGTYPE(QUEUE) DEFBIND(NOTFIXED)
REPLACE
DEFINE QREMOTE(FINCLT) RNAME(' ') RQMNAME(FINCLT) CLUSNL(ALL) REPLACE
DEFINE QREMOTE(SALESRV) RNAME(' ') RQMNAME(SALESRV) CLUSNL(ALL) REPLACE
```

2. Verifique la configuración ejecutando el programa de solicitud de ejemplo.
   a) Iniciar el programa de supervisor desencadenante en el gestor de colas de SALESRV

      En Windows, abra una ventana de mandatos y ejecute el mandato `runmqtrm -m SALESRV`
   b) Ejecute el programa de solicitud de ejemplo y envíe una solicitud.

      En Windows, abra una ventana de mandatos y ejecute el mandato `amqsreq A.SALES FINCLT`

      El mensaje de solicitud se reproduce en la pantalla y después de 15 segundos el programa de ejemplo finaliza.
3. Cree las definiciones para aislar la cola SALES en el clúster `Q.SALES` y separe los mensajes de clúster para el clúster SALES y FINANCE en el gestor de colas de pasarela.

   Ejecute el mandato: `runmqsc` *QmgrName* < *filename* donde los archivos se listan en la lista siguiente y el nombre de archivo casi coincide con el nombre del gestor de colas.

Cambios para aislar la cola SALES en un nuevo clúster y separar las colas de transmisión de clúster de pasarela
chgsaler1.txt

```
DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES)
ALTER QMGR REPOS(' ') REPOSNL(CLUSTERS)
DEFINE CHANNEL(Q.SALES.SALER2) CHLTYPE(CLUSSDR) CONNAME('localhost(1417)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
```

chgsaler2.txt

```
DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES)
ALTER QMGR REPOS(' ') REPOSNL(CLUSTERS)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SALER2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1417)')
CLUSTER(Q.SALES) REPLACE
```

chgsalesrv.txt

```
DEFINE NAMELIST (CLUSTERS) NAMES(SALES, Q.SALES)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SAVESRV) CHLTYPE(CLUSRCVR) CONNAME('localhost(1419)')
CLUSTER(Q.SALES) REPLACE
ALTER QLOCAL (SALES) CLUSTER(' ') CLUSNL(CLUSTERS)
```

chggate.txt

```
ALTER NAMELIST(ALL) NAMES(SALES, FINANCE, Q.SALES)
ALTER QMGR DEFCLXQ(CHANNEL)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR)  CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.GATE)  CHLTYPE(CLUSRCVR) CONNAME('localhost(1420)')
CLUSTER(Q.SALES) REPLACE
DEFINE QLOCAL (XMITQ.Q.SALES.SALESRV) USAGE(XMITQ) CLCHNAME(Q.SALES.SALESRV) REPLACE
DEFINE QLOCAL (XMITQ.SALES)     USAGE(XMITQ) CLCHNAME(SALES.*)  REPLACE
DEFINE QLOCAL (XMITQ.FINANCE)    USAGE(XMITQ) CLCHNAME(FINANCE.*) REPLACE
```

4. Elimine la cola SALES del clúster SALES.

   Ejecute el mandato **MQSC** mostrado en la Figura 15 en la página 62:

   ```
   ALTER QLOCAL(SALES) CLUSTER('Q.SALES') CLUSNL(' ')
   ```

   Figura 15. Elimine la cola SALES en el gestor de colas SALESRV del clúster SALES
5. Conmute los canales a las nuevas colas de transmisión.

   El requisito es detener e iniciar todos los canales que el gestor de colas de GATE está utilizando. Para hacerlo con el menor número de mandatos, detenga e inicie el gestor de colas

   ```
   endmqm -i GATE
   strmqm GATE
   ```

Qué hacer a continuación

1. Ejecute de nuevo el programa de solicitud de ejemplo para verificar que la nueva configuración funciona; vea el paso “2” en la página 61
2. Supervise los mensajes que fluyen a través de todas las colas de transmisión de clúster en el gestor de colas de GATE:

   a. Modifique la definición de cada cola de transmisión de clúster para activar la supervisión de colas.

```
ALTER QLOCAL(SYSTEM.CLUSTER.TRANSMIT.
name) STATQ(ON)
```

b. Compruebe que la supervisión de estadísticas del gestor de colas es OFF, para minimizar la salida y establecer el intervalo de supervisión en un valor inferior para realizar varias pruebas convenientemente.

```
ALTER QMGR STATINT(60) STATCHL(OFF) STATQ(OFF) STATMQI(OFF) STATACLS(OFF)
```

c. Reinicie el gestor de colas de GATE.

d. Ejecute el programa de solicitud de ejemplo unas cuantas veces para verificar que un número igual de mensajes fluyan a través de SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV y SYSTEM.CLUSTER.TRANSMIT.QUEUE. Las solicitudes fluyen a través de SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV y las respuestas a través de SYSTEM.CLUSTER.TRANSMIT.QUEUE.

```
amqsmon -m GATE -t statistics
```

e. Los resultados a lo largo de un par de intervalos son los siguientes:

```
C:\Documents and Settings\Admin>amqsmon -m GATE -t statistics
MonitoringType: QueueStatistics
QueueManager: 'GATE'
IntervalStartDate: '2012-02-27'
IntervalStartTime: '14.59.20'
IntervalEndDate: '2012-02-27'
IntervalEndTime: '15.00.20'
CommandLevel: 700
ObjectCount: 2
QueueStatistics: 0
QueueName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'
CreateDate: '2012-02-24'
CreateTime: '15.58.15'
...
Put1Count: [0, 0]
Put1FailCount: 0
PutBytes: [435, 0]
GetCount: [1, 0]
GetBytes: [435, 0]
...
QueueStatistics: 1
QueueName: 'SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV'
CreateDate: '2012-02-24'
CreateTime: '16.37.43'
...
PutCount: [1, 0]
PutFailCount: 0
Put1Count: [0, 0]
Put1FailCount: 0
PutBytes: [435, 0]
GetCount: [1, 0]
GetBytes: [435, 0]
...
MonitoringType: QueueStatistics
QueueManager: 'GATE'
```

```
IntervalStartDate: '2012-02-27'
IntervalStartTime: '15.00.20'
IntervalEndDate: '2012-02-27'
IntervalEndTime: '15.01.20'
CommandLevel: 700
ObjectCount: 2
QueueStatistics: 0
QueueName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'
CreateDate: '2012-02-24'
CreateTime: '15.58.15'
...
PutCount: [2, 0]
PutFailCount: 0
Put1Count: [0, 0]
Put1FailCount: 0
PutBytes: [863, 0]
GetCount: [2, 0]
GetBytes: [863, 0]
...
QueueStatistics: 1
QueueName: 'SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV'
CreateDate: '2012-02-24'
CreateTime: '16.37.43'
...
PutCount: [2, 0]
PutFailCount: 0
Put1Count: [0, 0]
Put1FailCount: 0
PutBytes: [863, 0]
GetCount: [2, 0]
GetBytes: [863, 0]
...
2 Records Processed.
```

Se ha enviado un mensaje de solicitud y respuesta en el primer intervalo y dos en el segundo. Puede inferir que los mensajes de solicitud se han colocado en `SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV`y los mensajes de respuesta en `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.

Agrupación en clúster: conmutación de colas de transmisión de clúster
Planifique cómo entrarán en vigor los cambios de las colas de transmisión de clúster de un gestor de colas de producción existente.

Antes de empezar

Si reduce el número de mensajes que el proceso de conmutación debe transferir a la nueva cola de transmisión, la conmutación se completa más rápidamente. Consulte Cómo funciona el proceso de conmutación de un canal de clúster emisor a una cola de transmisión diferente para conocer las razones para intentar vaciar la cola de transmisión antes de continuar.

Acerca de esta tarea

Puede elegir entre dos modos de hacer que los cambios de las colas de transmisión de clúster entren en vigor.

1. Dejar que el gestor de colas realice los cambios automáticamente. Este es el valor predeterminado. El gestor de colas conmuta los canales de clúster emisor con cambios de colas de transmisión pendientes cuando se inicia un canal de clúster emisor a continuación.

2. Realizar los cambios manualmente. Puede realizar los cambios en un canal de clúster emisor cuando se detenga. Puede conmutarlo de una cola de transmisión de clúster a otro antes de que el canal de clúster emisor se inicie.

¿Qué factores tiene en cuenta al decidir cuál de las dos opciones debe elegir y cómo gestiona la conmutación?

Procedimiento

- Opción 1: Dejar que el gestor de colas realice los cambios automáticamente; consulte “Conmutación de canales de clúster emisor activos a otro conjunto de colas de transmisión de clúster” en la página 66.

  Elija esta opción si desea que el gestor de colas realice la conmutación automáticamente.

  Una forma alternativa de describir esta opción es decir que el gestor de colas conmuta un canal de clúster emisor sin que sea necesario forzar que el canal se detenga. Tiene la opción de forzar la detención del canal y, a continuación, iniciar el canal, para que la conmutación se produzca antes. El conmutador se inicia cuando se inicia el canal y se ejecuta mientras se ejecuta el canal, que es diferente a la opción 2. En la opción 2, el conmutador tiene lugar cuando se detiene el canal.

  Si elige esta opción permitiendo que la conmutación se produzca automáticamente, el proceso de conmutación se inicia cuando se inicia un canal de clúster emisor. Si el canal no se detiene, se inicia después de que quede inactivo, si hay un mensaje para procesar. Si el canal está detenido, inícielo con el mandato `START CHANNEL`.

  El proceso de conmutación se completa cuando no quedan mensajes para el canal de clúster emisor en la cola de transmisión a la que el canal prestaba servicio. En cuanto se produzca esta situación, los mensajes recién llegados para el canal de clúster emisor se almacenan directamente en la nueva cola de transmisión. Hasta entonces, los mensajes se almacenan en la cola de transmisión antigua y el proceso de conmutación transfieren mensajes de la antigua cola de transmisión a la nueva cola de transmisión. El canal de clúster emisor reenvía los mensajes de la nueva cola de transmisión de clúster durante todo el proceso de conmutación.

  El momento en el que finaliza el proceso de conmutación depende del estado del sistema. Si realiza los cambios en una ventana de mantenimiento, evalúe de antemano si el proceso de conmutación se completará a tiempo. Si se completa a tiempo depende de si el número de mensajes a la espera de transferencia de la antigua cola de transmisión llega a cero.

  La ventaja del primer método es que es automático. Un inconveniente es que si el tiempo para realizar los cambios de configuración está limitado a una ventana de mantenimiento, debe tener la seguridad de poder controlar el sistema para que complete el proceso de conmutación dentro de la ventana de mantenimiento. Si no puede estar seguro, la opción 2 puede ser más conveniente.

- Opción 2: Efectuar los cambios manualmente; consulte “Conmutación de un canal de clúster emisor detenido a otra cola de transmisión de clúster” en la página 67.

  Elija esta opción si desea controlar todo el proceso de conmutación manualmente o si desea conmutar un canal detenido o inactivo. Es una buena elección si conmuta unos cuantos canales de clúster emisor y realizar la conmutación durante una ventana de mantenimiento.

  Como descripción alternativa de esta opción, se puede decir que conmuta el canal de clúster emisor mientras el canal de clúster emisor está detenido.

  Si elige esta opción tiene control completo sobre el momento en que se produce la conmutación.

  Puede estar seguro de completar el proceso de conmutación en una cantidad de tiempo fija, dentro de una ventana de mantenimiento. El momento en que se produce la conmutación depende de cuántos mensajes se tienen que transferir de una cola de transmisión a la otra. Si siguen llegando mensajes, el proceso puede tardar un podo en transferir todos los mensajes.

  Tiene la opción de conmutar el canal sin transferir mensajes de la cola de transmisión anterior. La conmutación es "instantánea".

  Cuando se reinicia el canal de clúster emisor, se inicia el proceso de mensajes en la cola de transmisión que le acaba de asignar.

La ventaja del segundo método es que se tiene control sobre el proceso de conmutación. El inconveniente es que el usuario debe identificar los canales de clúster emisor para conmutarlos, ejecutar los mandatos necesarios y resolver cualquier canal pendiente que pueda impedir la detención del canal de clúster emisor.

Conceptos relacionados

Cómo seleccionar qué tipo de cola de transmisión de clúster se debe utilizar
Cómo elegir entre diferentes opciones de configuración de cola de transmisión de clúster.

Cómo funciona el proceso de conmutación de un canal de clúster emisor a una cola de transmisión diferente

Tareas relacionadas

Agrupación en clúster: Ejemplo de configuración de varias colas de transmisión de clúster
En esta tarea seguirá los pasos para asignar varias colas de transmisión de clúster a tres clústeres solapados. Los requisitos son separar los flujos de mensajes destinados a una cola de clúster de todos los demás flujos de mensajes y almacenar los mensajes destinados a clústeres diferentes en colas de transmisión de clúster diferentes.

Conmutación de canales de clúster emisor activos a otro conjunto de colas de transmisión de clúster
Esta tarea le ofrece tres opciones para conmutar canales de clúster emisor activos. Una opción es dejar que el gestor de colas realice la conmutación de forma automática, lo que no afecta a las aplicaciones en ejecución. Las otras opciones consisten en detener e iniciar los canales manualmente, o en reiniciar el gestor de colas.

Antes de empezar

Cambie la configuración de la cola de transmisión de clúster. Puede cambiar el atributo del gestor de colas **DEFCLXQ** o añadir o modificar el atributo **CLCHNAME** de las colas de transmisión.

Si reduce el número de mensajes que el proceso de conmutación debe transferir a la nueva cola de transmisión, la conmutación se completa más rápidamente. Consulte Cómo funciona el proceso de conmutación de un canal de clúster emisor a una cola de transmisión diferente para conocer las razones para intentar vaciar la cola de transmisión antes de continuar.

Acerca de esta tarea

Utilice los pasos de la tarea como base para diseñar su propio plan para realizar cambios de configuración de colas de transmisión de clúster.

Procedimiento

1. Opcional: Registre el estado de canal actual

   Cree un registro del estado de los canales actuales y guardados que prestan servicio a colas de transmisión de clúster. Los mandatos siguientes muestran el estado asociado con colas de transmisión de clúster del sistema. Añada sus propios mandatos para visualizar el estado asociado con colas de transmisión de clúster que haya definido. Utilice un convenio, como por ejemplo XMITQ. *ChannelName*, para nombrar las colas de transmisión de clúster que defina para facilitar la visualización del estado del canal para dichas colas de transmisión.

   ```
   DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
   DISPLAY CHSTATUS(*) SAVED WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
   ```

2. Conmute colas de transmisión.

   - No haga nada. El gestor de colas conmuta canales de clúster emisor cuando se reinician después de estar detenidos o inactivos.

     Elija esta opción si no hay reglas ni cuestiones a tener en cuenta sobre la modificación de una configuración de gestor de colas. Los cambios no afectan a las aplicaciones en ejecución.

- Reinicie el gestor de colas. Todos los canales de clúster emisor se detienen y se reinician automáticamente a demanda.

  Elija esta opción para iniciar todos los cambios inmediatamente. Las aplicaciones en ejecución se interrumpen cuando el gestor de colas concluye y se reinicia.
- Detenga canales de clúster emisor individuales y reinícielos.

  Elija esta opción para conmutar unos cuantos canales inmediatamente. Las aplicaciones en ejecución experimentar un breve retardo en la transferencia de mensajes entre la detención y el inicio del canal de mensajes. El canal de clúster emisor permanece en ejecución, excepto durante el tiempo en que lo haya detenido. Durante el proceso de conmutación los mensajes se entregan a la cola de transmisión anterior, el proceso de conmutación los transfiere a la nueva cola de transmisión y el canal de clúster emisor los reenvía desde la nueva cola de transmisión.

3. Opcional: Supervise los canales a medida que se conmutan

   Visualice el estado del canal y la profundidad de la cola de transmisión durante la conmutación. El ejemplo siguiente muestra el estado de las colas de transmisión de clúster del sistema.

   ```
   DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
   DISPLAY CHSTATUS(*) SAVED WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
   DISPLAY QUEUE('SYSTEM.CLUSTER.TRANSMIT.*') CURDEPTH
   ```

4. Opcional: Supervise los mensajes AMQ7341 `La cola de transmisión para el canal` *`NombreCanal`* `se ha conmutado de la cola` *`NombreCola`* `a` *`NombreCola`*, que se graban en el registro de errores del gestor de colas.

Conmutación de un canal de clúster emisor detenido a otra cola de transmisión de clúster
Si elige realizar cambios manualmente, realiza los cambios en un canal de clúster emisor cuando este se ha detenido y lo cambia de una cola de transmisión de clúster a otra antes de que se inicie el canal de clúster emisor.

Antes de empezar

Es posible que haga algunos cambios de configuración y ahora desea hacerlos efectivos sin iniciar los canales de clúster emisor que se ven afectados. De forma alternativa, realice los cambios de configuración que requiere como uno de los pasos de la tarea.

Si reduce el número de mensajes que el proceso de conmutación debe transferir a la nueva cola de transmisión, la conmutación se completa más rápidamente. Consulte Cómo funciona el proceso de conmutación de un canal de clúster emisor a una cola de transmisión diferente para conocer las razones para intentar vaciar la cola de transmisión antes de continuar.

Acerca de esta tarea

Esta tarea conmuta las colas de transmisión servidas por canales de clúster emisor detenidos o inactivos. Puede realizar esta tarea porque un canal de clúster emisor está detenido y desea cambiar su cola de transmisión inmediatamente. Por ejemplo, por alguna razón, un canal de clúster emisor no se inicia o tiene algún otro problema de configuración. Para resolver el problema, decide crear un canal de clúster emisor y asociar la cola de transmisión del canal de clúster emisor antiguo con el nuevo canal de clúster emisor que ha definido.

Un escenario más probable es si desea controlar cuándo se realiza la reconfiguración de las colas de transmisión de clúster. Para controlar plenamente la reconfiguración, debe detener los canales, cambiar la configuración y conmutar las colas de transmisión.

Procedimiento

1. Detenga los canales que piensa conmutar
   a) Detenga los canales en ejecución o inactivos que piense conmutar. La detención de un canal de clúster emisor inactivo impide que se inicie mientras realiza cambios de configuración.

```
STOP CHANNEL(ChannelName) MODE(QUIESCSE) STATUS(STOPPED)
```

2. Opcional: Realice los cambios de configuración.

   Por ejemplo, consulte “Agrupación en clúster: Ejemplo de configuración de varias colas de transmisión de clúster” en la página 56.

3. Cambie los canales de clúster emisor a las colas de transmisión de clúster nuevas.

   Multi En Multiplatforms, emita el mandato siguiente:

   ```
   runswchl -m QmgrName -c ChannelName
   ```

   z/OS En z/OS, utilice la función SWITCH del mandato CSQUTIL para conmutar los mensajes o para supervisar lo que está sucediendo. Utilice el mandato siguiente.

   ```
   SWITCH CHANNEL(channel_name) MOVEMSGS(YES)
   ```

   Para obtener más información, consulte Función SWITCH.

   El mandato **runswchl** o CSQUTIL SWITCH transfiere los mensajes de la cola de transmisión anterior a la cola de transmisión nueva. Cuando el número de mensajes de la cola de transmisión anterior para este canal llega a cero, la conmutación se completa. El mandato es síncrono. El mandato graba mensajes de progreso en la ventana durante el proceso de conmutación.

   Durante la fase de transferencia, los mensajes existentes y nuevos destinados al canal de clúster emisor se transfieren en orden a la nueva cola de transmisión.

   Puesto que el canal de clúster emisor está detenido, los mensajes se acumulan en la nueva cola de transmisión. Compare el canal de clúster emisor detenido con el paso “2” en la página 66 de “Conmutación de canales de clúster emisor activos a otro conjunto de colas de transmisión de clúster” en la página 66. En este paso, el canal de clúster emisor está en ejecución, de modo que los mensajes no se acumulan necesariamente en la nueva cola de transmisión.

4. Opcional: Supervise los canales a medida que se conmutan

   En una ventana de mandatos diferente, visualice la profundidad de cola de transmisión durante la conmutación. El ejemplo siguiente muestra el estado de las colas de transmisión de clúster del sistema.

   ```
   DISPLAY QUEUE('SYSTEM.CLUSTER.TRANSMIT.*') CURDEPTH
   ```

5. Opcional: Supervise los mensajes AMQ7341 `La cola de transmisión para el canal` *`NombreCanal`* `se ha conmutado de la cola` *`NombreCola`* `a` *`NombreCola`*, que se graban en el registro de errores del gestor de colas.

6. Reinicie los canales de clúster emisor que ha detenido.

   Los canales no se inician automáticamente, puesto que los ha detenido, colocándolos en el estado STOPPED.

   ```
   START CHANNEL(ChannelName)
   ```

Referencia relacionada

runswchl
RESOLVE CHANNEL
STOP CHANNEL

Agrupación en clúster: procedimientos recomendados de migración y modificación
En este tema se proporcionan instrucciones para planificar y administrar clústeres de IBM MQ. Esta información es una guía basada en las pruebas y los comentarios de los clientes.

1. "Desplazamiento de objetos en un clúster" en la página 69 (Procedimientos recomendados para mover objetos dentro de un clúster, sin instalar ningún fixpack ni versiones nuevas de IBM MQ).
2. "Actualizaciones e instalaciones de mantenimiento" en la página 70 (Procedimientos recomendados para mantener activada y en ejecución la arquitectura del clúster de trabajo mientras se aplican mantenimiento o actualizaciones y se prueba la nueva arquitectura).

Desplazamiento de objetos en un clúster

Aplicaciones y sus colas

Cuando tiene que mover una instancia de cola alojada en un gestor de colas para alojarla en otro gestor de colas, puede trabajar con los parámetros de equilibrio de la carga de trabajo para garantizar que la transición se realice sin problemas.

Cree una instancia de la cola allí donde deba alojarse por primera vez, pero utilice los valores de equilibrio de la carga de trabajo del clúster para seguir enviando mensajes a la instancia original hasta que la aplicación esté preparada para la conmutación. Para ello, siga estos pasos:

1. Establezca la propiedad **CLWLRANK** de la cola existente en un valor alto, por ejemplo cinco.
2. Cree la nueva instancia de la cola y establezca su propiedad **CLWLRANK** en cero.
3. Realice la configuración adicional del nuevo sistema, por ejemplo, despliegue e inicie las aplicaciones consumidoras con la nueva instancia de la cola.
4. Establezca la propiedad **CLWLRANK** de la nueva instancia de cola para que sea superior a la instancia original, por ejemplo nueve.
5. Permita que la instancia de cola original procese los mensajes en cola en el sistema y, a continuación, suprima la cola.

Movimiento de gestores de colas completos

Si el gestor de colas se queda en el mismo host, pero la dirección IP cambia, el proceso es el siguiente:

- El DNS, si se utiliza correctamente, puede ayudar a simplificar el proceso. Para obtener información sobre el uso de DNS estableciendo el atributo de canal Nombre de conexión (CONNAME) , consulte ALTER CHANNEL.
- Si mueve un repositorio completo, asegúrese de tener como mínimo otro repositorio completo que se esté ejecutando correctamente (sin problemas con el estado de canal, por ejemplo) antes de realizar cambios.
- Suspenda el gestor de colas mediante el mandato SUSPEND QMGR para evitar que se acumule el tráfico.
- Modifique la dirección IP del sistema. Si la definición de canal CLUSRCVR utiliza una dirección IP en el campo CONNAME, modificar esta entrada de dirección IP. Es posible que deba vaciar la memoria caché de DNS para garantizar que las actualizaciones estén disponibles en todas partes.
- Cuando el gestor de colas se vuelve a conectar a los repositorios completos, las definiciones automáticas de canal se resuelven automáticamente por si mismas.
- Si el gestor de colas aloja un repositorio completo y la dirección IP cambia, es importante garantizar que los parciales cambien lo antes posible para que los canales CLUSSDR definidos manualmente apunten a la nueva ubicación. Hasta que se realice este cambio, los gestores de colas sólo podrán ponerse en contacto con el repositorio completo restante (sin modificar) y pueden aparecer mensajes de aviso relacionados con la definición de canal incorrecta.
- Reanude el gestor de colas mediante el mandato RESUME QMGR.

Si el gestor de colas debe desplazarse a un nuevo host, es posible copiar los datos del gestor de colas y restaurarlos desde una copia de seguridad. No obstante, no se recomienda este proceso a menos

que no haya otras opciones; es preferible crear un gestor de colas en una máquina nueva y replicar las colas y las aplicaciones, tal y como se describe en la sección anterior. Esta situación proporciona un mejor mecanismo de aplazamiento/retrotracción.

Si desea realmente mover un gestor de colas completo utilizando la copia de seguridad, siga estos procedimientos recomendados:

- Maneje todo el proceso como una restauración del gestor de colas desde la copia de seguridad, aplicando los procesos que utilizaría normalmente para la recuperación del sistema, según corresponda para su entorno de sistema operativo.
- Utilice el mandato **REFRESH CLUSTER** después de la migración para descartar toda la información de clúster retenida localmente (incluidos los canales definidos automáticamente que estén pendientes) y fuerce su reconstrucción.

  Nota: Para los clústeres de gran tamaño, el uso del mandato **REFRESH CLUSTER** puede interrumpir el clúster mientras está en curso y, a partir de entonces, de nuevo a intervalos de 27 días cuando los objetos de clúster envían automáticamente actualizaciones de estado a todos los gestores de colas interesados. Consulte La renovación en un clúster grande puede afectar el rendimiento y la disponibilidad del clúster.

Cuando se crea un gestor de colas y se replica la instalación desde un gestor de colas existente en el clúster (como se ha descrito anteriormente en este tema), no trate nunca los dos gestores de colas diferentes como si fueran el mismo. En particular, no asigne al nuevo gestor de colas el mismo nombre de gestor de colas y la misma dirección IP. El intento de 'insertar' un gestor de colas de sustitución es una causa frecuente de problemas en los clústeres de IBM MQ. La memoria caché espera recibir actualizaciones, incluido el atributo **QMID**, y el estado puede estar dañado.

Si se crean accidentalmente dos gestores de colas diferentes con el mismo nombre, se recomienda utilizar el mandato RESET CLUSTER**QMID** para expulsar la entrada incorrecta del clúster.

Actualizaciones e instalaciones de mantenimiento

Evite el denominado escenario de tipo big bang (por ejemplo, detener toda la actividad del clúster y del gestor de colas, aplicar todas las actualizaciones y todo el mantenimiento a todos los gestores de colas y luego iniciar todo al mismo tiempo). Los clústeres están diseñados para seguir funcionando con varias versiones de gestores de colas coexistiendo, por lo que se recomienda un enfoque de mantenimiento por fases bien planificado.

Tenga un segundo plan:

- ¿Ha realizado copias de seguridad?
- Evite utilizar la nueva funcionalidad de clúster de forma inmediata: espere hasta que esté seguro de que los gestores de colas se han actualizado al nuevo nivel y de que no va a volver a una versión anterior de ninguno de ellos. Utilizar una función nueva en un clúster en el que algunos gestores de colas todavía se encuentran en un nivel anterior puede dar lugar a un comportamiento no definido.

  Un repositorio almacena un registro que recibe en su propia versión. Si el registro que recibe se encuentra en una versión posterior, los atributos de la versión posterior se descartan cuando se almacena el registro. Un gestor de colas de IBM MQ 9.3 que recibe información sobre un gestor de colas de IBM MQ 9.4 sólo almacena información de IBM MQ 9.3 . Un repositorio de IBM MQ 9.4 que recibe un registro de IBM MQ 9.3 almacena valores predeterminados para los atributos introducidos en la versión posterior. Los valores predeterminados definen los valores de los atributos que no están incluidos en el registro que recibe.

Migre primero los repositorios completos. Aunque pueden reenviar información que no comprenden, no pueden conservarla, por lo que no es el enfoque recomendado, a menos que sea absolutamente necesario. Para obtener más información, consulte la sección Migración del clúster de gestores de colas.

Agrupación en clúster: utilización de las recomendaciones de REFRESH CLUSTER
Puede utilizar el mandato **REFRESH CLUSTER** para descartar toda la información retenida localmente sobre un clúster y reconstruir esa información a partir de los repositorios completos en el clúster. Es poco probable que necesite utilizar este mandato, excepto en circunstancias excepcionales. Si necesitara

utilizar este mandato, existen algunas consideraciones especiales sobre cómo se utiliza. Esta información es una guía basada en las pruebas y los comentarios de los clientes.

Ejecute REFRESH CLUSTER sólo si verdaderamente lo necesita

La tecnología de clúster de IBM MQ garantiza que cualquier cambio en la configuración del clúster como, por ejemplo, un cambio en una cola de clúster, sea reconocible automáticamente para cualquier miembro del clúster que necesite saber la información. No es necesario realizar más pasos administrativos para lograr esta propagación de información.

Si esta información no llega a los gestores de colas del clúster cuando es necesario, por ejemplo, si otro gestor de colas del clúster no conoce una cola de clúster cuando una aplicación intenta abrirla por primera vez, se produce un problema en la infraestructura del clúster. Por ejemplo, es posible que un canal no pueda iniciarse entre un gestor de colas y un gestor de colas de repositorio completo. Por lo tanto, las situaciones donde se observen incoherencias deben investigarse. Si es posible, resuelva la situación sin utilizar el mandato **REFRESH CLUSTER**.

En raras circunstancias que están documentadas en esta documentación del producto, o cuando lo solicite el soporte de IBM, puede utilizar el mandato **REFRESH CLUSTER** para descartar toda la información retenida localmente sobre un clúster y reconstruir esa información a partir de los repositorios completos en el clúster.

La actualización en un clúster de gran tamaño puede afectar al rendimiento y la disponibilidad del clúster

El uso del mandato **REFRESH CLUSTER** puede ser perjudicial para el clúster mientras está en curso, por ejemplo, creando un aumento repentino del trabajo para los repositorios completos a medida que procesan la repropagación de los recursos del clúster del gestor de colas. Si está actualizando un clúster de gran tamaño (es decir, de varios cientos de gestores de colas) debe evitar el uso del mandato en el trabajo diario, si es posible, y utilizar métodos alternativos para corregir incoherencias específicas. Por ejemplo, si una cola de clúster no se propaga correctamente en el clúster, una técnica de investigación inicial de actualizar la definición de cola de clúster ,por ejemplo, modificando su descripción), vuelve a propagar la configuración de la cola en el clúster. Este proceso permite identificar el problema y posiblemente resolver una incoherencia temporal.

Si no se pueden utilizar métodos alternativos y tiene que ejecutar **REFRESH CLUSTER** en un clúster grande, debe hacerlo en horas de menor actividad o durante una ventana de mantenimiento para evitar el impacto en las cargas de trabajo del usuario. También debe evitar actualizar un clúster grande en un solo lote. Es mejor escalonar la actividad como se explica en “Evitar problemas de rendimiento y disponibilidad cuando los objetos de clúster envían actualizaciones automáticas” en la página 71.

Evitar problemas de rendimiento y disponibilidad cuando los objetos de clúster envían actualizaciones automáticas

Después de que se defina un nuevo objeto de clúster en un gestor de colas, se genera una actualización para este objeto cada 27 días desde el momento de la definición, y se envía a cada repositorio completo en el clúster y a otros gestores de colas de interesados. Cuando emite el mandato **REFRESH CLUSTER** a un gestor de colas, restablece el reloj para esta actualización automática en todos los objetos definidos localmente en el clúster especificado.

Si actualiza un clúster de gran tamaño (es decir, de varios cientos de gestores de colas) en un único lote, o en otras circunstancias como volver a crear un sistema a partir de la copia de seguridad de la configuración, después de 27 días, todos los gestores de colas volverán a anunciar todas sus definiciones de objetos en los repositorios completos al mismo tiempo. Esto podría volver a ralentizar de forma significativa la ejecución del sistema o incluso que éste no estuviera disponible, hasta que se hayan completado todas las actualizaciones. Por lo tanto, cuando haya que renovar o volver a crear varios gestores de colas en un clúster grande, habrá que escalonar la actividad durante varias horas o varios días, de modo que las actualizaciones automáticas sucesivas no penalicen el rendimiento del sistema de forma periódica.

La cola del historial del clúster del sistema

Cuando se emite un mandato **REFRESH CLUSTER**, el gestor de colas realiza una instantánea del estado del clúster antes de la renovación y la almacena en SYSTEM.CLUSTER.HISTORY.QUEUE (SCHQ) si está definido en el gestor de colas. Esta instantánea está únicamente indicada para el servicio de IBM, en el caso de que se produzcan problemas más adelante con el sistema.

La cola SCHQ se define de forma predeterminada en los gestores de colas distribuidos durante el proceso de arranque. Para la migración de z/OS, la SCHQ debe definirse manualmente.

Los mensajes de la SCHQ caducan al cabo de tres meses.

Conceptos relacionados

"Consideraciones sobre REFRESH CLUSTER para clústeres de publicación/suscripción" en la página 108
La emisión del mandato **REFRESH CLUSTER** hace que el gestor de colas descarte temporalmente la información sobre un clúster guardada localmente, incluidos los temas de clúster y sus suscripciones de proxy asociadas.

Referencia relacionada

Problemas de aplicación vistos al ejecutar REFRESH CLUSTER

Referencia de mandatos MQSC: REFRESH CLUSTER

Agrupación en clúster: disponibilidad, multiinstancia y recuperación de desastres

En este tema se proporcionan instrucciones para planificar y administrar clústeres de IBM MQ. Esta información es una guía basada en las pruebas y los comentarios de los clientes.

La agrupación en clúster de IBM MQ en sí misma no es una solución de alta disponibilidad, pero en algunos casos se puede utilizar para mejorar la disponibilidad de los servicios que utilizan IBM MQ, por ejemplo, teniendo varias instancias de una cola en distintos gestores de colas. En esta sección se proporcionan instrucciones para garantizar que la infraestructura de IBM MQ tenga la máxima disponibilidad posible para que pueda utilizarse en este tipo de arquitectura.

Nota: Hay otras soluciones de alta disponibilidad y recuperación tras desastre disponibles para IBM MQ, consulte Configuración de alta disponibilidad, recuperación y reinicio.

Disponibilidad de recursos de clúster

El motivo de la recomendación habitual de mantener dos repositorios completos es que la pérdida de uno de ellos no es una situación crítica para la correcta ejecución del clúster. Incluso si ambos no están disponibles, hay un periodo de gracia de 60 días para los conocimientos existentes en los repositorios parciales, aunque los recursos nuevos o a los que no se haya accedido previamente (colas, por ejemplo) no estén disponibles en este evento.

Utilización de clústeres para mejorar la disponibilidad de la aplicación

Un clúster puede ayudar en el diseño de aplicaciones altamente disponibles (por ejemplo, una aplicación de servidor de tipo solicitud/respuesta), utilizando varias instancias de la cola y la aplicación. Si es necesario, los atributos de prioridad pueden dar preferencia a la aplicación 'activa', a menos que, por ejemplo, un gestor de colas o un canal deje de estar disponible. Esto es muy útil para conmutar rápidamente y continuar procesando mensajes nuevos cuando se produce un problema.

No obstante, los mensajes que se han entregado a un determinado gestor de colas en un clúster se mantienen sólo en esa instancia de la cola y no están disponibles para su proceso hasta que se recupere ese gestor de colas. Por este motivo, para conseguir una verdadera alta disponibilidad de los datos, se recomienda utilizar otras tecnologías como, por ejemplo, los gestores de colas multiinstancia.

Gestores de colas multiinstancia

La alta disponibilidad de software (multiinstancia) es una oferta incorporada para mantener disponibles los mensajes existentes. Consulte Uso de IBM MQ con configuraciones de alta disponibilidad, Crear un gestor de colas de varias instancias y la sección siguiente para obtener más información. Con esta técnica, puede lograr que cualquier gestor de colas en un clúster tenga una alta disponibilidad, siempre que todos los gestores de colas del clúster estén en ejecución como mínimo IBM WebSphere MQ 7.0.1. Si algunos de los gestores de colas del clúster se encuentran en niveles

anteriores, pueden perder la conectividad con los gestores de colas multiinstancia si se migran tras error a una IP secundaria.

Como se ha descrito anteriormente en este tema, si hay dos repositorios completos configurados, serán altamente disponibles prácticamente por su naturaleza. Si es necesario, se pueden utilizar gestores de colas multiinstancia / alta disponibilidad de software de IBM MQ en los repositorios completos. No hay ninguna razón firme para utilizar estos métodos y, de hecho, para las interrupciones temporales, estos métodos pueden suponer un coste adicional de rendimiento durante la migración tras error. No se recomienda utilizar la HA de software en lugar de ejecutar dos repositorios completos, porque en el caso de la caída de un único canal, por ejemplo, no se migraría tras error necesariamente, sino que podrían quedar repositorios parciales sin capacidad de consultar recursos del clúster.

Recuperación tras desastre

La recuperación tras desastre, por ejemplo la recuperación cuando se dañan los discos que almacenan los datos de un gestor de colas, es difícil de gestionar; IBM MQ puede ayudar, pero no puede hacerlo automáticamente. La única opción "verdadera" de recuperación tras desastre en IBM MQ (excluyendo cualquier sistema operativo u otras tecnologías de réplica subyacentes) es la restauración a partir de una copia de seguridad. Debe tener en cuenta algunos puntos específicos del clúster en estos casos:

- Tenga cuidado cuando pruebe los escenarios de recuperación tras desastre. Por ejemplo, si se prueba la operación de los gestores de colas de copia de seguridad, se debe tener cuidado al ponerlas en línea en la misma red, ya que es posible unirse accidentalmente al clúster activo y empezar a "robar" mensajes alojando las colas con el mismo nombre que en los gestores de colas de clúster activas.
- Las pruebas de recuperación ante desastre no deben interferir con un clúster en un entorno real. Algunas de las técnicas para evitar interferencias son las siguientes:
  - Separación de red completa o separación a nivel de cortafuegos.
  - z/OS No se inicia la iniciación de canal o el espacio de direcciones de z/OS **chinit** .
  - No emitir el certificado TLS en directo para el sistema de recuperación ante desastres hasta que, o a menos que, se produzca el escenario de recuperación ante desastres.
- Cuando se restaura una copia de seguridad de un gestor de colas en el clúster, es posible que la copia de seguridad no esté sincronizada con el resto del clúster. El mandato **REFRESH CLUSTER** puede resolver las actualizaciones y sincronizarse con el clúster, pero el mandato **REFRESH CLUSTER** debe utilizarse como último recurso. Consulte “Agrupación en clúster: utilización de las recomendaciones de REFRESH CLUSTER” en la página 70. Revise la documentación de proceso interno y la documentación de IBM MQ para ver si no se ha realizado un paso simple antes de recurrir a la utilización del mandato.
- Para las recuperaciones, las aplicaciones deben ocuparse de la reproducción y pérdida de datos. Se debe decidir si las colas se van a dejar en un estado conocido o si existe suficiente información en algún lugar para gestionar las reproducciones.

Planificación de su red de publicación/suscripción distribuida

Puede crear una red de gestores de colas en la que las suscripciones que se crean en un gestor de colas reciben los mensajes coincidentes que ha publicado una aplicación conectada a otro gestor de colas de la red. Para seleccionar una topología adecuada, debe tener en cuenta sus requisitos de control manual, tamaño de red, frecuencia de cambios, disponibilidad y escalabilidad.

Antes de empezar

Esta tarea presupone que comprende qué son las redes de publicación/suscripción distribuidas y cómo funcionan. Para obtener una visión general técnica, consulte Redes de publicación/suscripción distribuidas.

Acerca de esta tarea

Existen tres topologías básicas para una red de publicación/suscripción:

- Clústeres de direccionamiento directo
- Clúster de direccionamiento de host de tema
- Jerarquía

En las dos primeras topologías, el punto inicial es una configuración de clúster de IBM MQ. La tercera topología se puede crear con o sin un clúster. Consulte la sección “Planificación de sus gestores de colas y clústeres distribuidos” en la página 20 para obtener información acerca de cómo planificar la red de gestor de colas subyacente.

Un *Clúster de direccionamiento directo* es la topología de configuración más sencilla cuando ya está presente un clúster. Cualquier tema que defina en cualquier gestor de colas está disponible automáticamente en cada uno de los gestores de colas del clúster y las publicaciones se direccionan directamente desde cualquier gestor de colas al que se conecte una aplicación de publicación hasta cada uno de los gestores de colas donde existan suscripciones coincidentes. Esta simplificada de la configuración confía en IBM MQ para mantener un alto nivel de compartición de la información y la conectividad entre cada uno de los gestores de colas del clúster. En las redes pequeñas y simples (esto es, con un número reducido de gestores de colas y un conjunto suficientemente estático de publicadores y suscriptores) esto resulta aceptable. Sin embargo, cuando se utilizan en entornos de mayor tamaño y más dinámicos la carga adicional puede resultar prohibitiva. Consulte “Direccionamiento directo en clústeres de publicación/suscripción” en la página 79.

Un *Clúster de direccionamiento de host de tema* proporciona las mismas ventajas que un clúster de direccionamiento directo ya que todos los temas que defina en cualquier gestor de colas del clúster estarán disponibles automáticamente en cada uno de los gestores de colas del clúster. Sin embargo, los clústeres de direccionamiento de host de tema requieren que seleccione detenidamente los gestores de colas que alojan cada tema, ya que toda la información y las publicaciones de dicho tema pasan a través de estos gestores de colas de host de tema. Esto significa que el sistema no tiene que mantener canales y flujos de información entre todos los gestores de colas. Sin embargo, también significa que es posible que las publicaciones ya no se envíen directamente a los suscriptores sino que pueden direccionarse a través de un gestor de colas de host de tema. Por estos motivos, es posible que exista una carga adicional en el sistema, sobretodo en los gestores de colas que alojan los temas, por lo que debe planificar con atención la topología. Esta topología es especialmente eficaz para las redes que contienen muchos gestores de colas o que alojan un conjunto dinámico de publicadores y suscriptores (es decir, los publicadores o suscriptores que se añaden o eliminan con frecuencia). Se pueden definir hosts de temas adicionales para mejorar la disponibilidad de los direccionamientos y para escalar horizontalmente la carga de trabajo de las publicaciones. Consulte “Direccionamiento de host de tema en clústeres de publicación/suscripción” en la página 84.

La *Jerarquía* es la que requiere más configuración manual y es la topología para difícil de modificar. Debe configurar manualmente las relaciones entre cada gestor de colas de la jerarquía y sus relaciones directas. Una vez configuradas las relaciones, al igual que para las dos topologías anteriores, las publicaciones se direccionarán a las suscripciones de los otros gestores de colas de la jerarquía. Las publicaciones se direccionan utilizando las relaciones de la jerarquía. Esto permite configurar topologías muy específicas, pero también da como resultado que las publicaciones requieran muchos "saltos" a través de los gestores de colas hasta llegar a las suscripciones. Siempre existe una sola ruta a través de una jerarquía para una publicación, por lo tanto, la disponibilidad de cada uno de los gestores de colas resulta crítica. Normalmente, las jerarquías solo resultan preferibles cuando no se puede configurar un único clúster, por ejemplo, cuando se abarcan varias organizaciones. Consulte “Direccionamiento en las jerarquías de publicación/suscripción” en la página 109.

Siempre que sea necesario, se pueden combinar las tres topologías mencionadas para resolver requisitos topográficos específicos. Para ver un ejemplo, consulte la sección Combinar los espacios de temas de varios clústeres.

Para elegir una topología adecuada para su red de publicación/suscripción distribuida, debe tener en cuenta las preguntas siguientes:

- ¿De qué tamaño será su red?
- ¿Cuánto control manual necesita sobre su configuración?
- ¿Cuál será el dinamismo de su sistema, tanto en términos de temas y suscripciones como en términos de gestores de colas?
- ¿Cuáles son sus requisitos de disponibilidad y escalabilidad?
- ¿Los gestores de colas pueden conectarse directamente entre sí?

Procedimiento

- Calcule de qué tamaño debe ser su red.
  a) Calcule cuántos temas necesita.
  b) Calcule el número de publicadores y suscriptores que puede llegar a tener.
  c) Calcule el número de gestores de colas implicados en las actividades de publicación/suscripción.

  Consulte también el tema "Agrupación en clúster de la publicación/suscripción: Mejoras prácticas" en la página 94, en especial las secciones siguientes:

  – Cómo medir el sistema
  – Motivos para limitar el número de gestores de colas del clúster implicados en la actividad de publicación/suscripción
  – Cómo decidir qué temas se han de incluir en el clúster

  Si su red tendrá muchos gestores de colas y manejará muchos publicadores y suscriptores, probablemente necesite utilizar un clúster de direccionamiento directo o una jerarquía. Los clústeres de direccionamiento directo prácticamente no requieren ninguna configuración manual y pueden resultar una buena solución para las redes pequeñas o estáticas.
- Considere cuánto control manual necesitará sobre qué gestor de colas alojará cada tema, publicador o suscriptor.
  a) Considere si algunos de los gestores de colas son menos capaces que otros.
  b) Considere si los enlaces de comunicaciones con algunos de los gestores de colas son más frágiles que otros.
  c) Identifique los casos en los que espera que un tema tenga muchas publicaciones y pocos suscriptores.
  d) Identifique los casos en los que espera que un tema tenga muchos suscriptores y pocas publicaciones.

  En todas las topologías, las publicaciones se entregan a las suscripciones en otros gestores de colas. En un clúster de direccionamiento directo estas publicaciones toman la ruta más corta para llegar a las suscripciones. En un clúster de direccionamiento de host de tema o en una jerarquía puede controla la ruta que toma las publicaciones. Si sus gestores de colas difieren con respecto a sus posibilidades o si tienen diferentes niveles de disponibilidad y conectividad, probablemente desee asignar cargas de trabajo específicas a gestores de colas específicos. Puede llevarlo a cabo utilizando un clúster de direccionamiento de host de tema o una jerarquía.

  En todas las topologías, cuando se coubican las aplicaciones de publicación en el mismo gestor de colas que las suscripciones, siempre que sea posible, se minimiza la carga adicional y se aumenta el rendimiento. En los clústeres de direccionamiento de host de tema, puede colocar los publicadores o suscriptores en los gestores de colas que alojan el tema. Esto elimina cualquier "salto" adicional entre los gestores de colas que pasan una publicación a un suscriptor. Este método es especialmente eficaz en los casos en los que un tema tiene muchas publicaciones y pocos suscriptores o muchos suscriptores y pocos publicadores. Consulte, por ejemplo, el tema Direccionamiento de host de tema mediante publicadores o suscriptores centralizados.

  Consulte también el tema "Agrupación en clúster de la publicación/suscripción: Mejoras prácticas" en la página 94, en especial las secciones siguientes:

- Cómo decidir qué temas se han de incluir en el clúster
- Ubicación del publicador y la suscripción

- Considere si la actividad de red será muy dinámica o no.
  a) Calcule la frecuencia con las que se añadirán o eliminarán suscriptores a diferentes temas.

     Cuando se añade o elimina una suscripción desde un gestor de colas y ésta es la primera o la última suscripción de dicha serie de tema específica, esta información se comunica a los otros gestores de colas de la topología. En una clúster de direccionamiento directo y una jerarquía, esta información de suscripciones se propaga a cada gestor de colas de la topología,independientemente de si tienen o no publicadores sobre el tema. Si la topología consta de muchos gestores de colas, es posible que esto genere una sobrecarga importante en el rendimiento. En un clúster de direccionamiento de host de tema, esta información solo se propaga a los gestores de colas que alojan un tema en clúster que se correlaciona con la serie de tema de la suscripción.

     Consulte también la sección Cambio de suscripción y series de temas dinámicos del tema “Agrupación en clúster de la publicación/suscripción: Mejoras prácticas” en la página 94.

     Nota: En cada sistema dinámico, en el que el conjunto de series de temas exclusivo se modifica rápidamente y de forma constante, es posible que resulte mejor pasar el modelo a la modalidad "publicación en todas partes". Consulte Rendimiento de suscripción en redes de publicación/suscripción.
  b) Considere si los gestores de la topología han de ser muy dinámicos.

     Una jerarquía requiere que cada cambio realizado en el gestor de colas de la topología se inserte o elimine manualmente de la jerarquía, prestando atención cuando se cambian los gestores de colas en los niveles más altos de la jerarquía. Normalmente, los gestores de colas de una jerarquía también utilizan las conexiones de canal configurado. Debe mantener estas conexiones, añadiendo y eliminando los canales a medida que se añaden o eliminan gestores de colas de la jerarquía.

     En un clúster de publicación/suscripción, los gestores de colas se conectan automáticamente a cualquier otro gestor de colas que sea necesario cuando se unen por primera vez al clúster y, automáticamente, pasan a informarse sobre los temas y suscripciones.
- Considere los requisitos de disponibilidad de su ruta y la escalabilidad del tráfico de publicaciones.
  a) Decida si necesita que esté disponible siempre una ruta desde un gestor de colas de publicación a un gestor de colas de suscripción, incluso cuando un gestor de colas no esté disponible.
  b) Considere el nivel de escalabilidad que necesita para la red. Decida si el nivel de tráfico de publicaciones es demasiado elevado para direccionarlo a través de un solo gestor de colas o canal y si dicho nivel de tráfico de publicaciones debe manejarlo una sola rama de tema o si se puede distribuir en varias ramas de temas.
  c) Considere si necesita mantener el orden de los mensajes.

  Dado que un clúster de direccionamiento directo envía mensajes directamente desde los gestores de colas de publicación a los gestores de colas de suscripción, no es necesario que tenga en cuenta la disponibilidad de los gestores de colas intermedios de la ruta. Del mismo modo, no es necesario considera el escalado a los gestores de colas intermedios. Sin embargo, como se ha mencionado anteriormente, la actividad adicional de mantener automáticamente canales y flujos de información entre todos los gestores de colas del clúster puede afectar de forma importante el rendimiento, sobretodo en un entorno de gran tamaño o dinámico.

  Un clúster de direccionamiento de host de tema se puede ajustar para temas individuales. Puede asegurarse de que cada rama del árbol de temas que tenga una carga de trabajo de publicaciones considerable esté definida en un gestor de colas diferente y que el rendimiento y la disponibilidad de dicho gestor de colas sean suficientes para la carga de trabajo prevista para dicha rama del árbol de temas. También puede mejorar adicionalmente la disponibilidad y el escalado horizontal definiendo cada tema en varios gestores de colas. Esto permite que el sistema pase por los gestores de colas de host de tema que no están disponibles y se equilibre la carga de trabajo del tráfico de publicaciones entre los mismos. Sin embargo, cuando define un tema concreto en varios gestores de colas se presentan las siguientes restricciones:

  - Se pierde el orden de los mensajes entre publicaciones.
  - NO se pueden utilizar las publicaciones retenidas. Consulte "Consideraciones de diseño acerca de las publicaciones retenidas en los clústeres de publicación/suscripción" en la página 107.

  No se puede configurar un direccionamiento de alta disponibilidad y escalabilidad en una jerarquía a través de varias rutas.

  Consulte también la sección Tráfico de publicaciones del tema "Agrupación en clúster de la publicación/suscripción: Mejoras prácticas" en la página 94.
- Basándose en estos cálculos, utilice los enlaces proporcionados como ayuda para decidir si utiliza un clúster de direccionamiento de host de tema, un clúster de direccionamiento directo o una combinación de estas topologías.

Qué hacer a continuación

Ahora está preparado para configurar su red de publicación/suscripción distribuida.

Tareas relacionadas

Configuración de un clúster de gestores de colas

Configuración de la gestión de colas distribuidas

Configurar un clúster de publicación/suscripción

Conexión de un gestor de colas a una jerarquía de publicación/suscripción

Diseño de clústeres de publicación/suscripción

Existen dos topologías básicas de clúster de publicación/suscripción: el *direccionamiento directo* y el *direccionamiento de host de tema*. Cada uno de ellos tiene ventajas diferentes. Cuando diseñe su clúster de publicación/suscripción, elija la topología que mejor se ajuste a sus requisitos de red.

Para obtener una visión general de las dos topologías de clúster de publicación/suscripción, consulte Clústeres de publicación/suscripción. Como ayuda para evaluar sus requisitos de red, consulte el tema "Planificación de su red de publicación/suscripción distribuida" en la página 73 y el tema "Agrupación en clúster de la publicación/suscripción: Mejoras prácticas" en la página 94.

Por lo general, las dos topologías de clúster proporcionan las siguientes ventajas:

- Configuración simple sobre una topología de clúster punto a punto.
- Manejo automático de los gestores de colas que se unen y abandonan el clúster.
- Facilidad de escalar los suscriptores y publicadores adicionales, añadiendo gestores de colas adicionales y distribuyendo los suscriptores y publicadores adicionales entre ellos.

Sin embargo, las dos topologías tienen ventajas diferentes a medida que los requisitos pasan a ser más específicos.

Clústeres de publicación/suscripción direccionados de forma directa

En el caso del direccionamiento directo, cada gestor de colas del clúster envía publicaciones desde las aplicaciones conectadas a cualquier gestor de colas del clúster que tenga una suscripción coincidente.

Un clúster de publicación/suscripción de direccionamiento directo ofrece las ventajas siguientes:

- Los mensajes destinados a una suscripción en un gestor de colas específico en el mismo clúster se transportan directamente a ese gestor de colas y no tienen que pasar a través de un gestor de colas intermedio. Esto puede mejorar el rendimiento en comparación con una topología de direccionamiento de host de tema o una topología jerárquica.
- Dado que todos los gestores de colas están conectados directamente entre sí, no existe un único punto de anomalía en la infraestructura de direccionamiento de esta topología. Si un gestor de colas no está disponible, las suscripciones en otros gestores de colas del clúster todavía pueden recibir mensajes de los publicadores en los gestores de colas disponibles.
- Su configuración es sencilla, especialmente en un clúster existente.

Aspectos a tener en cuenta cuando se utiliza un clúster de publicación/suscripción de direccionamiento directo:

- Todos los gestores de colas de un clúster reconocen automáticamente a los otros gestores de colas del clúster.
- Los gestores de colas de un clúster que alojan una o varias suscripciones a un tema en clúster crean automáticamente canales de emisor de clúster con todos los demás gestores de colas del clúster, incluso cuando dichos gestores de colas no están publicando mensajes en ningún tema en clúster.
- La primera suscripción en un gestor de colas a una serie de tema en un tema en clúster da como resultado el envío de un mensaje a los otros gestores de colas del clúster. De forma parecida, la última suscripción en una serie de tema que se va a suprimir también da como resultado un mensaje. Cuantas más series de tema individuales se utilicen en un tema en clúster y cuanto más alta sea la tasa de cambios de las suscripciones, mayor será la comunicación interna entre los gestores de colas.
- Cada gestor de colas del clúster mantiene el conocimiento de las series de temas suscritas de las que se informa, incluso que el gestor de colas no está publicando ni está suscrito a dichos temas.

Por los motivos mencionados, todos los gestores de colas con un tema de direccionamiento directo que se hayan definido tendrán una actividad general adicional. Cuantos más gestores de colas haya en el clúster, mayor será la actividad general. Del mismo modo, cuanto mayor sea el número de series suscritas y mayor sea su tasa de cambios, mayor será la actividad general. Esto puede generar una carga de trabajo excesiva en los gestores de colas de sistemas pequeños en un clúster de publicación/suscripción de direccionamiento directo dinámico o de gran tamaño. Para obtener más información consulte el tema Rendimiento de la publicación/suscripción de direccionamiento directo.

Si sabe que un clúster no puede asumir la sobrecarga del uso de publicación/suscripción del clúster de direccionamiento directo, puede utilizar Clústeres de publicación/suscripción direccionados de host de tema. De forma alternativa, en situaciones extremas, puede inhabilitar por completo la función de publicación/suscripción en clúster estableciendo el atributo del gestor de colas **PSCLUS** en DISABLED en cada gestor de colas del clúster. Consulte “Inhabilitación de la publicación/suscripción en un clúster” en la página 105. Esto impedirá que se cree cualquier tema en clúster y, por lo tema, se asegura de que la red no entre en ninguna actividad adicional asociada con la publicación/suscripción en clúster.

Clústeres de publicación/suscripción con direccionamiento de host de tema

Con el direccionamiento de host de tema, los gestores de colas en los que los temas en clúster se definen de forma administrativa pasan a convertirse en direccionadores de las publicaciones. Las publicaciones de gestores de colas que no son de host en el clúster se direccionan a través del gestor de colas de host a cualquier gestor de colas del clúster con una suscripción coincidente.

Un clúster de publicación/suscripción de direccionamiento de host de tema ofrece las siguientes ventajas adicionales en relación con un clúster de publicación/suscripción de direccionamiento directo:

- Solo los gestores de colas en los que se han definido temas de direccionamiento de host de tema reconocen a todos los otros gestores de colas del clúster.
- Sólo es necesario que los gestores de colas de host de tema se conecten a todos los otros gestores de colas del clúster y, normalmente, solo se conectarán a aquellos donde existan suscripciones. Por lo tanto, habrá muchos menos canales en ejecución entre los gestores de colas.
- Los gestores de colas que alojan una o varias suscripciones a un tema en clúster automáticamente crean canales de emisor de clúster solo con los gestores de colas que alojan un tema de clúster que se correlaciona con la serie de tema de la suscripción.
- La primera suscripción de un gestor de colas a una serie de tema bajo un tema en clúster da como resultado el envío de un mensaje a un gestor de colas del clúster que aloja el tema en clúster. De forma parecida, la última suscripción en una serie de tema que se va a suprimir también da como resultado un mensaje. Cuantas más series de tema individuales se utilicen en un tema en clúster, y cuanto más alta sea la tasa de cambios de suscripciones, mayor será la comunicación interna entre los gestores de colas pero únicamente entre los hosts de suscripciones y los hosts de temas.
- Más control sobre la configuración física. Con el direccionamiento directo todos los gestores de colas tienen que participar en el clúster de publicación/suscripción, lo que aumenta su actividad general. En

el caso del direccionamiento de host de tema, solo los gestores de colas de host de tema reconocen a los otros gestores de colas y sus suscripciones. Los gestores de colas de host de tema se eligen de forma explícita, por lo tanto, puede asegurarse de que los gestores de colas se ejecuten en el equipo adecuado, y puede utilizar sistemas menos potentes para los otros gestores de colas.

Aspectos a tener en cuenta cuando se utiliza un clúster de publicación/suscripción de direccionamiento de host de tema:

- Se introduce un "salto" adicional entre un gestor de colas de publicación y un gestor de colas de suscripción cuando el publicador o el suscriptor no está ubicado en un gestor de colas de host de tema. La latencia que origina el "salto" adicional puede hacer que el direccionamiento de host de tema sea menos eficaz que el direccionamiento directo.
- En clústeres de gran tamaño, el direccionamiento de host de tema alivia los importantes problemas de rendimiento y escalado que se producen con el direccionamiento directo.
- Puede optar por definir todos los temas en un gestor de colas individual, o en un número muy pequeño de gestores de colas. Si lo prefiere, asegúrese de que los gestores de colas de host de tema se alojen en sistemas potentes con buena conectividad.
- Puede definir el mismo tema en más de un gestor de colas. Esto mejora la disponibilidad del tema, y también mejora la escalabilidad porque la carga de trabajo de IBM MQ equilibra las publicaciones para un tema en todos los hosts de ese tema. Tenga en cuenta, sin embargo, que al definir el mismo tema en más de un gestor de colas se pierde el orden de los mensajes para ese tema.
- Al alojar distintos temas en distintos gestores de colas, puede mejorar la escalabilidad sin perder el orden de los mensajes.

Tareas relacionadas

Escenario: Creación de un clúster de publicación/suscripción

Configurar un clúster de publicación/suscripción

Ajuste de redes de publicación/suscripción distribuidas

Resolución de problemas de publicación/suscripción distribuida

Direccionamiento directo en clústeres de publicación/suscripción

Las publicaciones de cualquier gestor de colas de publicación se direccionan directamente a cualquier otro gestor de colas del clúster con una suscripción coincidente.

Para obtener una introducción sobre cómo se direccionan los mensajes entre gestores de colas en jerarquías de publicación/suscripción y clústeres, consulte Redes de publicación/suscripción distribuidas.

Un clúster de publicación/suscripción de direccionamiento directo se comporta de este modo:

- Todos los gestores de colas tienen información sobre todos los otros gestores de colas.
- Todos los gestores de colas con suscripciones a los temas del clúster crean canales con todos los otros gestores del clúster y les informan acerca de sus suscripciones.
- Los mensajes publicados por una aplicación se direccionan de forma directa desde el gestor de colas al que está conectada a cada gestor de colas donde existe una suscripción coincidente.

El diagrama siguiente muestra un clúster de gestores de colas que no se utiliza actualmente para actividades de publicación/suscripción o punto a punto. Tenga en cuenta que cada gestor de colas del clúster solo se conecta a y desde los gestores de colas de repositorio completo.

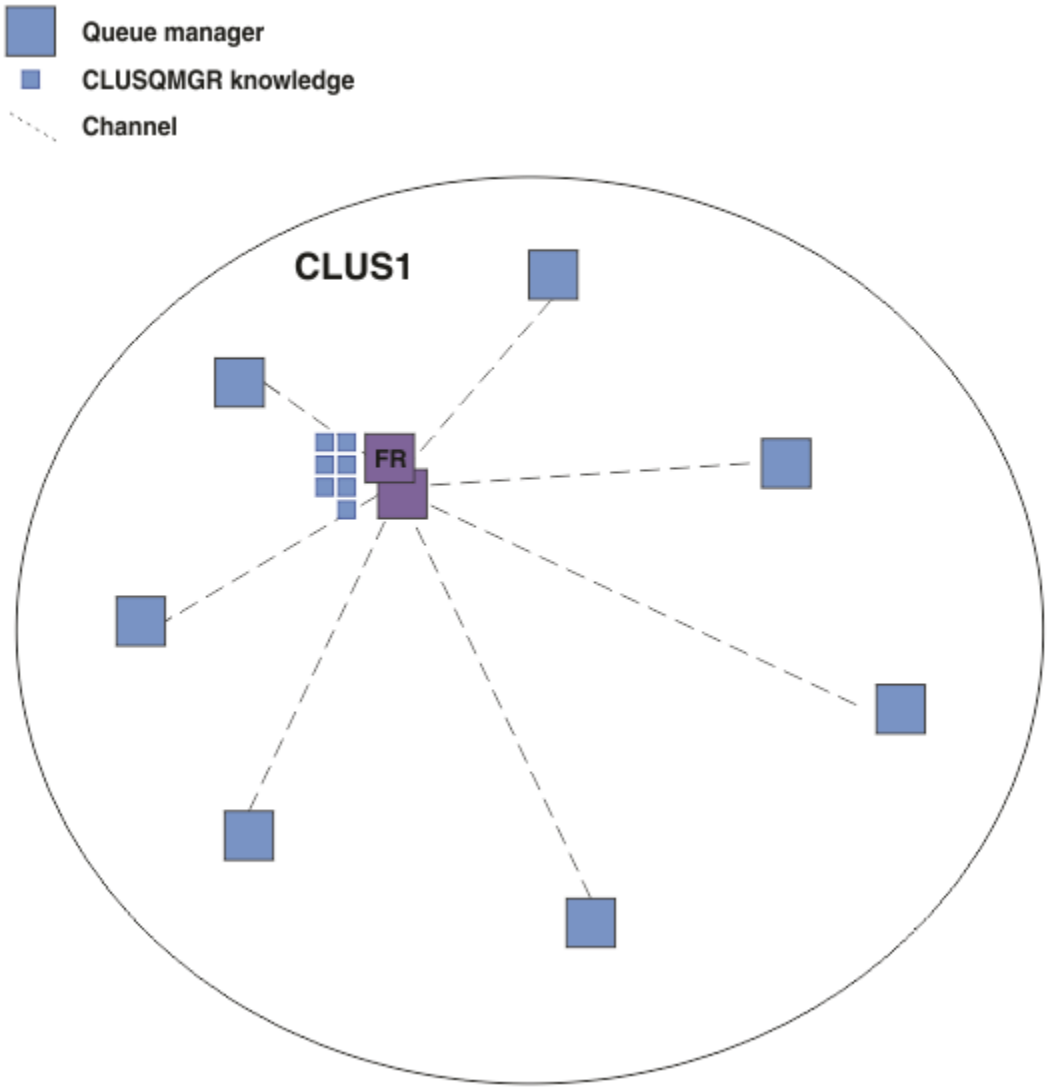


Figura 16. Un clúster de gestores de colas

Para que las publicaciones fluyan entre los gestores de colas de un clúster de direccionamiento directo, debe agrupar en un clúster una rama del árbol de temas como se describe en la sección Configurar un clúster de publicación/suscripción y especificar *direccionamiento directo*.

En un clúster de publicación/suscripción de direccionamiento directo, se define el objeto de tema en un gestor de colas específico del clúster. Cuando lo lleva a cabo, la información del objeto y de todos los otros gestores de colas del clúster se transfiere automáticamente a todos los gestores de colas del clúster mediante los gestores de colas de repositorio completo. Esto sucede antes de que cualquier gestor de colas haga referencia al tema:

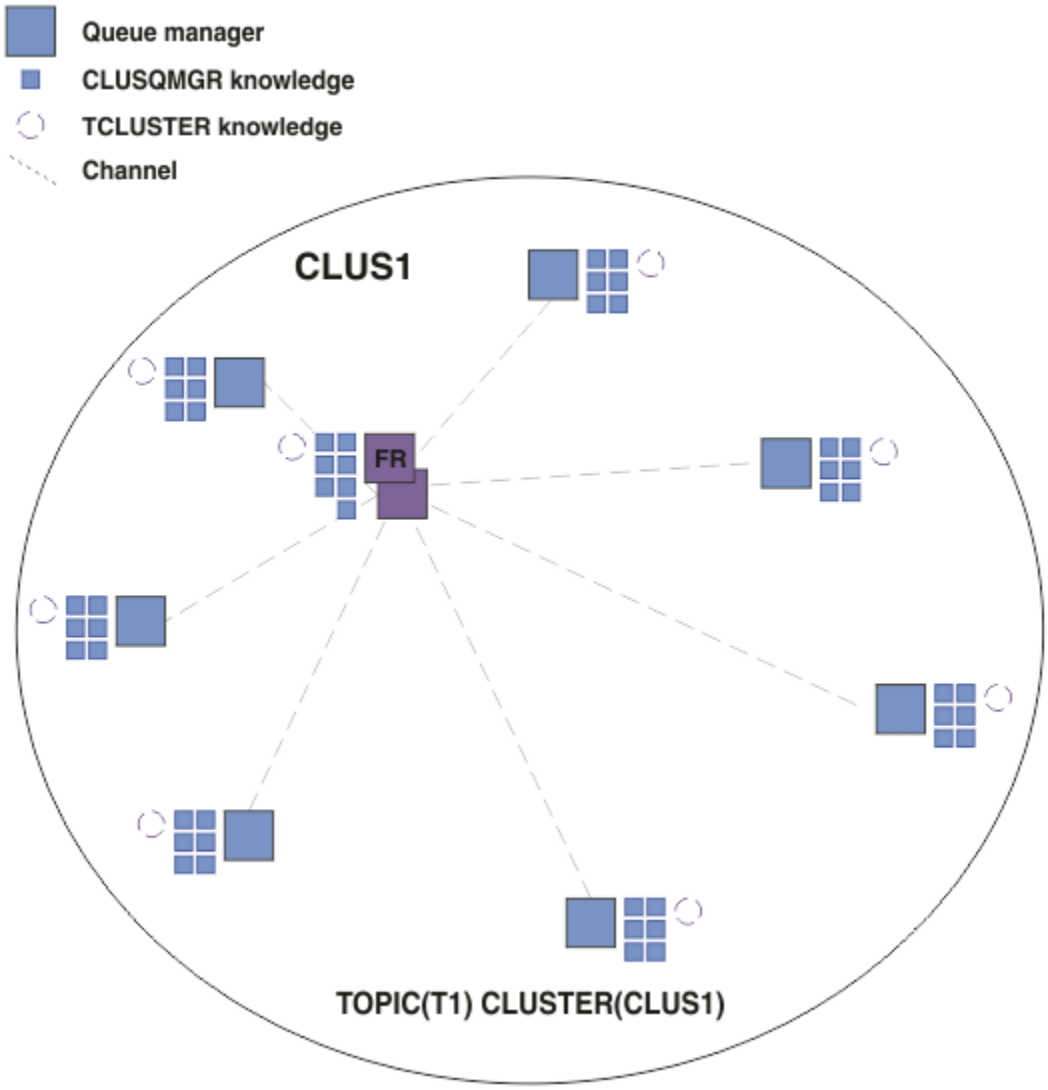


Figura 17. Un clúster de publicación/suscripción de direccionamiento directo

Cuando se crea una suscripción, el gestor de colas que aloja la suscripción establece un canal con cada gestor de colas del clúster y envía detalles de la suscripción. Este conocimiento de suscripción distribuida se representa mediante una suscripción de proxy en cada gestor de colas. Cuando se produce una publicación en cualquier gestor de colas del clúster que coincida con la serie de tema de la suscripción de proxy, se establece un canal de clúster desde el gestor de colas de publicador a cada gestor de colas que aloja una suscripción, y el mensaje se envía a cada uno de ellos.

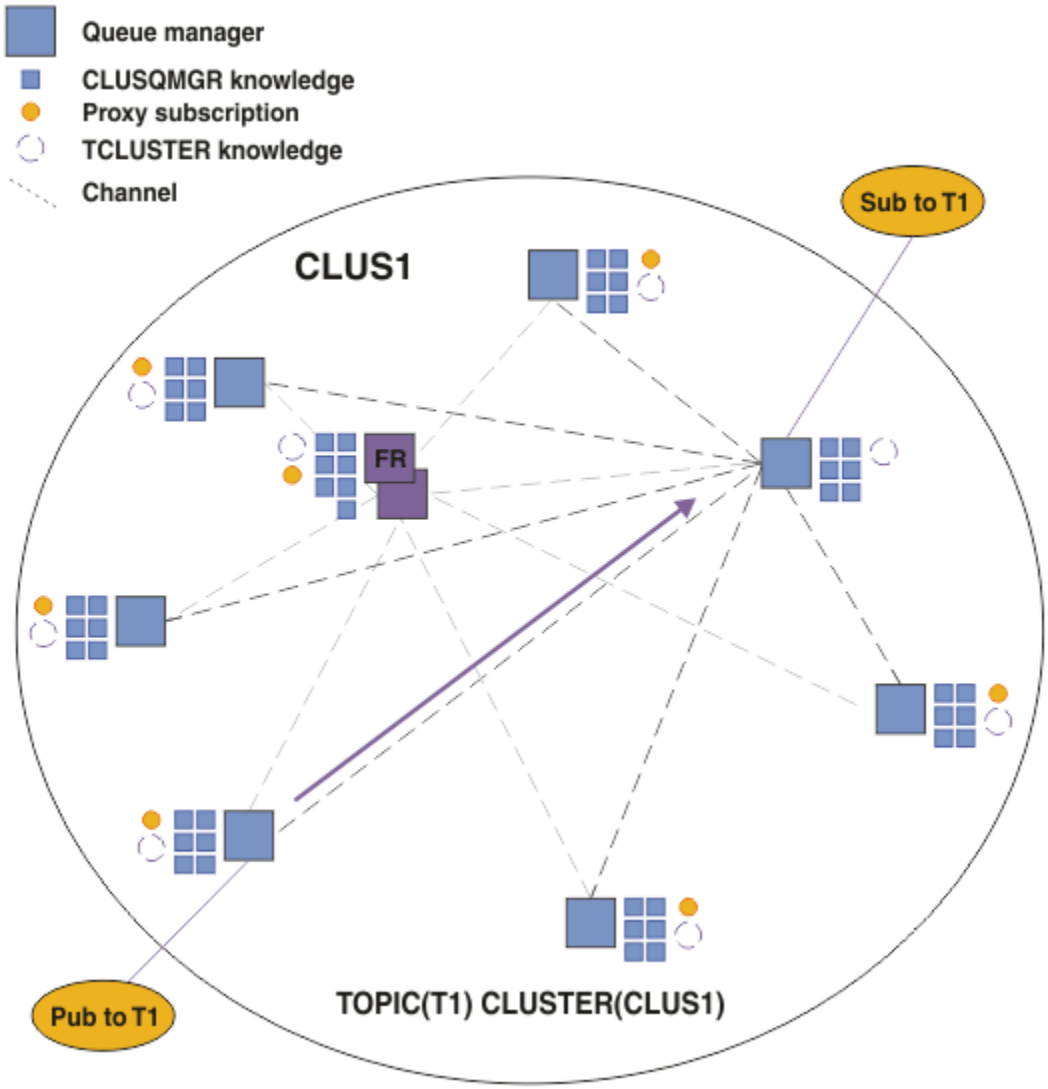


Figura 18. Un clúster de publicación/suscripción de direccionamiento directo con un publicador y suscriptor para un tema de clúster

El direccionamiento directo de las publicaciones a los gestores de colas que alojan las suscripciones simplifica la configuración y minimiza la latencia de la entrega de las publicaciones a las suscripciones.

Sin embargo, en función de la ubicación de las suscripciones y los publicadores, rápidamente su clúster puede pasar a estar totalmente interconectado, de modo que todo gestor de colas tendrá una conexión directa con todos los otros gestores de colas. Este puede ser o no aceptable en su entorno. Del mismo modo, si el conjunto de series de temas a los se suscriben cambia con frecuencia, la actividad general de propagar dicha información entre todos los gestores de colas también puede pasar a ser importante. Todos los gestores de colas de un clúster de publicación/suscripción direccionado directamente pueden dar cabida a esta actividad general adicional.

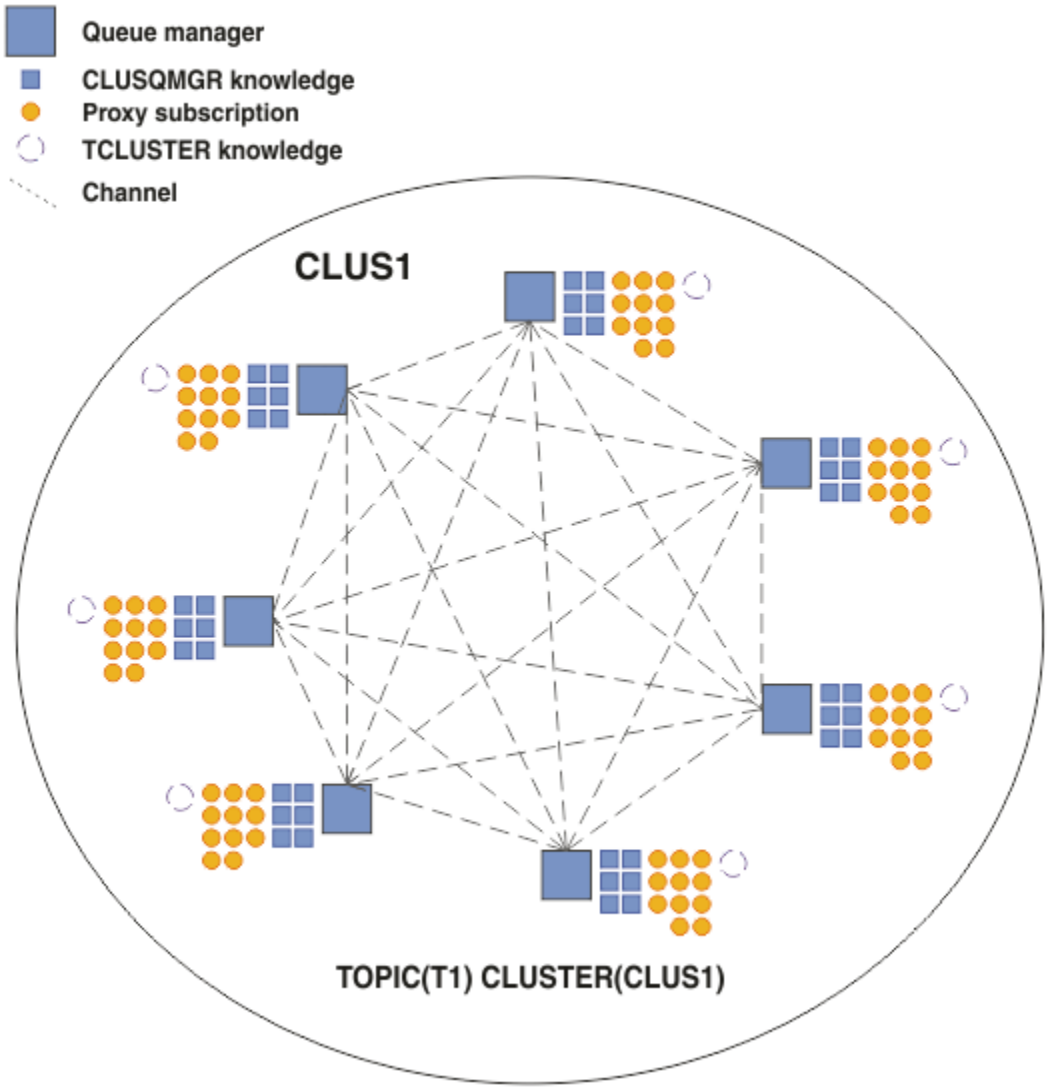


Figura 19. Un clúster de publicación/suscripción de direccionamiento directo totalmente interconectado

Resumen y consideraciones adicionales

La creación o administración de un clúster de publicación/suscripción de direccionamiento directo necesita una ligera intervención manual y proporciona un direccionamiento directo entre publicadores y suscriptores. Para determinadas configuraciones suele ser la topología más apropiada, especialmente en los clústeres con pocos gestores de colas o en aquellos donde resulta aceptable una alta conectividad de los gestores de colas y donde las suscripciones cambian con frecuencia. Sin embargo, también impone determinadas restricciones en su sistema:

- La carga en cada gestor de colas es proporcional al número total de gestores de colas del clúster. Por lo tanto, en los clústeres más grandes, los gestores de colas individuales y todo el sistema puede tener problemas de rendimiento.
- De forma predeterminada, todas las series de temas del clúster se propagan a través del clúster y las publicaciones solo se propagan a los gestores de colas remotos que tienen una suscripción con el tema asociado. Por lo tanto, los cambios rápidos en el conjunto de suscripciones pueden convertirse en un factor limitante. Puede cambiar este comportamiento predeterminado y, en su lugar, propagar todas las publicaciones a todos los gestores de colas, lo cual elimina la necesidad de las suscripciones de proxy. Esto disminuye el tráfico de conocimiento de las suscripciones pero es probable que aumente el tráfico de publicaciones y el número de canales que establece cada gestor de colas. Consulte Rendimiento de suscripción en redes de publicación/suscripción.

Nota: También se aplica una restricción similar a las jerarquías.

- Debido a la naturaleza interconectada de los gestores de colas de publicación/suscripción, se necesita tiempo para que las suscripciones de proxy se propaguen alrededor de todos los nodos de la red. No necesariamente, las publicaciones remotas comienzan con suscripciones de forma inmediata, por lo tanto, es posible que no se envíen publicaciones anticipadas tras una suscripción a una nueva serie de tema. Puede eliminar los problemas ocasionados por el retardo de la suscripción propagando todas las publicaciones a todos los gestores de colas, con lo cual se elimina la necesidad de suscripciones del proxy. Consulte Rendimiento de suscripción en redes de publicación/suscripción.

  Nota: También se aplica esta restricción a las jerarquías.

Antes de utilizar el direccionamiento directo, explore los métodos alternativos que se describen detalladamente en las secciones "Direccionamiento de host de tema en clústeres de publicación/suscripción" en la página 84 y "Direccionamiento en las jerarquías de publicación/suscripción" en la página 109.

Direccionamiento de host de tema en clústeres de publicación/suscripción

Las publicaciones de gestores de colas que no son de host en el clúster se direccionan a través del gestor de colas de host a cualquier gestor de colas del clúster con una suscripción coincidente.

Para obtener una introducción sobre cómo se direccionan los mensajes entre gestores de colas en jerarquías de publicación/suscripción y clústeres, consulte Redes de publicación/suscripción distribuidas.

Para comprender el comportamiento y las ventajas del direccionamiento de host de tema, se recomienda comprender la sección "Direccionamiento directo en clústeres de publicación/suscripción" en la página 79.

Un clúster de publicación/suscripción de direccionamiento de host de tema se comporta de este modo:

- Los objetos de temas administrados por el clúster se definen manualmente en los gestores de colas individuales del clúster. Se hace referencia a éstos como *gestores de colas de host de tema*.
- Cuando se realiza una suscripción en un gestor de colas del clúster, se crean los canales desde el gestor de colas del host de suscripción a los gestores de colas de host de tema y únicamente se crean las suscripciones de proxy en los gestores de colas que alojan el tema.
- Cuando una aplicación publica información para un tema, el gestor de colas conectado siempre reenvía la publicación a un gestor de colas que aloja el tema, el cual lo pasa a todos los gestores de colas del clúster que tengan suscripciones coincidentes para el tema.

Este proceso se describe detalladamente en los siguientes ejemplos.

Direccionamiento de host de tema utilizando un solo host de tema

Para que las publicaciones fluyan entre los gestores de colas de un clúster de direccionamiento de host de tema, debe agrupar en un clúster una rama del árbol de temas como se describe en la sección Configurar un clúster de publicación/suscripción y especificar *direccionamiento de host de tema*.

Hay diferentes motivos para definir un objeto de tema de direccionamiento de host de tema en varios gestores de colas de un clúster. Sin embargo, para simplificar comenzaremos por un solo host de tema.

El diagrama siguiente muestra un clúster de gestores de colas que no se utiliza actualmente para actividades de publicación/suscripción o punto a punto. Tenga en cuenta que cada gestor de colas del clúster solo se conecta a y desde los gestores de colas de repositorio completo.

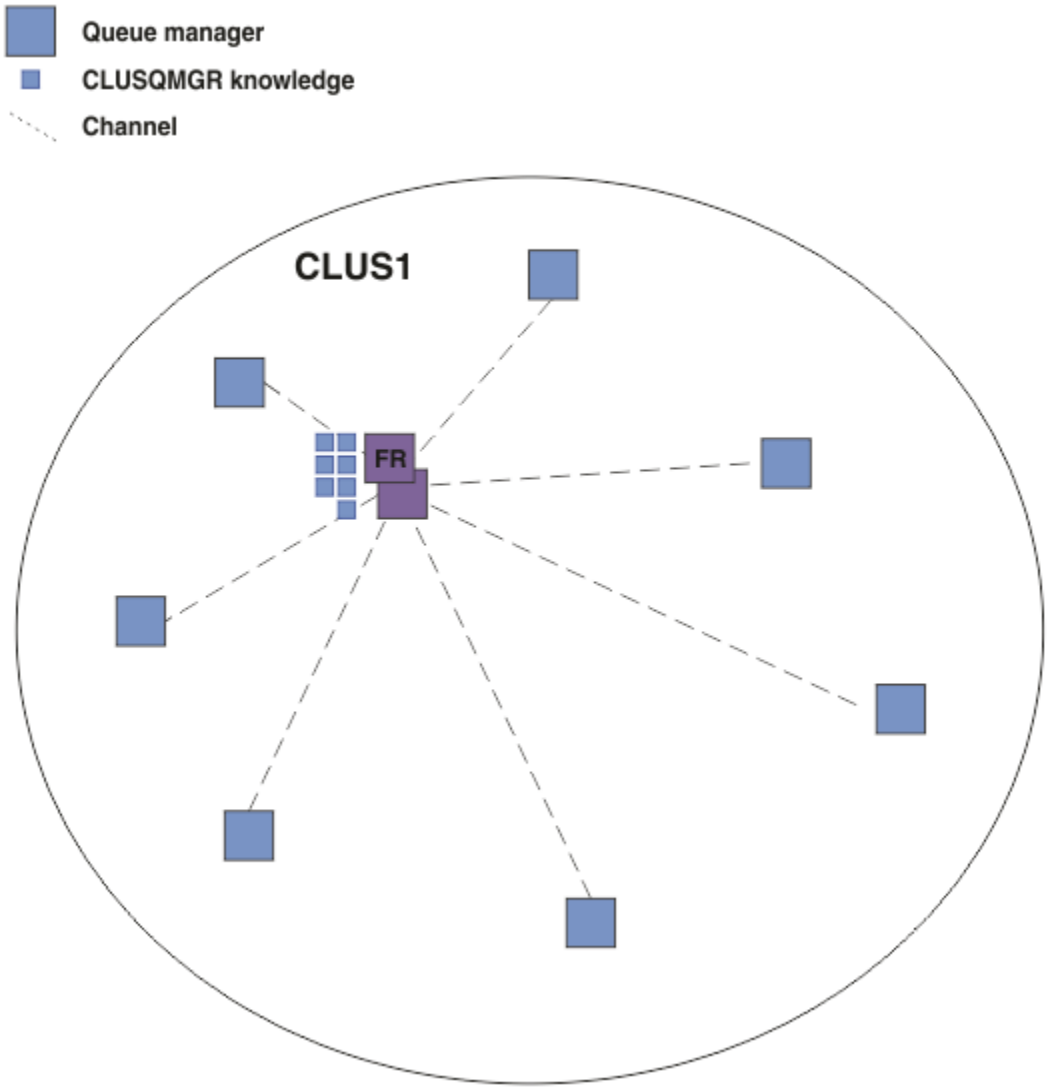


Figura 20. Un clúster de gestores de colas

En un clúster de publicación/suscripción de direccionamiento de host de tema, se define el objeto de tema en un gestor de colas específico del clúster. A continuación, el tráfico de publicación/suscripción fluye a través de dicho gestor de colas, lo que aumenta su carga de trabajo y lo convierte en un gestor de colas crítico para el clúster. Por estos motivos no se recomienda utilizar un gestor de colas de repositorio completo sino utilizar otro gestor de colas del clúster. Cuando define el objeto de tema en el gestor de colas del clúster, automáticamente los gestores de colas de repositorio completo pasan la información del objeto y su host a todos los demás gestores de colas del clúster. Tenga en cuenta que, a diferencia del *direccionamiento dirigido*, no se informa a cada gestor de colas sobre cada uno de los otros gestores de colas del clúster.

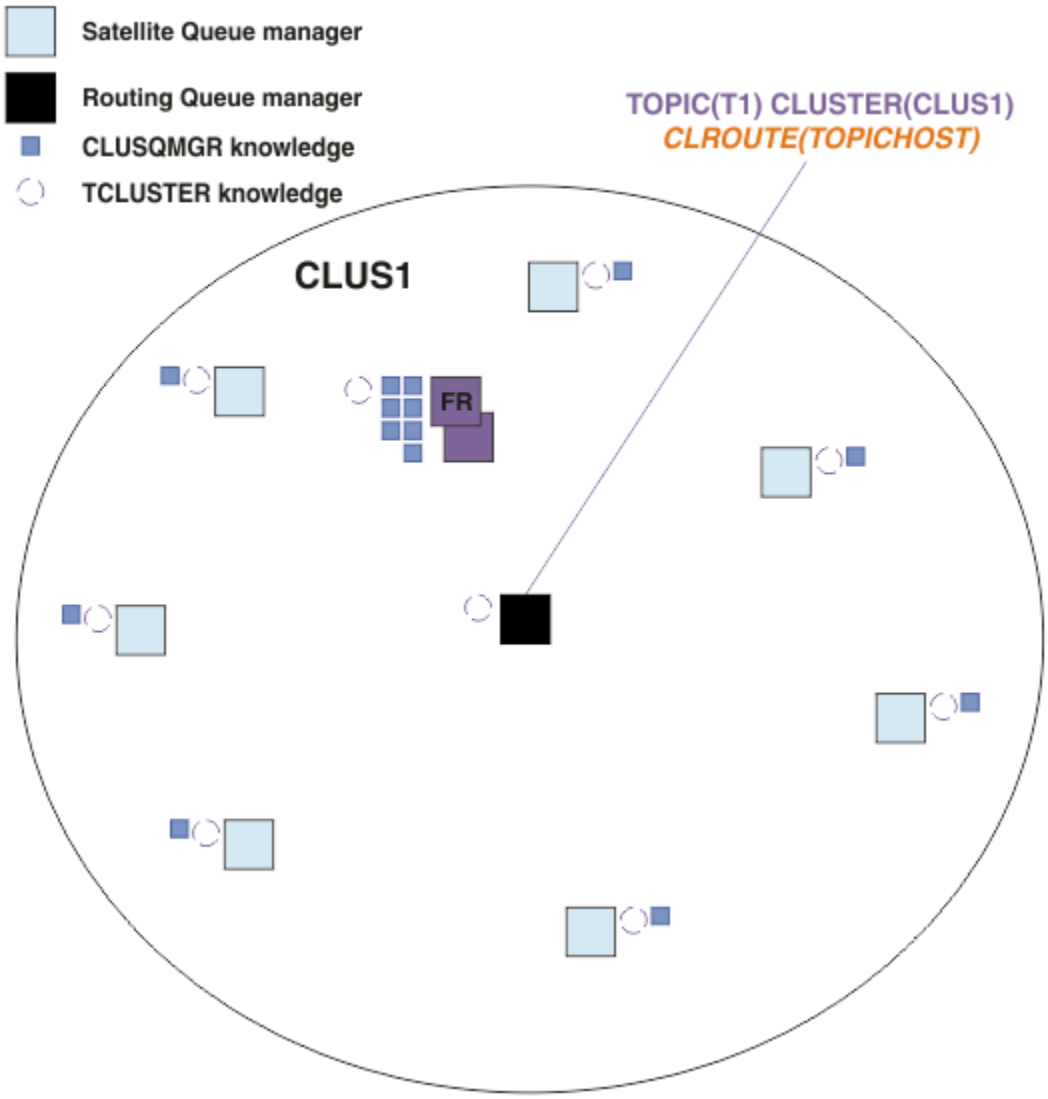


Figura 21. Un clúster de publicación/suscripción de direccionamiento de host de tema con un tema definido en un host de tema

Cuando se crea una suscripción en un gestor de colas, se crea un canal entre el gestor de colas de suscripción y el gestor de colas de host de tema. El gestor de colas de suscripción solo se conecta al gestor de colas del host de tema y envía detalles de la suscripción (con el formato de una *suscripción de proxy*). El gestor de colas de host de tema no reenvía esta información de suscripción a ningún otro gestor de colas adicional del clúster.

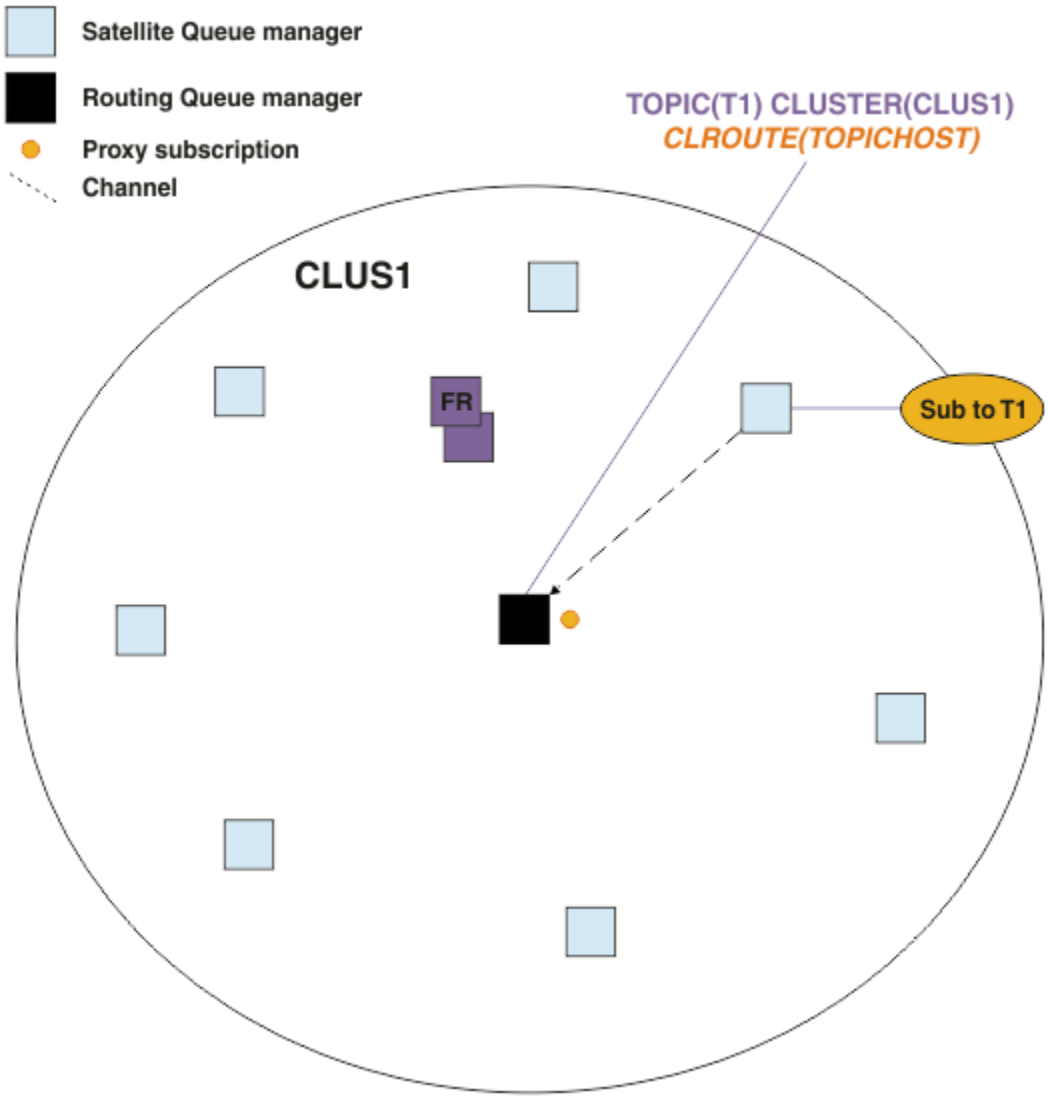


Figura 22. Un clúster de publicación/suscripción de direccionamiento de host de tema con un tema definido en un host de tema y un suscriptor

Cuando una aplicación de suscripción se conecta con otro gestor de colas y se publica un mensaje, se crea un canal entre el gestor de colas de publicación y el gestor de colas de host de tema y se reenvía el mensaje a dicho gestor de colas. El gestor de colas de publicación no tiene ninguna información acerca de las suscripciones de los otros gestores de colas del clúster, por lo tanto, se reenvía el mensaje al gestor de colas de host de tema, incluso si no existen suscriptores para dicho tema en el clúster. El gestor de colas de publicación solo se conecta al gestor de colas del host del tema. Las publicaciones se direccionan, a través del host de tema, a los gestores de colas de suscripción, si existen.

Las suscripciones que están en el mismo gestor de colas que el publicador se realizan directamente, sin enviar primero los mensajes a un gestor de colas de host de tema.

Tenga en cuenta que debido al rol crítico que desempeña cada gestor de colas de host de tema, debe elegir qué gestores de colas pueden manejar los requisitos de carga, disponibilidad y conectividad necesarios para el alojamiento de temas.

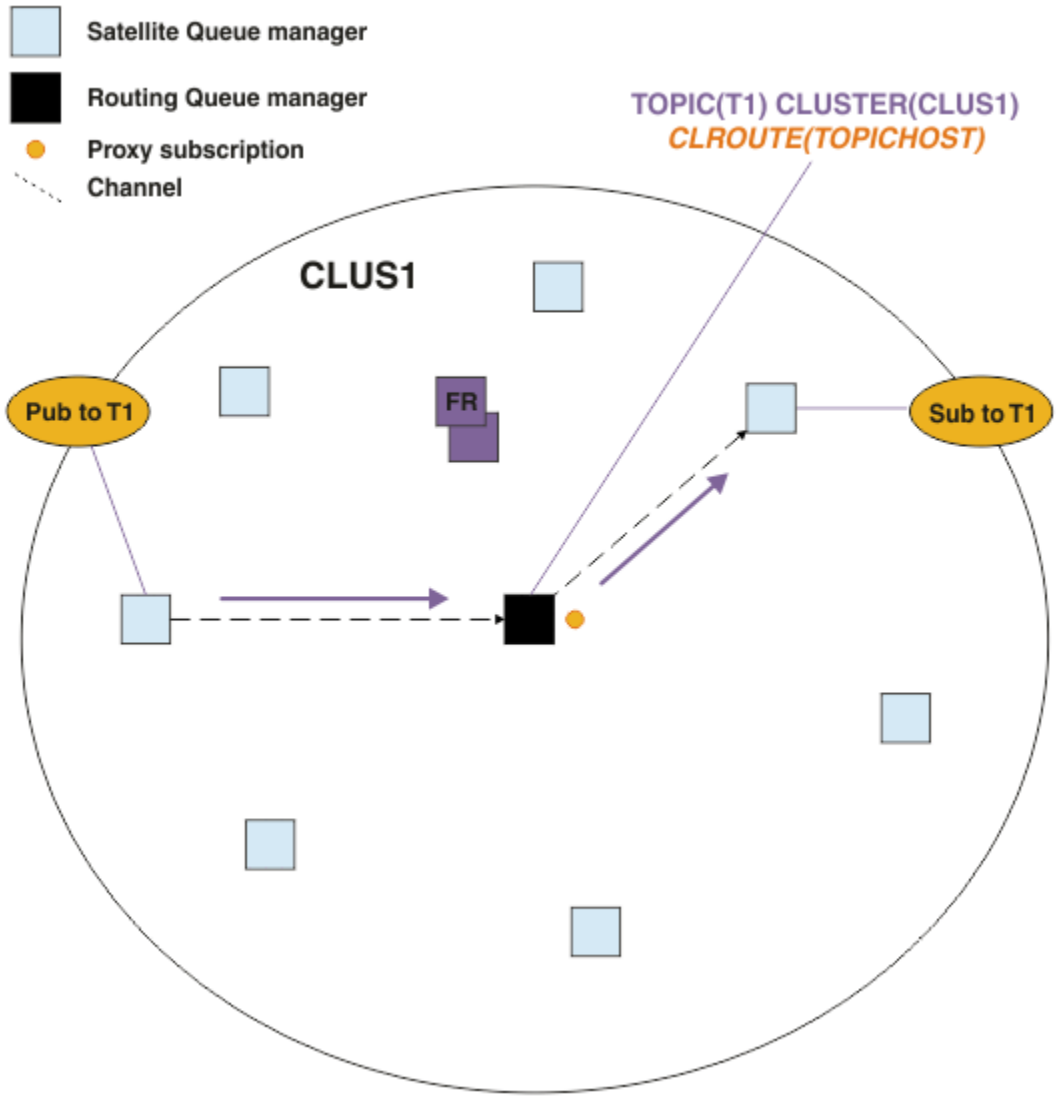


Figura 23. Un clúster de publicación/suscripción de direccionamiento de host de tema con un tema, un suscriptor y un publicador

División del árbol de temas entre varios gestores de colas

Un gestor de colas que aloja un tema direccionado solo es responsable de la información de las suscripciones y de la publicación de mensajes relacionados con la rama del árbol de temas para la que se ha configurado su objeto de tema administrado. Si las diferentes aplicaciones de publicación/suscripción del clúster utilizan temas diferentes, puede configurar diferentes gestores de colas para que alojen las diferentes ramas del clúster del árbol de temas. Esto permite el escalado disminuyendo el tráfico de publicaciones, la información de las suscripciones y los canales en cada gestor de colas de host de tema del clúster. Debe utilizar este método para diferenciar las ramas del árbol de temas que tienen un volumen elevado.

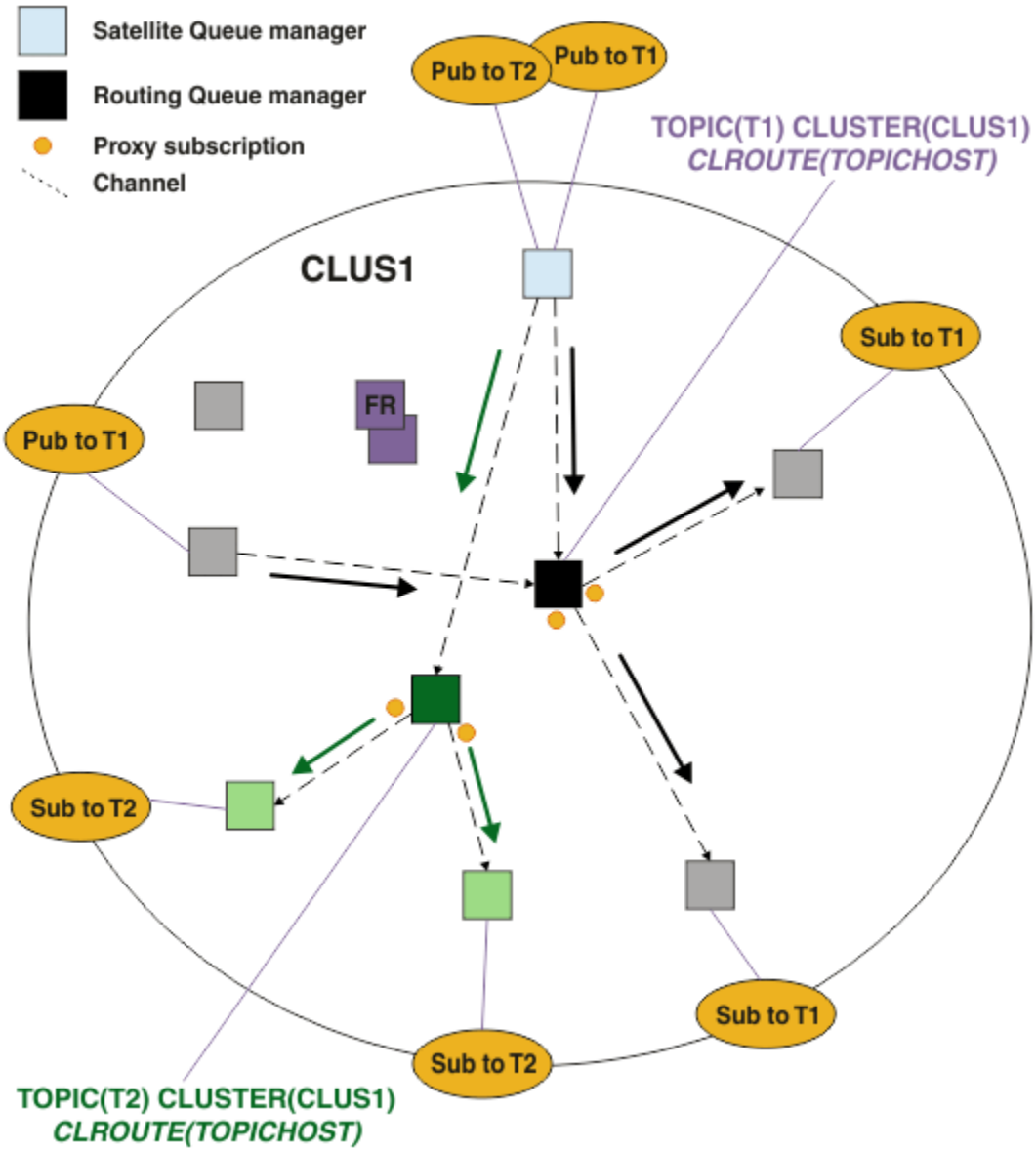


Figura 24. Un clúster de publicación/suscripción de direccionamiento de host de tema con dos temas, cada uno de ellos definido en un host de tema

Por ejemplo, utilizando los temas descritos en Árboles de temas, si el tema T1 se ha configurado con una serie de tema de `/USA/Alabama` y el tema T2 se ha configurado con una serie de tema de `/USA/Alaska`, un mensaje publicado en `/USA/Alabama/Mobile` se direccionará a través del gestor de colas que aloja T1 y un mensaje publicado en `/USA/Alaska/Juneau` se direccionará a través del gestor de colas que aloja T2.

Nota: No puede hacer que una sola suscripción abarque varias ramas agrupadas en clúster del árbol de temas utilizando un comodín que se encuentra en una posición más alta del árbol de temas que los puntos agrupados en clúster. Consulte Suscripciones de comodín.

Direccionamiento de host de tema mediante varios hosts de temas para un solo tema

Si un solo gestor de colas es responsable del direccionamiento de un tema y dicho gestor de colas no está disponible o no puede manejar la carga de trabajo, las publicaciones no fluirán puntualmente hacia las suscripciones.

Si necesita una mayor flexibilidad, escalabilidad y equilibrio de la carga de trabajo de los que obtiene cuando define un tema en un único gestor de colas, puede definir un tema en más de un gestor de colas. Cada mensaje individual publicado se direcciona a través de un único host de tema. Cuando existen varias definiciones de host de tema coincidente, se selecciona uno de los hosts de tema. La selección se realiza del mismo modo que para las colas de clúster. Esto permite direccionar los mensajes a los hosts de temas disponibles, evitando así los que no están disponibles, y permite equilibrar la carga de mensajes entre varios gestores de colas de host de tema y canales. Sin embargo, no se mantiene el orden entre los diferentes mensajes si se utilizan varios hosts de temas para el mismo tema en el clúster.

El siguiente diagrama muestra un clúster de direccionamiento de host de tema en el que se ha definido el mismo tema en dos gestores de colas. En este ejemplo, los gestores de colas de suscripción envían información acerca del tema suscrito a los dos gestores de colas de host de tema con el formato de una suscripción de proxy:

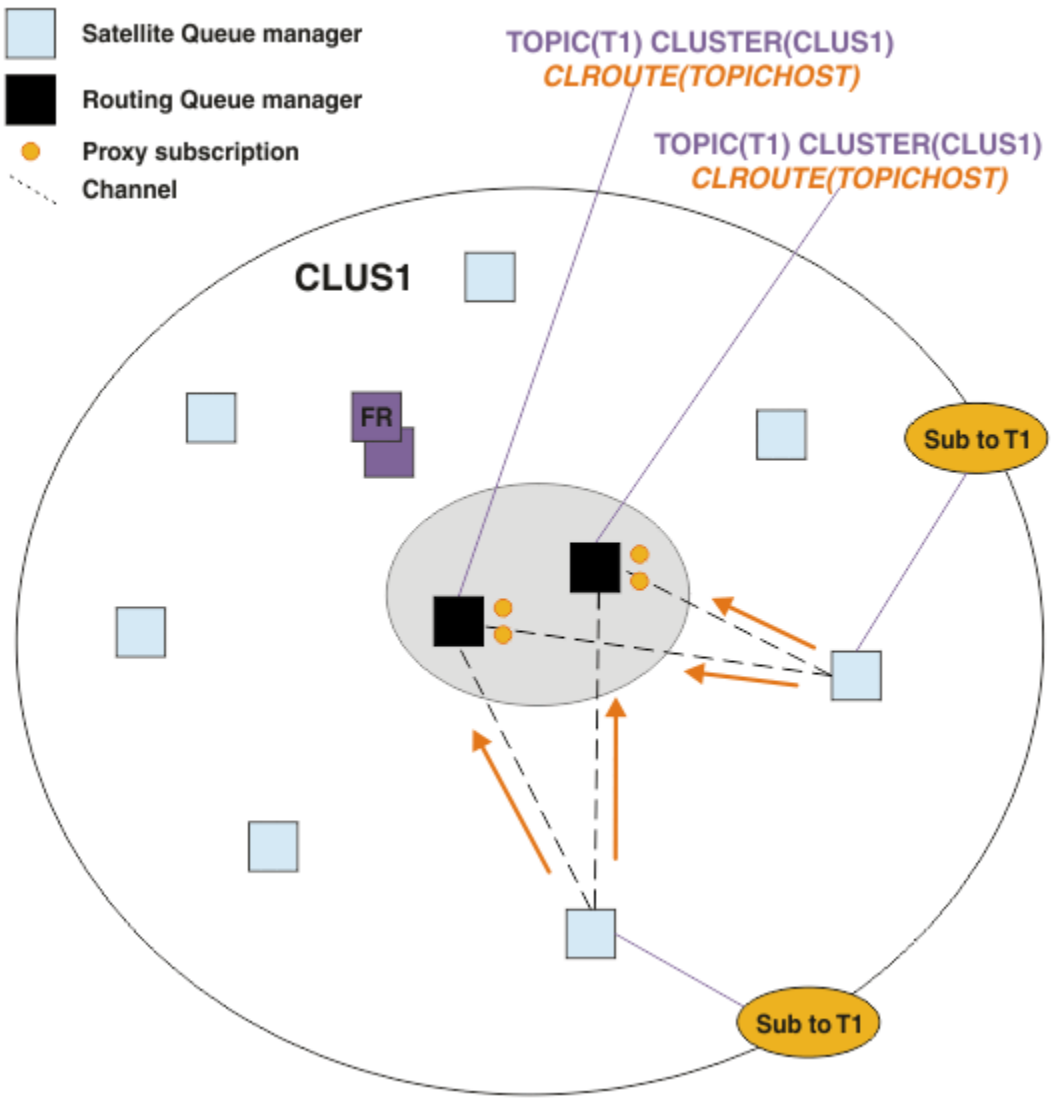


Figura 25. Creación de suscripciones de proxy en un clúster de publicación/suscripción de varios hosts de temas

Cuando se realiza una publicación en un gestor de colas que no es de host, el gestor de colas envía una copia de la publicación a *uno* de los gestores de colas de host de tema para dicho tema. El sistema selecciona el host en función del comportamiento predeterminado del algoritmo de gestión de la carga de trabajo del clúster. En un sistema típico, esto es similar a una distribución rotativa entre cada gestor de colas de host de tema. No existe ninguna afinidad entre los mensajes procedentes de la misma aplicación de publicación; esto es similar a utilizar un enlace de clúster de tipo `NOTFIXED`.

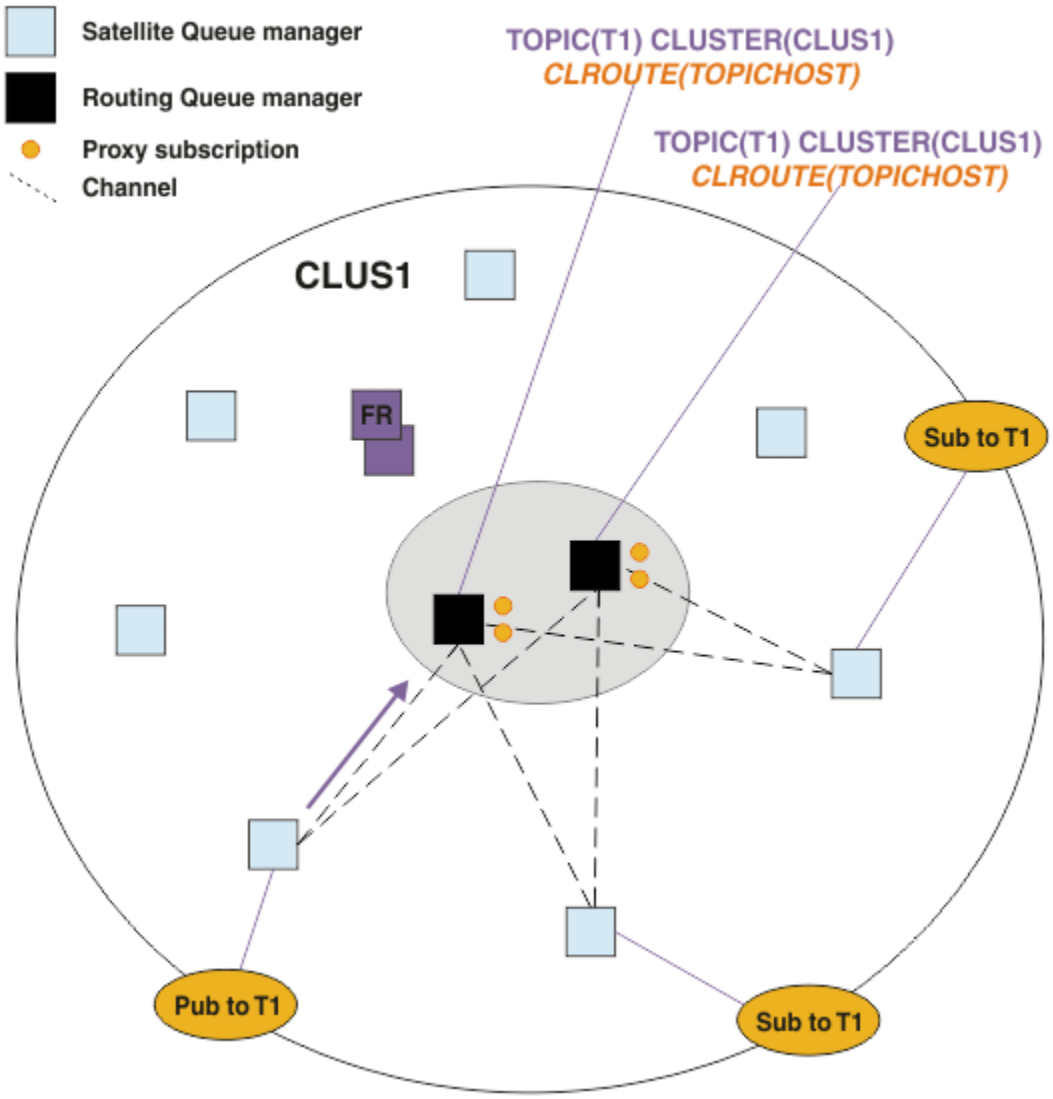


Figura 26. Creación de publicaciones en un clúster de publicación/suscripción de varios hosts de temas

Las publicaciones de entrada para el gestor de colas de host de tema se reenvían, a continuación, a todos los gestores de colas que tengan registrada una suscripción de proxy coincidente:

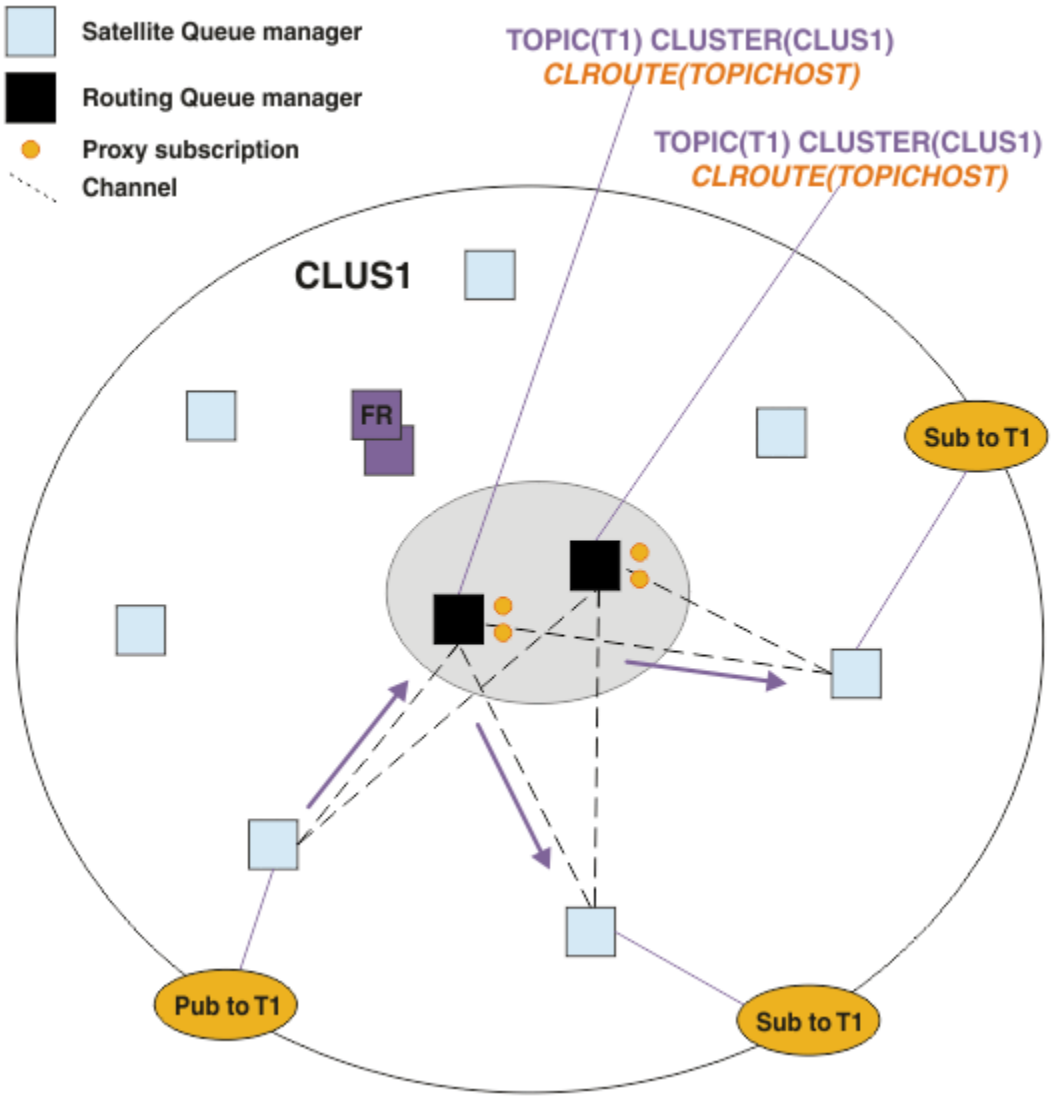


Figura 27. Direccionamiento de publicaciones a suscriptores en un clúster de publicación/suscripción de varios hosts de temas

Cómo hacer que los suscriptores y publicadores sean locales para un gestor de colas de host de tema

Los ejemplos anteriores muestran el direccionamiento entre los publicadores y suscriptores en los gestores de colas que no alojan objetos de tema de direccionamiento administrado. En estas topologías, los mensajes requieren varios *saltos* para alcanzar las suscripciones.

Cuando no se desea el salto adicional, puede resultar adecuado conectar los publicadores clave a los gestores de colas que alojan el tema. Sin embargo, si hay varios hosts de temas para un tema y un único publicador, todo el tráfico de publicación se direccionará a través del gestor de colas de host de tema al que está conectado el publicador.

Del mismo modo, si existen suscripciones clave, éstas deben estar ubicadas en un gestor de colas de host de tema. Sin embargo, si hay varios hosts del tema direccionado, solo una proporción de las publicaciones evitará el salto adicional y el resto se direccionará, en primer lugar, a través de los otros gestores de colas de host de tema.

Puede encontrar información adicional sobre las topologías descritas en la sección: Direccionamiento de host de tema mediante publicadores o suscriptores centralizados.

Nota: Es necesario realizar una planificación especial si se cambia la configuración cuando se coubican los publicadores o suscriptores con los hosts de temas direccionados. Por ejemplo, consulte la sección Adición de hosts de temas adicionales a un clúster de direccionamiento de host de tema.

Resumen y consideraciones adicionales

Un clúster de publicación/suscripción de direccionamiento de host de tema le proporciona un control preciso sobre qué gestores de colas alojan cada tema, y dichos gestores de colas se convierten en los gestores de colas de *direccionamiento* de dicha rama del árbol de temas. Adicionalmente, los gestores de colas sin suscripciones o publicadores no necesitan conectarse a los gestores de colas de host de tema y los gestores de colas con suscripciones no necesitan conectarse a los gestores de colas que no alojan un tema. Esta configuración puede reducir de forma importante el número de conexiones entre los gestores de colas del clúster y la cantidad de información que se está pasando entre los gestores de colas. Esto resulta especialmente cierto en los clústeres de gran tamaño en los que solo un subconjunto de gestores de colas realizan el trabajo de publicación/suscripción. Esta configuración también proporciona algún control sobre la carga de los gestores de colas individuales del clúster, de modo que, por ejemplo, puede optar por alojar los temas más activos en sistemas más potentes y flexibles. En determinadas configuraciones, en especial en los clústeres de gran tamaño, suele ser una topología más adecuada que el *direccionamiento directo*.

Sin embargo, el direccionamiento de host de tema también impone determinadas restricciones en el sistema:

- La configuración y el mantenimiento del sistema requieren más planificación que para el direccionamiento directo. Tiene que decidir qué apunta al clúster en el árbol de temas y la ubicación de las definiciones de tema en el clúster.
- Al igual que para los temas direccionados directamente, cuando se define un tema direccionado de host de tema nuevo, la información se envía a los gestores de colas de depósito completo y desde allí directamente a todos los miembros del clúster. Este suceso hace que se inicien canales en cada miembro del clúster desde los depósitos completos si aún no se han iniciado.
- Las publicaciones siempre se envían a un gestor de colas de host desde un gestor de colas no de host, incluso si no existen suscripciones en el clúster. Por lo tanto, debe utilizar los temas direccionados cuando normalmente se espera que existan suscripciones o cuando la sobrecarga de conectividad y conocimiento globales es mayor que el riesgo de tráfico de publicación adicional.

  Nota: Como se ha descrito anteriormente, este riesgo se puede mitigar si los publicadores se convierten en locales para un host de tema.
- Los mensajes que se publican en los gestores de colas no de host no van directamente al gestor de colas que aloja la suscripción, siempre se direccionan a través de un gestor de colas de host de tema. Este enfoque puede aumentar la actividad general total en el clúster, aumentar la latencia de mensajes y disminuir el rendimiento.

  Nota: Como se ha descrito anteriormente, este riesgo se puede mitigar si suscripciones o los publicadores se convierten en locales para un host de tema.
- La utilización de un gestor de colas de host de tema único presenta un punto único de anomalía para todos los mensajes que se publican en un tema. Puede eliminar este único punto de anomalía definiendo varios hosts de temas. Sin embargo, tener varios hosts afecta el orden de los mensajes publicados recibidos por las suscripciones.
- Los gestores de colas de host de tema causan una carga de mensajes adicional, porque es necesario que procesen el tráfico de publicación de varios gestores de colas. Esta carga se puede reducir: Utilice varios hosts de temas para un solo tema (en cuyo caso no se mantiene el orden de mensajes) o utilice gestores de colas diferentes para alojar los temas direccionados para diferentes ramas del árbol de temas.

Antes de utilizar el direccionamiento de host de tema, explore los métodos alternativos que se describen detalladamente en las secciones “Direccionamiento directo en clústeres de publicación/suscripción” en la página 79 y “Direccionamiento en las jerarquías de publicación/suscripción” en la página 109.

Agrupación en clúster de la publicación/suscripción: Mejoras prácticas

El uso de temas de clúster facilita la ampliación del dominio de publicación/suscripción entre gestores de colas, pero puede provocar problemas si los mecanismos y las implicaciones no se comprenden en su totalidad. Existen dos modelos para compartir información y direccionar la publicación. Implementar el modelo que mejor se ajuste a sus necesidades empresariales individuales y cuyo rendimiento sea el mejor en el clúster seleccionado.

La información sobre métodos recomendados de las secciones siguientes no proporciona una solución que se ajuste a todos los tamaños, sino que comparte los métodos comunes para la resolución de problemas. Supone que tiene una comprensión básica de los clústeres de IBM MQ y de la mensajería de publicación/suscripción y que está familiarizado con la información de Redes de publicación/suscripción distribuidas y “Diseño de clústeres de publicación/suscripción” en la página 77.

Cuando utiliza un clúster para la mensajería punto a punto, cada gestor de colas del clúster funciona solo cuando la información es necesaria. Esto es, solo busca información de otros recursos del clúster, por ejemplo, otros gestores de clústeres del clúster y colas de clúster, cuando las aplicaciones que se conectan a los mismos los utilizan. Cuando añade mensajería de suscripción/publicación a un clúster, aumenta el nivel en que se comparte la información y la conectividad entre los gestores de colas del clúster. Para poder seguir los métodos recomendados para los clústeres de publicación/suscripción, es necesario comprender todas las implicaciones de este cambio de comportamiento.

Para que pueda crear la mejor arquitectura, basada exactamente en sus necesidades, existen dos modelos para compartir la información y direccionar la publicación en los clústeres de publicación/suscripciones: *direccionamiento directo* y *direccionamiento al host de tema*. Para poder realizar la mejor selección, debe comprender ambos modelos y los diferentes requisitos que satisface cada modelo. Estos requisitos se describen en las secciones siguientes junto con “Planificación de su red de publicación/suscripción distribuida” en la página 73:

- “Motivos para limitar el número de gestores de colas del clúster implicados en la actividad de publicación/suscripción” en la página 94
- “Cómo decidir qué temas se han de incluir en el clúster” en la página 95
- “Cómo medir el sistema” en la página 95
- “Ubicación del publicador y la suscripción” en la página 96
- “Tráfico de publicaciones” en la página 97
- “Cambio de suscripción y series de temas dinámicos” en la página 97

Motivos para limitar el número de gestores de colas del clúster implicados en la actividad de publicación/suscripción

Se ha de considerar la capacidad y el rendimiento cuando se utiliza la mensajería de publicación/suscripción en un clúster. Por lo tanto, se recomienda estudiar detenidamente la necesidad de la actividad de publicación/suscripción entre los gestores de colas y limitarla únicamente al número de gestores de colas que la requieren. Una vez identificado el conjunto de gestores de colas mínimo que necesitan la publicación y suscripción a los temas, éstos pueden convertirse en miembros de un clúster que solo contenga dichos gestores de colas y ningún otro gestor de colas.

Este método resulta especialmente útil si ya ha establecido un clúster que funciona bien para la mensajería punto a punto. Cuando convierte un clúster existente de gran tamaño en un clúster de publicación/suscripción, el método recomendado es crear inicialmente un clúster separado para el trabajo de publicación/suscripción donde se puedan probar las aplicaciones, en lugar de utilizar el clúster actual. Puede utilizar un subconjunto de gestores de colas que estén en uno o varios clústeres punto a punto y convertir a los componentes de este subconjunto en miembros del nuevo clúster de publicación/suscripción. Sin embargo, el repositorio completo de los gestores de colas de su nuevo clúster no deben ser miembros de ningún otro clúster; de este modo, se aísla la carga adiciona de los repositorios completos de clústeres existentes.

Si no puede crear un clúster nuevo y tiene que convertir un clúster existente de gran tamaño en un clúster de publicación/suscripción, no utilice un modelo de direccionamiento directo. Normalmente, el

modelo de host direccionado funciona mejor en clústeres de gran tamaño, ya que generalmente restringe la compartición de la información de publicación/suscripción y de la conectividad al conjunto de gestores de colas que realizan activamente el trabajo de publicación/suscripción, concentrándose en los gestores de colas que alojan los temas. La excepción a ello es que si se invoca una renovación manual de la información de suscripción en un gestor de colas que aloja una definición de tema, el gestor de colas que aloja el tema conectará con cada uno de los gestores de colas del clúster. Consulte Resincronización de suscripciones de proxy.

Si establece que un clúster no se debe utilizar para la publicación/suscripción debido a su tamaño o a su carga actual, el método recomendado es impedir que el clúster se convierta en un clúster de publicación/suscripción de forma imprevista. Utilice la propiedad del gestor de colas **PSCLUS** para impedir que nadie añada un tema de clúster a ningún gestor de colas del clúster. Consulte “Inhabilitación de la publicación/suscripción en un clúster” en la página 105.

Cómo decidir qué temas se han de incluir en el clúster

Es importante elegir cuidadosamente los temas que se añaden al clúster: cuanto más alto estén los temas en el árbol de temas, más extendido será su uso. Esto puede provocar que se propague más información de suscripciones y publicaciones de la necesaria. En el caso de que haya muchas y diferentes ramas del árbol de temas y algunas se deban agrupar en clúster y otras no, cree objetos de temas administrados en la raíz de cada rama que se deba agrupar en clúster y añádalos al clúster. Por ejemplo, si las ramas `/A`, `/B` y `/C` se deben agrupar en clúster, defina objetos de temas de clúster diferentes para cada rama.

Nota: El sistema le impide que anide las definiciones de temas de clúster en el árbol de temas. Solo tiene permiso para los temas de clúster en un punto del árbol de temas de cada subrama. Por ejemplo, no puede definir objetos de tema en clúster para `/A` y para `/A/B`. La anidación de temas en clúster puede llevar a confusión sobre qué objeto de clúster se aplica a qué suscripción, especialmente cuando las suscripciones utilizan comodines. Esto es todavía más importante cuando se utiliza el direccionamiento de host de temas, en el que las decisiones de direccionamiento están definidas de forma precisa en función de su asignación a los hosts de temas.

Si se deben añadir temas de clúster en una posición más alta del árbol de temas, pero algunas ramas del árbol por debajo del punto de clúster no requieren el comportamiento de clúster, puede utilizar los atributos de ámbito de suscripción y publicación para disminuir el nivel de compartición de la suscripción y publicación para temas adicionales.

No debe colocar el nodo raíz del tema en el clúster sin tener en cuenta el comportamiento visto. Haga que los temas globales sean lo más obvios posibles, por ejemplo, utilizando un cualificador de alto nivel en la serie del tema: `/global` o `/cluster`.

Existe un motivo para no desear que el nodo de tema raíz esté en el clúster. Esto es debido a que cada gestor de colas tiene una definición local para el nodo raíz, el objeto de tema `SYSTEM.BASE.TOPIC`. Cuando este objeto se agrupan en clúster en un gestor de colas del clúster, todos los otros gestores de colas lo reconocen. Sin embargo, cuando existe una definición local del mismo objeto, sus propiedades alteran temporalmente el objeto de clúster. Esto da como resultado que los gestores de colas actúen como si el tema no estuviera agrupado en clúster. Para resolver esta situación, debe agrupar en clúster cada definición de `SYSTEM.BASE.TOPIC`. Puede hacerlo para las definiciones de direccionamiento directo pero no para las definiciones direccionadas de host de temas, ya que esto convierte al gestor de colas en un host de temas.

Cómo medir el sistema

Generalmente, los clústeres de publicación/suscripción dan como resultado un patrón diferentes de canales de clúster a mensajería punto a punto en un clúster. El modelo punto a punto es un modelo 'opt in' pero, por naturaleza, los clústeres de publicación/suscripción no son tan discriminatorios en lo referente a la suscripción distribuida, sobretodo cuando se utilizan temas de direccionamiento directo. Por lo tanto, es importante identificar qué gestores de colas de un clúster de publicación/suscripción utilizarán canales de clúster para conectarse a otros gestores de colas y bajo qué circunstancias.

La tabla siguiente lista el conjunto típico de canales de emisor y receptor del clúster previsto para cada gestor de colas de un clúster de publicación/suscripción durante una ejecución normal, en función del rol del gestor de colas en el clúster de publicación/suscripción.

Tabla 5. Canales de emisor y receptor del clúster para cada método de direccionamiento.

Rol de gestor de colas	Receptores de clúster directos	Emisores de clúster directos	Receptores de clúster de temas	Emisores de clúster de temas
Depósito completo	AllQmgrs	AllQmgrs	AllQmgrs	AllQMgrs
Host de definición de tema	n/d	n/d	AllSubs+AllPubs (1)	AllSubs (1)
Suscripciones creadas	AllPubs (1)	AllQMgrs	AllHosts	AllHosts
Publicadores conectados	AllSubs (1)	AllSubs (1)	AllHosts	AllHosts
Sin publicadores ni suscriptores	AllSubs (1)	None (1)	None (2)	None (2)

Clave:

AllQmgrs
Un canal a y desde cada gestor de colas del clúster.

AllSubs
Un canal a y desde cada gestor de colas del clúster en el que se ha creado una suscripción.

AllPubs
Un canal a y desde cada gestor de colas del clúster en el que se ha conectado una aplicación de publicación.

AllHosts
Un canal a y desde cada gestor de colas del clúster en el que se ha configurado una definición del objeto de tema de clúster.

Ninguna
Ningún canal a o desde otros gestores de colas del clúster con la única finalidad de mensajería de publicación/suscripción.

Notas:

1. Si se realiza una renovación de las suscripciones del proxy del gestor de colas desde este gestor de colas, es posible que se cree automáticamente una canal a y desde todos los otros gestores de colas del clúster.
2. Si se realiza una renovación de las suscripciones del proxy del gestor de colas desde este gestor de colas, es posible que se cree automáticamente una canal a y desde cualquier otro gestor de colas del clúster que aloje una definición de un tema de clúster.

La tabla anterior muestra que normalmente el direccionamiento del host de temas utiliza un número mucho menor de canales de emisor y receptor que el direccionamiento directo. Si la conectividad del canal se ha de tener en cuanta en determinados gestores de colas de un clúster, por motivos de capacidad o para poder establecer determinados canales (por ejemplo, a través de cortafuegos), la solución preferida sería, por lo tanto, el direccionamiento de host de temas.

Ubicación del publicador y la suscripción

La publicación/suscripción en clúster permite que los mensajes publicados en un gestor de colas se entreguen a las suscripciones en cualquier otro gestor de colas del clúster. Por lo que respecta a la mensajería punto a punto, el coste de transmitir mensajes entre los gestores de colas puede ir en detrimento del rendimiento. Por lo tanto, siempre que sea posible, se debe intentar crear suscripciones a temas en los mismos gestores de colas en los que se publican los mensajes.

Cuando se utiliza el direccionamiento de hosts de temas dentro de un clúster, es importante tener en cuenta también la ubicación de las suscripciones y los publicadores en relación con los gestores de colas que alojan los temas. Cuando el publicador no está conectado a un gestor de colas que sea un host del tema de clúster, los mensajes publicados se envían siempre a un gestor de colas que aloja temas. Del mismo modo, cuando se crea una suscripción a un gestor de colas que no es un host de temas para un tema de clúster, los mensajes publicados desde otros gestores de colas del clúster siempre se envían, en primer lugar, a un gestor de colas que aloja temas. Más específicamente, si la suscripción se encuentra en un gestor de colas que aloja el tema, pero existen uno o varios gestores de colas que también alojan el mismo tema, una proporción de las publicaciones de los otros gestores de colas se direccionan a través de estos otros gestores de colas que alojan temas. Consulte Direccionamiento de hosts de temas utilizando publicadores o suscriptores centralizados para obtener más información sobre cómo diseñar un clúster de publicación/suscripción con direccionamiento de host de temas para minimizar la distancia entre publicadores y suscripciones.

Tráfico de publicaciones

Los mensajes que publica una aplicación conectada a un gestor de colas de un clúster se transmiten a las suscripciones de otros gestores de colas utilizando los canales de emisor.

Cuando utiliza el direccionamiento directo, los mensajes publicados toma la ruta más corta entre los gestores de colas. Esto es, pasan directamente desde el gestor de colas de publicación a cada uno de los gestores de colas con suscripciones. Los mensajes no se transmiten a los gestores de colas que no tienen suscripciones para el tema. Consulte Suscripciones de proxy en una red de publicación/suscripción.

Cuando la tasa de mensajes de publicación entre cualquier gestor de colas y otro en el clúster sea elevada, la infraestructura del canal de clúster entre estos dos puntos deber poder mantener dicha tasa. Para ello puede ser necesario ajustar los canales y las colas de transmisión que se están utilizando.

Cuando utiliza el direccionamiento de host de temas, cada mensaje publicado en un gestor de colas que no es un host de tema se transmite a un gestor de colas de host de temas. Esto es así independientemente de si existen una o varias suscripciones en cualquier otro lugar del clúster. Esto presenta factores adicionales que deben tenerse en cuenta durante la planificación:

- ¿Es aceptable la latencia adicional de enviar, en primer lugar, cada publicación a un gestor de colas de host de temas?
- ¿Puede cada gestor de colas de host de temas sostener la tasa de publicaciones de entrada y salida? Considere un sistema con publicadores en muchos y diferentes gestores de colas. Si todos envían sus mensajes a un conjunto muy pequeño de gestores de colas de host de temas, estos hosts de temas pueden convertirse en un cuello de botella durante el proceso de dichos mensajes y su direccionamiento a los gestores de colas suscriptores.
- ¿Está previsto que una parte importante de los mensajes publicados no tendrán un suscriptor coincidente? Si es así y la tasa de publicación de dichos mensajes es alta, es posible que lo mejor sea convertir el gestor de colas de un publicador en un host de temas. En esta situación, cualquier mensaje publicado en el que no existan suscripciones en el clúster no se transmitirá a ningún otro gestor de colas.

Estos problemas también pueden resolverse introduciendo varios hosts de temas para distribuir la carga de publicación entre los mismos:

- En los casos en los que haya muchos temas distintos y cada uno de ellos tenga una tasa de tráfico de publicaciones, considere alojarlos en gestores de colas diferentes.
- Si no se pueden separar los temas en hosts de temas diferentes, puede definir el mismo objeto de tema en varios gestores de colas. De este modo, se equilibra la carga de trabajo de las publicaciones entre cada uno de ellos para su direccionamiento. Sin embargo, solo resulta adecuado cuando no se requieren un orden de publicación de mensajes.

Cambio de suscripción y series de temas dinámicos

Otro punto a tener en cuenta es el efecto que tiene en el rendimiento del sistema la propagación de las suscripciones del proxy. Normalmente, un gestor de colas envía un mensaje de suscripción de

proxy a determinados gestores de colas del clúster cuando se crea en dicho gestor de colas la primera suscripción para una determinada serie de tema de clúster (no sólo un objeto de tema configurado). Del mismo modo, se envía un mensaje de supresión de la suscripción del proxy cuando se suprime la última suscripción para una serie de tema de clúster específica.

Para el direccionamiento directo, cada gestor de colas con suscripciones envía dichas suscripciones de proxy a cada uno de los otros gestores de colas del clúster. En el direccionamiento de hosts de temas, todo gestor de colas con suscripciones solo envía las suscripciones de proxy a cada gestor de colas que aloja una definición para dicho tema de clúster. Por lo tanto, en el direccionamiento directo, cuantos más gestores de colas haya en el clúster mayor será la actividad general de mantenimiento de las suscripciones del proxy entre los mismos. Mientras que en el direccionamiento de host de temas, el número de gestores de colas del clúster no es un factor.

En ambos modelos de direccionamiento, si una solución de publicación/suscripción consta de muchas series de temas exclusivas con suscripciones, o si frecuentemente se realizan suscripciones y se anulan las suscripciones a los temas de un gestor de colas del clúster, se producirá una actividad general importante en dicho gestor de colas, debido a la generación de mensajes de distribución y supresión de las suscripciones del proxy. En el direccionamiento directo, esto se agrava por la necesidad de enviar estos mensajes a cada gestor de colas del clúster.

Si la tasa de cambio de suscripciones es demasiado elevada para acomodarla, incluso dentro de un sistema de direccionamiento de host de temas, consulte la sección Rendimiento de las suscripciones en las redes de publicación/suscripción para obtener información acerca de cómo disminuir la actividad general de las suscripciones del proxy.

Definición de temas de clúster

Los temas de clúster son temas administrativos con el atributo **cluster** definido. La información sobre temas de clúster se envía a todos los miembros de un clúster y se combina con temas locales para crear partes de un espacio de tema que abarque varios gestores de colas. Esto permite que los mensajes publicados sobre un tema en un gestor de colas se entreguen a las suscripciones de otros gestores de colas del clúster.

Cuando se define un tema de clúster en un gestor de colas, la definición de tema de clúster se envía a los gestores de colas de depósito completo. Los depósitos completos propagan entonces la definición de tema de clúster a todos los gestores de colas del clúster, dejando el mismo tema de clúster disponible para publicadores y suscriptores en cualquier gestor de colas del clúster. El gestor de colas en los que se crea un tema de clúster se conoce como host de tema de clúster. El tema de clúster puede utilizarlo cualquier gestor de colas del clúster, pero las modificaciones de un tema de clúster deben realizarse en el gestor de colas donde se ha definido dicho tema (el host), momento en el cual la modificación se propaga a todos los miembros del clúster a través de los depósitos completos.

Cuando utiliza el direccionamiento directo, la ubicación de la definición del tema en clúster no afecta directamente al comportamiento del sistema, ya que todos los gestores de colas del clúster utilizan la definición de tema del mismo modo. Por lo tanto, debe definir el tema en cualquier gestor de colas que vaya a ser miembro del clúster mientras el tema sea necesario y que esté en un sistema lo suficientemente fiable como para realizar contactos con regularidad con los gestores de colas de repositorio completo.

Cuando utiliza el direccionamiento de host de tema, la ubicación de la definición del tema en clúster es muy importante, ya que los otros gestores de colas del clúster crean canales a este gestor de colas y envían información de suscripción y publicaciones al mismo. Para seleccionar el mejor gestor de colas para alojar la definición de tema, debe comprender el direccionamiento de host de tema. Consulte “Direccionamiento de host de tema en clústeres de publicación/suscripción” en la página 84.

Si tiene un tema de clúster y un objeto de tema local, el tema local tiene prioridad. Consulte “Varias definiciones de temas de clúster con el mismo nombre” en la página 101.

Para obtener información sobre los mandatos a utilizar para visualizar temas de clúster, consulte la información relacionada.

Herencia de temas en clúster

Normalmente, las aplicaciones de publicación y suscripción de una topología de publicación/suscripción esperan funcionar del mismo modo, sin importar el gestor de colas del clúster al que estén conectadas. Por este motivo, los objetos de tema administrados en clúster se propagan a cada gestor de colas del clúster.

Un objeto de tema administrado hereda su comportamiento de otros objetos de tema administrados situados en una posición más alta en el árbol de temas. Esta herencia se genera cuando no se ha establecido un valor explícito para un parámetro de tema.

En el caso de la publicación/suscripción en clúster, es importante tener en cuenta dicha herencia ya que introduce la posibilidad de que los publicadores y suscriptores se comporten de forma diferente en función del gestor de colas al que se conecten. Si un objeto de tema de clúster deja cualquier parámetro para que se herede desde los objetos de tema de nivel superior, es posible que el tema se comporte de forma diferente en gestores de colas diferentes del clúster. Del mismo modo, si se definen localmente los objetos de tema por debajo de un objeto de tema en clúster en el árbol de temas significará que aquellos temas que se encuentren en una posición inferior todavía continúan en el clúster, pero es posible que los objetos locales cambien su comportamiento de un modo diferente al de los otros gestores de colas del clúster.

Suscripciones de comodín

Las suscripciones de proxy se crean cuando se realizan suscripciones locales a una serie de tema que se resuelve en un objeto de tema de clúster o por debajo. Si una suscripción comodín se realiza en un nivel superior de la jerarquía de temas que un tema de clúster, no se envían sus suscripciones de proxy por el clúster para el tema de clúster coincidente y, por lo tanto, no recibe publicaciones de otros miembros del clúster. No obstante, recibe publicaciones del gestor de colas local.

No obstante, si otra aplicación se suscribe a una serie de tema que se resuelve en o por debajo del tema de clúster, se generan las suscripciones de proxy y se propagan las publicaciones a este gestor de colas. Cuando llega, la suscripción de comodín superior se considera un destinatario legítimo de dichas publicaciones y recibe una copia. Si este no es el comportamiento necesario, establezca **`WILDCARD(BLOCK)`** en el tema en clúster. Esto hace que el comodín original no se considere una suscripción legítima e impida que reciba cualquier publicación (ya sea local o de cualquier otro lugar del clúster) en el tema de clúster o en sus subtemas.

Conceptos relacionados

Trabajar con temas administrativos

Trabajar con suscripciones

Referencia relacionada

DISPLAYTOPIC

DISPLAYTPSTATUS

DISPLAYSUB

Atributos de tema de clúster

Cuando un objeto de tema tiene establecido el atributo de nombre de clúster, la definición de tema se propaga a todos los gestores de colas del clúster. Cada gestor de colas utiliza los atributos de tema propagados para controlar el comportamiento de las aplicaciones de publicación/suscripción.

Un objeto de tema tiene varios atributos que se aplican a los clústeres de publicación/suscripción. Algunos controlan el comportamiento general de las aplicaciones de publicación/suscripción y otros controlan cómo se utiliza el tema en todo el clúster.

Se debe configurar una definición de objeto de tema en clúster de modo que puedan utilizarla correctamente todos los gestores de colas del clúster.

Por ejemplo, si las colas de modelo que se han de utilizar para las suscripciones gestionadas (`MDURMDL` y `MNDURMDL`) se establecen en un nombre de cola no predeterminado, dicho modelo nombrado debe definirse en todos los gestores de colas donde se crearán suscripciones gestionadas.

De forma similar, si algún atributo se establece en ASPARENT, el comportamiento del tema dependerá de los nodos superiores del árbol de temas (consulte Objetos de tema administrativo) en cada gestor de colas individual del clúster. Esto puede generar un comportamiento diferente en la publicación o suscripción en diferentes gestores de colas.

Los atributos principales que están directamente relacionados con el comportamiento de publicación/suscripción en todo el clúster son los siguientes:

CLROUTE
Este parámetro controla el direccionamiento de los mensajes entre los gestores de colas a los que están conectados los publicadores y los gestores de colas en los que existen suscripciones coincidentes.

- Se ha de configurar la ruta para que sea directa entre estos gestores de colas o a través de un gestor de colas que aloja una definición del tema en clúster. Consulte Clústeres de publicación/suscripción para obtener más detalles.
- No puede modificar **CLROUTE** mientras esté establecido el parámetro **CLUSTER**. Para modificar **CLROUTE**, en primer lugar, establezca la propiedad **CLUSTER** para que esté en blanco. De esta manera, se impide que las aplicaciones que utilicen el tema se comporten del modo en clúster. A su vez, esto genera una interrupción en las publicaciones que se están entregando a las suscripciones, por lo tanto, mientras realice la modificación también debe desactivar temporalmente la mensajería de publicación/suscripción.

PROXYSUB
Este parámetro controla cuándo se realizan las suscripciones de proxy.

- FIRSTUSE es el valor predeterminado y hace que las suscripciones de proxy se envíen como respuesta a las suscripciones locales de un gestor de colas de una topología de publicación/suscripción distribuida y se cancelen cuando ya no sean necesarias. Para obtener información detallada acerca de por qué puede ser conveniente modificar el valor predeterminado FIRSTUSE de este atributo, consulte el tema Reenvío de suscripciones de proxy individuales y *publicación en todas partes* .
- Para habilitar la *publicación en todas partes*, debe establecer el parámetro **PROXYSUB** en FORCE para un objeto de tema de alto nivel. Esto genera una suscripción de proxy de comodín única que coincide con todos los temas debajo de este objeto de tema en el árbol de temas.

Nota: Si se establece el atributo **PROXYSUB(FORCE)** en un clúster de publicación/suscripción ocupado o de gran tamaño, es posible que se genere una carga excesiva en los recursos del sistema. El atributo **PROXYSUB(FORCE)** se propaga a cada uno de los gestores de colas y no simplemente al gestor de colas donde se ha definido el tema. Esto hace que cada gestor de colas del clúster cree una suscripción de proxy con comodín.

Se envía una copia de un mensaje de este tema, publicada en cualquier gestor de colas, a cada uno de los gestores de colas del clúster, ya sea directamente o a través de un gestor de colas de host de tema, en función del valor **CLROUTE**.

Cuando se trata de un tema de direccionamiento directo, cada gestor de colas crea canales emisores de clúster para cada uno de los otros gestores de colas. Cuando el tema no es de direccionamiento directo, se crean canales a cada gestor de colas de host de tema desde cada gestor de colas del clúster.

Para obtener más información sobre el parámetro **PROXYSUB** cuando se utiliza en clústeres, consulte Rendimiento de la publicación/suscripción de direccionamiento directo.

PUBSCOBE y SUBSCOPE
Estos parámetros determinan si este gestor de colas propaga publicaciones a los gestores de colas de la topología (clúster de publicación/suscripción o jerarquía) o restringe el ámbito simplemente a su gestor de colas. Puede hacer el trabajo equivalente de forma programada utilizando MQPMO_SCOPE_QMGR y MQSO_SCOPE_QMGR.

PUBSCOPE
Si un objeto de tema de clúster se define con **PUBSCOPE(QMGR)**, la definición se comparte con el clúster, pero el ámbito de publicaciones basadas en ese tema sólo es local y no se envían a otros gestores de colas del clúster.

SUBSCOPE
Si un objeto de tema de clúster se define con **SUBSCOPE(QMGR)**, la definición se comparte con el clúster, pero el ámbito de las suscripciones que se basan en dicho tema sólo es local, por lo tanto, no se envían suscripciones de proxy a otros gestores de colas en el clúster.

Estos dos atributos se utilizan conjuntamente para aislar un gestor de colas a fin de que no interactúe con otros miembros del clúster en determinados temas. El gestor de colas no publica ni recibe publicaciones sobre esos temas a o desde otros miembros del clúster. Esta situación no impide la publicación o la suscripción si los objetos de tema están definidos en subtemas.

El establecimiento de **SUBSCOPE** en QMGR en una definición local de un tema no impide que otros gestores de colas del clúster propaguen sus suscripciones de proxy al gestor de colas si están utilizando una versión de clúster del tema, con **SUBSCOPE(ALL)**. No obstante, si la definición local también establece **PUBSCOPE** en QMGR, no se envían publicaciones a esas suscripciones de proxy desde este gestor de colas.

Conceptos relacionados
Ámbito de la publicación
Ámbito de la suscripción

Varias definiciones de temas de clúster con el mismo nombre
Puede definir el mismo objeto de tema del clúster con nombre en el clúster y, en determinadas situaciones, esto habilita un comportamiento específico. Cuando existe varias definiciones de temas de clúster con el mismo nombre, la mayor parte de las propiedades deben coincidir. Si no es así, se emiten errores o avisos en función de la importancia de la falta de coincidencia.

En general, si existe una discrepancia en las propiedades de varias definiciones de temas del clúster, se emiten avisos, y cada gestor de colas del clúster utiliza una de las definiciones de objetos de temas. La definición que utiliza cada gestor de colas no es determinante ni coherente entre los gestores de colas del clúster. Dichas discrepancias se deben resolver a la mayor brevedad posible.

Durante la configuración o mantenimiento del clúster, algunas veces es necesario crear varias definiciones de temas del clúster que no sean idénticas. Sin embargo, esto solo resulta útil como medida temporal y, por lo tanto, se trata como una condición de error potencial.

Cuando se detectan discrepancias, se graban los siguientes mensajes de aviso en cada registro de errores del gestor de colas:

- Multi En Multiplatforms, AMQ9465 y AMQ9466.
- z/OS En z/OS, CSQX465I y CSQX466I.

Las propiedades elegidas para cualquier serie de tema en cada gestor de colas se pueden determinar visualizando el estado del tema en lugar de las definiciones de objeto de tema, por ejemplo, utilizando **DISPLAY TPSTATUS**.

En algunas situaciones, un conflicto en las propiedades de la configuración es lo suficientemente grave como para detener la creación del objeto de tema o para marcar como no válidos los objetos con discrepancias y no propagarlos en el clúster. Consulte **CLSTATE** en DISPLAY TOPIC. Estas situaciones se producen si existe un conflicto en la propiedad de direccionamiento del clúster (**CLROUTE**) de las definiciones de temas. Adicionalmente, debido a la importancia de la coherencia entre las definiciones direccionadas a hosts de temas, las discrepancias adicionales se rechazan tal como se describe detalladamente en las secciones siguientes de este artículo.

Si se detecta el conflicto en el momento en que se define el objeto, el cambio de configuración se rechaza. Si posteriormente los gestores de colas de repositorio completo lo detectan, se graban los siguientes mensajes en los registros de errores de los gestores de colas:

- Multi En Multiplatforms: AMQ9879
- z/OS En z/OS: CSQX879E.

Cuando se definen varias definiciones del mismo objeto de tema en el clúster, una definición definida localmente tiene prioridad sobre una definición realizada de forma remota. Por lo tanto, si existe alguna diferencia en las definiciones, los gestores de colas que alojan las diferentes definiciones se comportarán de manera diferente unos de otros.

El efecto de definir un tema no de clúster con el mismo nombre que un tema de clúster desde otro gestor de colas

Se puede definir un objeto de tema administrado que no está agrupado en clúster en un gestor de colas que está en un clúster y, al mismo tiempo, definir el mismo objeto de tema nombrado como una definición de tema de clúster en un gestor de colas diferente. En este caso, el objeto de tema definido localmente tiene prioridad sobre todas las definiciones remotas del mismo nombre.

Esto tiene el efecto de impedir el comportamiento de agrupación en clúster del tema cuando se utiliza de este gestor de colas. Es decir, es posible que las suscripciones no reciban publicaciones de los publicadores remotos y que los mensajes de los publicadores no se propaguen a las suscripciones remotas del clúster.

Se debe prestar una atención especial antes de configurar un sistema de este tipo, ya que puede conllevar un comportamiento confuso.

Nota: Si un gestor de colas individual necesita impedir que se propaguen las publicaciones y suscripciones alrededor del clúster, incluso cuando el tema se ha agrupado en clúster en otro lugar, un método alternativo es establecer los ámbitos de publicación y suscripción únicamente al gestor de colas local. Consulte “Atributos de tema de clúster” en la página 99.

Varias definiciones de temas de clúster en un clúster de direccionamiento directo

Para el direccionamiento directo, no suele definir el mismo tema de clúster en más de un gestor de colas de clúster. Esto es debido a que el direccionamiento directo hace que el tema esté disponible en todos los gestores de colas del clúster, independientemente del gestor de colas donde se haya definido. Además, si se añaden varias definiciones de temas de clúster se aumenta de forma importante la actividad del sistema y la complejidad de su administración y esta mayor complejidad aumenta la posibilidad de un error humano:

- Cada definición da como resultado que se envíe un objeto de tema de clúster adicional a los otros gestores de colas del clúster, incluidos los gestores de colas de host de tema de clúster.
- Todas las definiciones de un tema específico de un clúster deben ser idénticas, de lo contrario, será difícil asegurarse de qué definición de tema está utilizando un gestor de colas.

Tampoco es esencial que solo el gestor de colas de host esté siempre disponible para que el tema funcione correctamente en todo el clúster, debido a que los gestores de colas de repositorio completo guardan la definición del tema de clúster en la memoria caché y los otros gestores de colas la guardan en sus repositorios parciales de clúster. Para obtener más información, consulte la sección Disponibilidad de los gestores de colas de host de tema que utilizan el direccionamiento directo.

En el caso de que necesite definir temporalmente un tema de clúster en un segundo gestor de colas, por ejemplo, cuando el host del tema existente se debe eliminar del clúster, consulte la sección Mover una definición de tema de clúster a un gestor de colas diferente en el clúster.

Si tiene que modificar una definición de tema de clúster, preste atención y modifíquela en el mismo gestor de colas en que se ha definido. Si intenta modificarla desde otro gestor de colas es posible que se cree accidentalmente una segunda definición del tema con atributos de tema que pueden estar en conflicto.

Varias definiciones de temas de clúster en un clúster de direccionamiento de host de tema

Cuando se define un tema de clúster con un direccionamiento de clúster de *host de tema*, el tema se propaga entre todos los gestores de colas del clúster, del mismo modo que en los temas de direccionamiento *directo*. Adicionalmente, toda la mensajería de publicación/suscripción de dicho tema se direcciona a través de los gestores de colas donde se ha definido dicho tema. Por lo tanto, la ubicación y el número de definiciones del tema en el clúster resulta importante (consulte la sección "Direccionamiento de host de tema en clústeres de publicación/suscripción" en la página 84).

Para asegurar la disponibilidad y escalabilidad adecuadas, si es posible, tenga varias definiciones de temas. Consulte la sección Disponibilidad de los gestores de colas de host de tema que utilizan el direccionamiento de host de tema .

Cuando añada o elimine definiciones adicionales de un tema de direccionamiento de *host de tema* en un clúster, debe tener en cuenta el flujo de mensajes en el momento en que se realiza el cambio de configuración. Si en el momento en que se realiza el cambio se están publicando mensajes para el tema en el clúster, es necesario un proceso gradual para añadir o eliminar una definición de tema. Consulte la sección Mover una definición de tema de clúster a un gestor de colas diferente y la sección Adición de hosts de temas adicionales a un clúster de direccionamiento de host de tema.

Como se ha descrito anteriormente, las propiedades de varias definiciones deben coincidir, con la posible excepción del parámetro **PUB**, como se describe en la sección siguiente. Cuando las publicaciones se direccionan a través de los gestores de colas de host de temas resulta incluso más importante que si existen múltiples definiciones éstas sean coherentes. Por lo tanto, si se detecta una incoherencia en la serie de tema o en el nombre del clúster se rechazará cuando se hayan configurado una o varias definiciones de temas para el direccionamiento de clúster de host de tema.

Nota: También se rechazarán las definiciones de temas de clúster si se intenta configurarlas por encima o por debajo de otro tema en el árbol de temas, cuando la definición de tema de clúster existente se haya configurado para el direccionamiento de host de tema. Esto impide que haya ambigüedades en el direccionamiento de las publicaciones con respecto a las suscripciones con comodines.

Manejo especial del parámetro PUB

El parámetro **PUB** se utiliza para controlar cuando pueden las aplicaciones publicar en un tema. En el caso del direccionamiento de host de tema en un clúster, también puede controlar qué gestores de colas de host de tema se utilizan para el direccionamiento de publicaciones. Por este motivo se permite que existan varias definiciones del mismo objeto de tema en el clúster, con valores diferentes para el parámetro PUB.

Si varias definiciones de clúster remoto de un tema tienen valores diferentes para este parámetro, el tema permite que las publicaciones se envíen y entreguen a las suscripciones cuando se cumplen las condiciones siguientes:

- Ningún objeto de tema coincidente definido en el gestor de colas al que está conectado el publicador se ha establecido en `PUB(DISABLED)`.
- Una o varias definiciones de temas de clúster se han establecido en `PUB(ENABLED)` o una o varias definiciones de temas se han establecido en `PUB(ASPARENT)` y los gestores de colas locales donde está conectado el publicador y donde se ha definido la suscripción están establecidos en `PUB(ENABLED)` en un punto superior del árbol de temas.

En el caso del direccionamiento de host de tema, cuando los mensajes los publican las aplicaciones conectadas a gestores de colas que no son hosts de temas, solo se direccionan los mensajes a los gestores de colas de host de tema si el parámetro **PUB** no se ha establecido de forma explícita en `DISABLED`. Por lo tanto, puede utilizar el valor `PUB(DISABLED)` para desactivar temporalmente el tráfico de mensajes a través de determinados hosts de temas. Es posible que desee hacerlo como preparación para el mantenimiento o eliminación de un gestor de colas o por los motivos que se describen en la sección Adición de hosts de temas adicionales a un clúster de direccionamiento de host de tema.

Disponibilidad de los gestores de colas de host de tema del clúster
Diseñe el clúster de publicación/suscripción para minimizar el riesgo de que, en caso de que un gestor de colas de host de tema pase a estar no disponible, el clúster no pueda procesar el tráfico para el tema. Si un gestor de colas de host de tema pasa a estar no disponible, el efecto que esto puede tener depende de si el clúster está utilizando el direccionamiento de host de tema o el direccionamiento directo.

Disponibilidad de los gestores de colas de host de tema que utilizan el direccionamiento directo

Para el direccionamiento directo, no suele definir el mismo tema de clúster en más de un gestor de colas de clúster. Esto es debido a que el direccionamiento directo hace que el tema esté disponible en todos los gestores de colas del clúster, independientemente del gestor de colas donde se haya definido.Consulte el tema Varias definiciones de temas de clúster en un clúster de direccionamiento de direccionamiento directo.

En un clúster, siempre que un host de un objeto de clúster (por ejemplo, una cola de clúster o un tema de clúster) pasa a estar no disponible durante un periodo de tiempo prolongado, el conocimiento acerca de estos objetos que tienen los otros miembros del clúster caducará finalmente. En el caso de un tema de clúster, si el gestor de colas de host de tema del clúster pasa a estar no disponible, los otros gestores de colas continúan procesando las solicitudes de publicación/suscripción para el tema en modalidad de clúster directo (esto es, envían las publicaciones a las suscripciones en los gestores de colas remotos) durante al menos 60 días desde la última comunicación del gestor de colas de host de tema con los gestores de colas de repositorio completo. Si el gestor de colas en el que ha definido el objeto de tema de clúster no vuelve a estar disponible nunca, se suprimen los objetos de tema almacenados en la memoria caché en los otros gestores de colas y el tema vuelve a ser un tema local, en cuyo caso, las suscripciones dejan de recibir publicaciones de las aplicaciones conectadas a los gestores de colas remotos.

Durante el periodo de 60 días para la recuperación del gestor de colas en el que ha definido un objeto de tema de clúster, no es muy necesario tomar medidas especiales para garantizar que un host de tema de clúster continúe disponible (sin embargo, tenga en cuenta que cualquier suscripción definida en el host de tema no disponible no permanecerá disponible). El período de 60 días es suficiente para atender los problemas técnicos y es probable que solo se supere el período debido a errores administrativos. Para reducir esta posibilidad, si el host de tema de clúster no está disponible, todos los miembros del clúster graban mensajes en el registro de errores cada hora, indicando que no se ha renovado el objeto de tema de clúster en caché. Responda a estos mensajes asegurándose de que se está ejecutando el gestor de colas en el que se ha definido el objeto de tema de clúster. Si no el gestor de colas de host de tema de clúster no puede volver a estar disponible, defina la misma definición de tema de clúster, con exactamente los mismos atributos, en otro gestor de colas del clúster.

Disponibilidad de los gestores de colas de host de tema que utilizan el direccionamiento de host de tema

En el caso del direccionamiento de host de tema, toda la mensajería de publicación/suscripción de un tema se direcciona a través de los gestores de colas donde se ha definido dicho tema. Por este motivo, es muy importante tener en cuenta la disponibilidad continuada de estos gestores de colas en el clúster. Si un host de tema pasa a estar no disponible y no existe otro host para el tema, se detiene inmediatamente el tráfico desde los publicadores a los suscriptores en los diferentes gestores de colas del clúster. Si están disponibles otros hosts de tema, los gestores de colas del clúster direccionan el nuevo tráfico de publicación a través de estos hosts de tema, lo cual proporciona una disponibilidad continuada del direccionamiento de mensajes.

En el caso de los temas directos, transcurridos 60 días, si el primer host de tema continúa sin estar disponible, el conocimiento de dicho tema de host de tema se elimina del clúster. Si esta es la última definición restante de este tema en el clúster, todos los otros gestores de colas dejan de reenviar publicaciones a cualquier host de tema para su direccionamiento.

Por lo tanto, para garantizar la disponibilidad y escalabilidad adecuadas resulta útil, si es posible, definir cada tema en al menos dos gestores de colas del clúster. Esto proporciona una protección adicional

en caso de que un gestor de colas de host de tema determinado pase a estar no disponible. Consulte también Varias definiciones de temas de clúster en un clúster de direccionamiento de host de tema.

Si no puede configurar varios hosts de tema (por ejemplo, debido a que necesita conservar el orden de los mensajes), y no puede configurar solamente un host de tema (porque la disponibilidad de un único gestor de colas no puede afectar el flujo de las publicaciones a las suscripciones a través de todos los gestores de colas del clúster), considere la posibilidad de configurar el tema como un tema de direccionamiento directo. Esto le evitará tener que confiar en un solo gestor de colas para todo el clúster, pero requiere que cada gestor de colas individual esté disponible para procesar las suscripciones y publicaciones alojadas localmente.

Inhabilitación de la publicación/suscripción en un clúster

Introducir el primer tema en clúster de direccionamiento directo en un clúster hace que cada gestor de colas del clúster reconozca cada uno de los otros gestores de colas y puede hacer que estos creen canales entre sí. Si no se desea, en su lugar, debe configurar la publicación/suscripción de direccionamiento de host de tema. Si la existencia de un tema en clúster de direccionamiento directo puede poner en peligro la estabilidad del clúster, debido a temas de escalado de cada gestor de colas, puede inhabilitar por completo la función de publicación/suscripción en clúster estableciendo **PSCLUS** en DISABLED en cada gestor de colas del clúster.

Como se ha descrito en “Direccionamiento directo en clústeres de publicación/suscripción” en la página 79, cuando se introduce en un clúster un tema en clúster de direccionamiento directo, se notifica a todos los repositorios parciales acerca de todos los otros miembros del clúster. El tema de clúster también puede crear suscripciones en todos los otros nodos (por ejemplo, donde se haya especificado **PROXYSUB(FORCE)**) y hacer que se inicien grandes cantidades de canales desde un gestor de colas, aunque no haya ninguna suscripción local. Esto supone una carga inmediata adicional en cada gestor de colas del clúster. En el caso de un clúster que contiene muchos gestores de colas, esto podría provocar una reducción significativa del rendimiento. Por lo tanto, se debe planificar cuidadosamente la introducción de la publicación/suscripción de direccionamiento directo en un clúster.

Si sabe que un clúster no puede asumir la sobrecarga del uso de la publicación/suscripción de direccionamiento directo, puede utilizar la publicación/suscripción de direccionamiento de host de tema. Para obtener una descripción general de las diferencias, consulte “Diseño de clústeres de publicación/suscripción” en la página 77.

Si prefiere inhabilitar por completo la función de publicación/suscripción para el clúster, puede hacerlo estableciendo el atributo del gestor de colas **PSCLUS** en DISABLED en todos los gestores de colas del clúster. Este valor inhabilita en el clúster tanto el direccionamiento directo como la publicación/suscripción de direccionamiento de host de tema ya que modifica tres aspectos de las funciones del gestor de colas:

- Un administrador de este gestor de colas ya no puede definir un objeto de tema Topic como de clúster.
- Las definiciones de tema o las suscripciones proxy entrantes de otros gestores de colas se rechazan y se registra un mensaje de aviso para informar al administrador de una configuración incorrecta.
- Los repositorios completos ya no comparten automáticamente la información sobre cada gestor de colas con todos los demás repositorios parciales cuando reciben una definición de tema.

Aunque **PSCLUS** es un parámetro de cada gestor de colas individual de un clúster, no está pensado para inhabilitar de forma selectiva la publicación/suscripción en un subconjunto de gestores de colas del clúster. Si realiza esta inhabilitación selectiva, frecuentemente verá mensajes de error. Esto es debido a que las suscripciones del proxy y las definiciones de temas se ven y rechazan constantemente si un tema se coloca en clúster en un gestor de colas donde está habilitado **PSCLUS**.

Por lo tanto, debe intentar establecer **PSCLUS** en DISABLED en cada gestor de colas del clúster. Sin embargo, en la práctica este estado puede resultar difícil de obtener y mantener debido a que, por ejemplo, los gestores de colas pueden unirse al clúster y dejarlo en cualquier momento. Como mínimo, debe asegurarse de que **PSCLUS** esté establecido en DISABLED en todos los gestores de colas de repositorio completo. De este modo, si posteriormente se define un tema en clúster en un gestor de colas ENABLED en el clúster, no todos los repositorios completos informarán a cada uno de los gestores de colas y, por lo tanto, el clúster estará protegido ante los posibles problemas de escalada en todos los

gestores de colas. En este caso, se informa acerca del origen del tema en clúster en los registros de errores de los gestores de colas de repositorio completo.

Si un gestor de colas participa en uno o varios clústeres de publicación/suscripción y además en uno o varios clústeres punto a punto, debe establecer **PSCLUS** en ENABLED en dicho gestor de colas. Por este motivo, cuando se solapa un clúster punto a punto con un clúster de publicación/suscripción, debe utilizar un conjunto diferente de repositorios completos en cada clúster. Este método permite que las definiciones de temas y la información sobre cada gestor de colas fluya únicamente en el clúster de publicación/suscripción.

Para evitar las configuraciones incoherentes, cuando cambia **PSCLUS** a ENABLED a DISABLED, no pueden existir objetos de tema en clúster en cualquier clúster del que sea miembro este gestor de colas. Deben suprimirse dichos temas, incluso los definidos de forma remota, antes de cambiar **PSCLUS** a DISABLED.

Para obtener más información sobre **PSCLUS**, consulte ALTER QMGR (PSCLUS).

Conceptos relacionados

Rendimiento de los clústeres de publicación/suscripción de direccionamiento directo

Publicación/suscripción y varios clústeres

Un solo gestor de colas puede ser miembro de más de un clúster. Esta disposición se conoce, algunas veces, como *clústeres solapados*. Mediante este tipo de solapamiento, se podrá acceder a los gestores de colas desde varios clústeres, y el tráfico de mensajes punto a punto se puede direccionar desde los gestores de colas de un clúster a los gestores de colas de otro clúster. Los temas en clúster de los clústeres de publicación/suscripción no proporcionan la misma posibilidad. Por lo tanto, se debe comprender con claridad su comportamiento cuando se utilizan varios clústeres.

A diferencia de lo que ocurre con una cola, no puede asociar una definición de tema con más de un clúster. El ámbito de un tema en clúster está limitado a los gestores de colas del mismo clúster para el que se ha definido el tema. Esto permite propagar las publicaciones únicamente a las suscripciones que están en estos gestores de colas del mismo clúster.

Árbol de temas de un gestor de colas

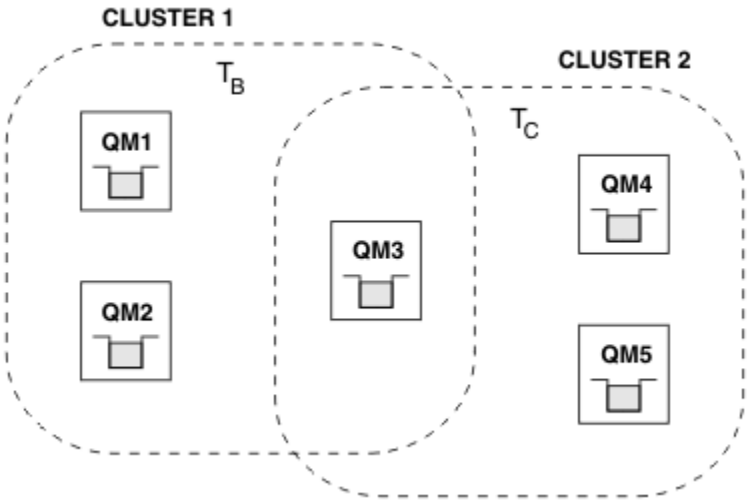


Figura 28. Clústeres solapados: dos clústeres cada uno de ellos suscritos a temas diferentes

Cuando un gestor de colas es un miembro de varios clústeres se le informa de todos los temas en clúster definido en cada uno de estos clústeres. Por ejemplo, en la figura anterior QM3 está al tanto de los objetos de tema en clúster administrados de T_B y T_C, mientras que QM1 solo es consciente de T_B. QM3 aplica ambas definiciones de tema a su tema local y, por lo tanto, tiene un comportamiento diferente en QM1 para determinados temas. Por este motivo, es importante que los temas en clúster de clústeres diferentes no interfieran unos con otros. La interferencia puede suceder cuando se define un tema en

clúster por encima o por debajo de otro tema en clúster en un clúster diferente (por ejemplo, tienen series de tema de `/Sport` y `/Sport/Football`) o incluso para la misma serie de tema en ambos. Otro tipo de interferencia es cuando se definen los objetos de tema en clúster administrados con el mismo nombre de objeto en clústeres diferentes pero para diferentes series de tema.

Si se crea una configuración de este tipo, la entrega de publicaciones a las suscripciones coincidentes pasa a ser muy dependiente de las ubicaciones relativas de los publicadores y suscriptores en relación con el clúster. Por este motivo, no puede confiar en una configuración de este tipo y debe modificarla para eliminar los temas que interfieren.

Durante la planificación de una topología de clústeres solapados con mensajería de publicación/suscripción, puede evitar cualquier interferencia tratando el árbol de temas y los nombres de objetos de tema en clúster como si abarcaran todos los clústeres solapados de la topología.

Integración de varios clústeres de publicación/suscripción

Si se requiere una mensajería de publicación/suscripción que abarque gestores de colas de clústeres diferentes, existen dos opciones disponibles:

- Conectar los clústeres entre sí utilizando una configuración de una jerarquía de publicación/suscripción. Consulte Combinación de espacios de temas de varios clústeres.
- Cree un clúster adicional que cubra los clústeres existentes e incluya todos los gestores de colas que necesiten la publicación o suscripción a un tema concreto.

Con la última opción, debe considerar detenidamente el tamaño del clúster y el mecanismo de direccionamiento del clúster más eficaz. Consulte “Diseño de clústeres de publicación/suscripción” en la página 77.

Consideraciones de diseño acerca de las publicaciones retenidas en los clústeres de publicación/suscripción

Existen algunas restricciones que se han de tener en cuenta cuando se diseña un clúster de publicación/suscripción para trabajar con publicaciones retenidas.

Consideraciones

Consideración 1: Los siguientes gestores de colas del clúster siempre almacenan la última versión de una publicación retenida:

- El gestor de colas del publicador
- En un clúster de direccionamiento de host de tema, el host de tema (siempre que solo haya un host de tema para el tema, como se describe en la siguiente sección de este artículo)
- Todos los gestores de colas con suscripciones que coincidan con la serie de tema de la publicación retenida

Consideración 2: Los gestores de colas que no reciben publicaciones retenidas actualizadas mientras no tienen ninguna suscripción. Por lo tanto, cualquier publicación retenida almacenada en un gestor de colas que ya no esté suscrita al tema pasará a estar caducada.

Consideración 3: Cuando se crea cualquier suscripción, si existe una copia local de una publicación retenida para la serie de tema, la copia local se entrega a la suscripción. Si es usted el primer suscriptor a cualquier serie de tema concreta, también se entregará una publicación retenida coincidente desde uno de los siguientes miembros del clúster:

- En un clúster de direccionamiento directo, el gestor de colas del publicador
- En un clúster de direccionamiento de host de tema, los hosts de temas para el tema concreto

La entrega de una publicación retenida de un host de tema o gestor de colas de publicación al gestor de colas de suscripción es asíncrona a las llamadas de MQSUB. Por lo tanto, si utiliza la llamada MQSUBRQ, es posible que la última publicación retenida se pierda hasta una llamada posterior a MQSUBRQ.

Implicaciones

En cualquier clúster de publicación/suscripción, cuando se realiza una primera suscripción, es posible que el gestor de colas local esté almacenando una copia caducada de una publicación retenida y ésta sea la copia que se entregue a la nueva suscripción. La existencia de una suscripción en el gestor de colas local significa que esto se resolverá la próxima vez que se actualice la publicación retenida.

En el caso de un clúster de publicación/suscripción de direccionamiento de host de tema, si configura más de un host de tema para un tema concreto, es posible que los suscriptores nuevos reciban la publicación retenida más reciente de un host de tema o es posible que reciban una publicación retenida caducada de otro host de tema (habiéndose perdido la más reciente). En el direccionamiento de host de tema, es habitual configurar varios hosts de temas para un tema concreto. Sin embargo, si espera que las aplicaciones utilicen las publicaciones retenidas, solo debe configurar un host de tema para cada tema.

Para cualquier serie de tema concreta, solo debe utilizar un único publicador y asegurarse de que el publicador utilice siempre el mismo gestor de colas. De lo contrario, es posible que las diferentes publicaciones retenidas puedan estar activas en gestores de colas diferentes para el mismo tema, lo que provoca un comportamiento imprevisto. Dado que se distribuyen varias suscripciones de proxy, es posible que se reciban varias publicaciones retenidas.

Si continúa preocupado por el modo en que utilizan los suscriptores las publicaciones caducadas, considere establecer la caducidad de mensajes cuando cree cada publicación retenida.

Puede utilizar el mandato **CLEAR TOPICSTR** para eliminar una publicación retenida de un clúster de publicación/suscripción. En determinadas circunstancias, es posible que tenga que emitir el mandato en varios miembros del clúster de publicación/suscripción, como se describe en **CLEAR TOPICSTR** .

Suscripción de comodín y publicaciones retenidas

Si utiliza suscripciones de comodín, en las suscripciones de proxy correspondientes que se entregan a los otros miembros del clúster de publicación/suscripción se utilizan comodines desde el separador de tema inmediatamente anterior al primer carácter de comodín. Consulte Comodines y temas de clúster.

Por lo tanto, el comodín utilizado puede coincidir con más series de tema y más publicaciones retenidas de los que coincidirán con la aplicación de suscripción.

Esto aumenta la cantidad de almacenamiento necesario para las publicaciones retenidas y, por lo tanto, debe asegurarse de que los gestores de colas de host tengan capacidad de almacenamiento suficiente.

Conceptos relacionados

Publicaciones retenidas

Reenvío de suscripciones de proxy individuales y publicación en todas partes

Consideraciones sobre REFRESH CLUSTER para clústeres de publicación/suscripción

La emisión del mandato **REFRESH CLUSTER** hace que el gestor de colas descarte temporalmente la información sobre un clúster guardada localmente, incluidos los temas de clúster y sus suscripciones de proxy asociadas.

El tiempo transcurrido desde la emisión del mandato **REFRESH CLUSTER** hasta el punto en que el gestor de colas recupera un conocimiento completo de la información necesaria para la publicación/suscripción en clúster depende del tamaño del clúster, de la disponibilidad y de la capacidad de respuesta de los gestores de colas de repositorio completo.

Durante el proceso de renovación, se produce una interrupción del tráfico de publicación/suscripción en un clúster de publicación/suscripción. Para clústeres grandes, el uso del mandato **REFRESH CLUSTER** puede interrumpir el clúster mientras está en curso, y de nuevo a intervalos de 27 días a partir de entonces cuando los objetos de clúster envían automáticamente actualizaciones de estado a todos los gestores de colas interesados. Consulte La renovación en un clúster grande puede afectar el rendimiento y la disponibilidad del clúster. Por estos motivos, sólo debe utilizarse el mandato **REFRESH CLUSTER** en un clúster de publicación/suscripción cuando así se lo indique el Centro de soporte de IBM.

La interrupción en el clúster puede aparecer externamente con los síntomas siguientes:

- Las suscripciones a temas de clúster en este gestor de colas no reciben publicaciones de los publicadores conectados a otros gestores de colas en el clúster.
- Los mensajes publicados en temas de clúster en este gestor de colas no se propagan a las suscripciones en otros gestores de colas.
- Las suscripciones a temas de clúster en este gestor de colas creadas durante este período no envían sistemáticamente suscripciones de proxy a otros miembros del clúster.
- Las suscripciones a temas de clúster en este gestor de colas suprimidas durante este período no eliminan sistemáticamente suscripciones de proxy de otros miembros del clúster.
- Pausas de 10 segundos o más en la entrega de mensajes.
- Anomalías de **MQPUT**, por ejemplo, MQRC_PUBLICATION_FAILURE.
- Publicaciones colocadas en la cola de mensajes no entregados con una razón de MQRC_UNKNOWN_REMOTE_Q_MGR

Por estas razones, las aplicaciones de publicación/suscripción se deben desactivar antes de emitir el mandato **REFRESH CLUSTER**.

Después de emitir un mandato **REFRESH CLUSTER** en un gestor de colas en un clúster de publicación/suscripción, espere a que todos los gestores de colas del clúster y los temas del clúster se hayan actualizado correctamente, y vuelva a sincronizar suscripciones proxy como se describe en Resincronización de suscripciones de proxy. Una vez que todas las suscripciones proxy se han resincronizado correctamente, reinicie las aplicaciones de publicación/suscripción.

Si un mandato **REFRESH CLUSTER** está tardando mucho tiempo en completarse, supervíselo consultando CURDEPTH de SYSTEM.CLUSTER.COMMAND.QUEUE.

Conceptos relacionados

"Agrupación en clúster: utilización de las recomendaciones de REFRESH CLUSTER" en la página 70
Puede utilizar el mandato **REFRESH CLUSTER** para descartar toda la información retenida localmente sobre un clúster y reconstruir esa información a partir de los repositorios completos en el clúster. Es poco probable que necesite utilizar este mandato, excepto en circunstancias excepcionales. Si necesitara utilizar este mandato, existen algunas consideraciones especiales sobre cómo se utiliza. Esta información es una guía basada en las pruebas y los comentarios de los clientes.

Referencia relacionada

Problemas de aplicación vistos al ejecutar REFRESH CLUSTER

Referencia de mandatos MQSC: REFRESH CLUSTER

Direccionamiento en las jerarquías de publicación/suscripción

Si su topología de gestores de colas distribuidos es una jerarquía de publicación/suscripción y se realiza una suscripción en un gestor de colas, de forma predeterminada, se crea una suscripción del proxy en cada gestor de colas de la jerarquía. Las publicaciones que se reciben en cualquier gestor de colas se direccionan a través de la jerarquía a cada gestor de colas que aloje una suscripción coincidente.

Para obtener una introducción sobre cómo se direccionan los mensajes entre gestores de colas en jerarquías de publicación/suscripción y clústeres, consulte Redes de publicación/suscripción distribuidas.

Cuando se realiza una suscripción a un tema en un gestor de colas de una jerarquía de publicación/suscripción distribuida, el gestor de colas gestiona el proceso mediante el cual la suscripción se propaga a los gestores de colas conectados. Las *suscripciones de proxy* fluyen a todos los gestores de colas de la red. Una suscripción de proxy proporciona a un gestor de colas la información que necesita para reenviar una publicación a aquellos gestores de colas que alojan suscripciones para dicho tema. Cada gestor de colas de una jerarquía de publicaciones solo reconoce sus relaciones directas. Las publicaciones que se colocan en un gestor de colas se envía, a través de sus relaciones directas, a dichos gestores de colas con suscripciones. En la imagen siguiente se ilustra este concepto, donde *Suscriptor 1* registra una suscripción para tema concreto en el gestor de colas *Asia* (1). Las suscripciones de proxy para esta suscripción en el gestor de colas *Asia* se reenvían a todos los otros gestores de colas de la red (2,3,4).

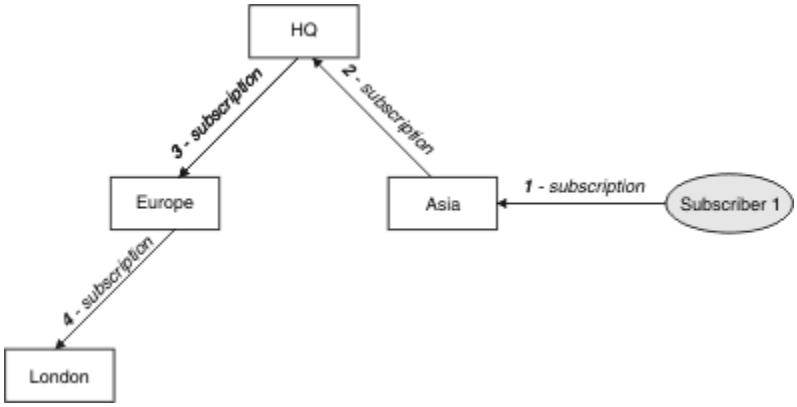


Figura 29. Propagación de suscripciones a través de una red de gestores de colas

Un gestor de colas consolida todas las suscripciones que se crean en él, sean de aplicaciones locales o de los gestores de colas remotos. Crea las suscripciones de proxy para los temas de las suscripciones con sus vecinos, a menos que ya exista una suscripción. Esto se ilustra en la siguiente imagen, en la que *Suscriptor 2* registra una suscripción para el mismo tema que en la Figura 29 en la página 110, en el gestor de colas *HQ* (5). La suscripción de este tema se reenvía al gestor de colas *Asia*, de modo que éste reconoce que las suscripciones existen en otro lugar de la red (6). La suscripción no se reenvía al gestor de colas *Europa*, porque ya se ha registrado una suscripción para este tema; consulte el paso 3 en la Figura 29 en la página 110.

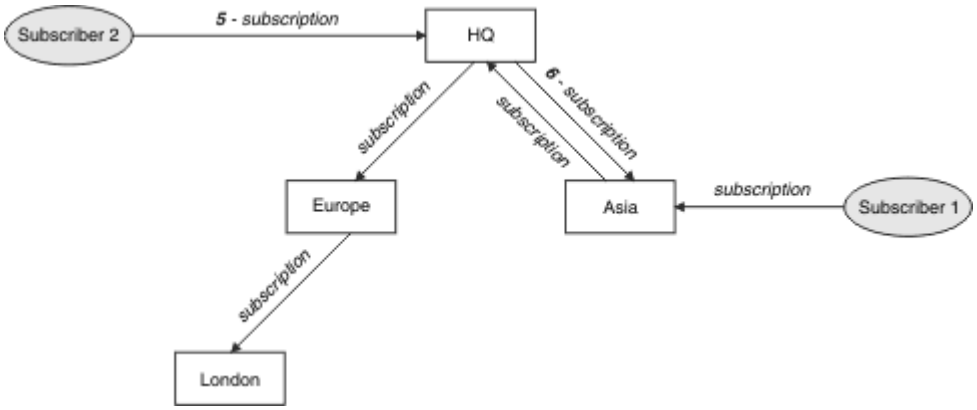


Figura 30. Varias suscripciones

Cuando una aplicación publica información para un tema, el gestor de colas receptor la reenvía a todos los gestores de colas que tienen suscripciones válidas para el tema. Es posible que la reenvíe a través de uno o varios gestores de colas intermedios. Esto se ilustra en la figura siguiente, en la que un publicador envía una publicación al gestor de colas *Europa* (7), sobre el mismo tema que en la Figura 30 en la página 110. Existe una suscripción para este tema desde *HQ* a *Europa*, de modo que la publicación se reenvía al gestor de colas *HQ* (8). Sin embargo, no existe ninguna suscripción desde *Londres* a *Europa* (solo desde *Europa* a *Londres*), de modo que la publicación no se reenvía al gestor de colas *Londres*. El gestor de colas *HQ* envía directamente la publicación al *Suscriptor 2* y al gestor de colas *Asia* (9). La publicación se reenvía al *Suscriptor 1* desde *Asia* (10).

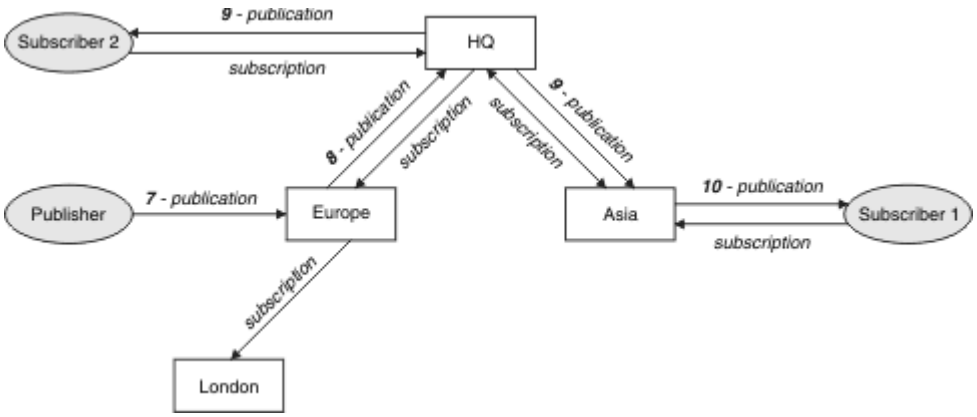


Figura 31. Propagación de publicaciones a través de una red de gestores de colas

Cuando un gestor de colas envía cualquier publicación o suscripción a otro gestor de colas, establece su propio ID de usuario en el mensaje. Si está utilizando una jerarquía de publicación/suscripción, y si el canal de entrada está configurado para colocar mensajes con la autorización del ID de usuario del mensaje, debe autorizar el ID de usuario del gestor de colas emisor. Consulte Utilización de los ID de usuario predeterminados con una jerarquía de gestores de colas.

Nota: Si, en su lugar, utiliza los clústeres de publicación/suscripción, el clúster maneja la autorización.

Resumen y consideraciones adicionales

Una jerarquía de publicación/suscripción le proporciona un control preciso sobre la relación entre los gestores de colas. Una vez creada, solo se necesita una ligera intervención para poder administrarla. Sin embargo, también impone determinadas restricciones en su sistema:

- Los nodos situados en la posición más alta de la jerarquía, en especial el nodo raíz, se deben alojar en un equipo potente, de alta disponibilidad y alto rendimiento. Esto es debido a que se espera fluya más tráfico de publicaciones a través de estos nodos.
- La disponibilidad de cada gestor de colas no de hoja de la jerarquía afecta a la capacidad de la red de permitir que los mensajes fluyan desde los publicadores a los suscriptores en otros gestores de colas.
- De forma predeterminada, todas las series de temas suscritas se propagan a través de la jerarquía y las publicaciones solo se propagan a los gestores de colas remotos que tienen una suscripción con el tema asociado. Por lo tanto, los cambios rápidos en el conjunto de suscripciones pueden convertirse en un factor limitante. Puede cambiar este comportamiento predeterminado y, en su lugar, propagar todas las publicaciones a todos los gestores de colas, lo cual elimina la necesidad de las suscripciones de proxy. Consulte Rendimiento de suscripción en redes de publicación/suscripción.

  Nota: También se aplica una restricción similar a los clústeres que se direccionan directamente.
- Debido a la naturaleza interconectada de los gestores de colas de publicación/suscripción, se necesita tiempo para que las suscripciones de proxy se propaguen alrededor de todos los nodos de la red. No necesariamente, las publicaciones remotas comienzan con suscripciones de forma inmediata, por lo tanto, es posible que no se envíen publicaciones anticipadas tras una suscripción a una nueva serie de tema. Puede eliminar los problemas ocasionados por el retardo de la suscripción propagando todas las publicaciones a todos los gestores de colas, con lo cual se elimina la necesidad de suscripciones del proxy. Consulte Rendimiento de suscripción en redes de publicación/suscripción.

  Nota: También se aplica esta restricción los clústeres que se direccionan directamente.
- En el caso de una jerarquía de publicación/suscripción, para añadir o eliminar gestores de colas es necesaria una configuración manual de la jerarquía, teniendo muy en cuenta la ubicación de estos gestores de colas y su dependencia de otros gestores de colas. A menos que esté añadiendo o eliminando los gestores de colas que se encuentran en la parte inferior de la jerarquía y que, por lo

tanto, no tienen debajo ninguna rama, también tendrá que configurar otros gestores de colas en la jerarquía.

Antes de utilizar una jerarquía de publicación/suscripción como su mecanismo de direccionamiento, explore los métodos alternativos que se describen detalladamente en "Direccionamiento directo en clústeres de publicación/suscripción" en la página 79 y en "Direccionamiento de host de tema en clústeres de publicación/suscripción" en la página 84.

Colas del sistema de publicación/suscripción distribuida

Los gestores de colas utilizan cuatro colas de sistema para la mensajería de publicación/suscripción. Debe tener en cuenta su existencia solo para la determinación de problemas o para fines de planificación de capacidad.

Consulte Equilibrio de productores y consumidores en las redes de publicación/suscripción como ayuda para supervisar estas colas.

Tabla 6. Colas del sistema de publicación/suscripción en Multiplatforms

Cola de sistema	**Finalidad**
SYSTEM.INTER.QMGR.CONTROL	Cola de control de publicación/suscripción distribuida de IBM MQ
SYSTEM.INTER.QMGR.FANREQ	Cola de entrada de proceso en abanico de suscripción proxy interna de publicación/suscripción distribuida de IBM MQ
SYSTEM.INTER.QMGR.PUBS	Publicaciones de publicación/suscripción distribuida de IBM MQ
SYSTEM.HIERARCHY.STATE	Estado de las relaciones de la jerarquía de publicación/suscripción distribuida de IBM MQ

z/OS En z/OS, configure los objetos de sistema necesarios al crear el gestor de colas, incluyendo las muestras CSQ4INSX, CSQ4INSR y CSQ4INSG en el conjunto de datos de entrada de inicialización CSQINP2. Para obtener más información, consulte la Tarea 13: Personalizar los conjuntos de datos de entrada de inicialización.

Los atributos de las colas del sistema de publicación/suscripción se muestran en la Tabla 7 en la página 112.

Tabla 7. Colas de sistema de publicación/suscripción distribuidas

Atributo	**Valor predeterminado**
DEFPSIST	Sí
DEFSOPT	SHARED
MAXMSGL	Multi En Multiplatforms: el valor del parámetro MAXMSGL del mandato ALTER QMGR z/OS En z/OS: 4194304 (es decir, 4 MB)
MAXDEPTH	999999999
SHARE	No disponible
z/OS z/OS STGCLASS	Este atributo sólo se utiliza en las plataformas z/OS

Nota: La única cola que contiene mensajes enviados por las aplicaciones es SYSTEM.INTER.QMGR.PUBS. **MAXDEPTH** se establece en su valor máximo para esta cola para permitir la acumulación temporal de mensajes publicados durante las interrupciones o los tiempos de carga excesiva. Si el gestor de colas se ejecuta en un sistema en el que la profundidad de la cola no se puede contener, esto se debe ajustar.

Tareas relacionadas

Resolución de problemas de publicación/suscripción distribuida

Errores de las colas del sistema de publicación/suscripción distribuidas

Pueden producirse errores cuando las colas del gestor de colas de publicación/suscripción distribuidas no están disponibles. Esto afecta a la propagación del conocimiento de las suscripciones en la red de publicación/suscripción y a la publicación en suscripciones que se encuentran en gestores remotos.

Si la cola de solicitud de diseminación SYSTEM.INTER.QMGR.FANREQ no está disponible, la creación de una suscripción puede generar un error y los mensajes de error se grabarán en el registro de errores del gestor de colas cuando las suscripciones de proxy deban entregarse a los gestores de colas conectados directamente.

Si la cola de estado de relación de jerarquía SYSTEM.HIERARCHY.STATE está disponible, se escribe un mensaje de error en el registro de errores del gestor de colas y el motor de publicación/suscripción se pone en la modalidad COMPAT. Para ver la modalidad de publicación/suscripción, utilice el mandato DISPLAY QMGR PSMODE.

Si alguna otra de las colas SYSTEM.INTER.QMGR no está disponible, se graba un mensaje de error en el registro de errores del gestor de colas y, aunque la función no está inhabilitada, es probable que los mensajes de publicación/suscripción se acumulen en colas en este o en gestores de colas remotos.

Si no está disponible la cola del sistema de publicación/suscripción o la cola de transmisión necesaria a un gestor de colas del clúster de publicación/suscripción, padre o hijo, se producen los siguientes resultados:

- No se entregan las publicaciones y es posible que una aplicación de publicación reciba un error. Para obtener detalles sobre cuándo la aplicación de publicación recibe un error, consulte los siguientes parámetros del mandato **DEFINE TOPIC** : **PMSGDLV** , **NPMSGDLV** y **USEDLQ** .
- Las publicaciones entre gestores de colas recibidas se restituyen en la cola de entrada y posteriormente se vuelven a intentar. Si se alcanza el umbral de restitución, las publicaciones no entregadas se colocan en la cola de mensajes no entregados. El registro de errores del gestor de colas contendrá detalles acerca del problema.
- Se restituye una suscripción de proxy no entregada a la cola de solicitudes diseminadas y se vuelve a intentar posteriormente. Si se alcanza el umbral de restitución, las suscripciones del proxy no entregadas no se entregan a ningún gestor de colas y se colocan en la cola de mensajes no entregados. El registro de errores del gestor de colas contendrá detalles del problema, incluidos los detalles acerca de cualquier acción administrativa correctiva que sea necesaria.
- Los mensajes del protocolo de relación de jerarquía fallan y el estado de la conexión se marca como ERROR. Para ver el estado de conexión, utilice el mandato **DISPLAY PUBSUB**.

Tareas relacionadas

Resolución de problemas de publicación/suscripción distribuida

Multi Planificación de los requisitos de almacenamiento y rendimiento en Multiplatforms

Debe configurar un almacenamiento realista y alcanzable así como objetivos de rendimiento para el sistema IBM MQ. Utilice los enlaces para obtener información sobre factores que afecten al almacenamiento y al rendimiento en la plataforma.

Los requisitos varían según los sistemas en los que utiliza IBM MQ y los componentes que desea utilizar.

Para obtener la información más reciente sobre los entornos hardware y software soportados, consulte Requisitos del sistema para IBM MQ.

IBM MQ almacena datos del gestor de colas en el sistema de archivos. Utilice los enlaces siguientes para obtener información sobre la planificación y la configuración de estructuras de directorios para su uso con IBM MQ:

- "Planificación del soporte del sistema de archivos en Multiplatforms" en la página 117
- "Requisitos para los sistemas de archivos compartidos en Multiplatforms" en la página 117
- "Compartición de archivos de IBM MQ en Multiplatforms" en la página 127
- Linux AIX "Estructura de directorios en sistemas AIX and Linux" en la página 129
- Windows "Estructura de directorios en sistemas Windows" en la página 139
- IBM i "Estructura de directorios en IBM i" en la página 142

Utilice los enlaces siguientes para obtener información acerca de los recursos del sistema, la memoria compartida y la prioridad de procesos en AIX and Linux:

- Linux AIX "IBM MQ y los recursos IPC de UNIX System V" en la página 147
- AIX "Memoria compartida en AIX" en la página 146
- Linux AIX "IBM MQ y prioridad de procesos en UNIX" en la página 147

Utilice los enlaces siguientes para obtener información acerca de los archivos de registro:

- "Elección de registro circular o lineal en Multiplatforms" en la página 146
- Cálculo del tamaño de registro

Conceptos relacionados

"Planning your IBM MQ environment on z/OS" en la página 147
When planning your IBM MQ environment, you must consider the resource requirements for data sets, page sets, Db2, Coupling Facilities, and the need for logging, and backup facilities. Use this topic to plan the environment where IBM MQ runs.

Tareas relacionadas

"Planificación de una arquitectura de IBM MQ" en la página 5
Cuando planifique su entorno de IBM MQ, tenga en cuenta el soporte que proporciona IBM MQ para las arquitecturas de uno o varios gestores de colas y para los estilos de mensajería de punto a punto y de publicación/suscripción. Además planifique los requisitos de recursos y su uso de los recursos de registro y copia de seguridad.

Referencia relacionada

Requisitos de hardware y software en AIX and Linux
Requisitos de hardware y software en Windows

Multi Requisitos de espacio de disco en Multiplatforms

Los requisitos de almacenamiento para IBM MQ dependen de los componentes que instale y de cuánto espacio necesite.

El almacenamiento de disco es necesario para los componentes opcionales que elige instalar, incluidos todos los componentes de requisito previo son necesarios. El requisito de almacenamiento total también depende del número de colas que se utilizan, el número y el tamaño de los mensajes de las colas y si los mensajes son persistentes. También necesita capacidad para archivar en disco, cinta u otros soportes, así como espacio para los programas de aplicación propios.

Las tablas siguientes muestran el espacio de disco aproximado necesario cuando se instalan distintas combinaciones del producto en distintas plataformas. (Los valores se redondean al valor de 5 MB más próximo, donde un MB es 1.048.576 bytes.)

- LTS “Requisitos de espacio de disco para Long Term Support” en la página 115
- CD “Requisitos de espacio de disco para Continuous Delivery” en la página 116

Requisitos de espacio de disco para Long Term Support

V 9.4.0 LTS

Tabla 8. Requisitos de espacio de disco de IBM MQ para multiplataformas para Long Term Support

Plataforma	Instalación de cliente “1” en la página 115	Instalación de servidor “2” en la página 115	Instalación completa “3” en la página 115
AIX AIX	335 MB	375 MB	1810 MB
IBM i IBM i (consulte Notas adicionales para IBM i)	485 MB	845 MB	1965 MB
Linux Linux for x86-64	270 MB	295 MB	2010 MB
Linux Linux on Power Systems - Little Endian	170 MB	190 MB	1400 MB
Linux Linux for IBM Z	255 MB	290 MB	1485 MB
Windows Windows (instalación de 64 bits) “4” en la página 115	295 MB	425 MB	2310 MB

Notas:

1. Una instalación de cliente incluye los siguientes componentes:
   - Tiempo de ejecución
   - Cliente
2. Una instalación de servidor incluye los siguientes componentes:
   - Tiempo de ejecución
   - Servidor
3. Una instalación completa incluye todos los componentes disponibles.
4. Windows No todos los componentes listados aquí son características instalables en sistemas Windows; su funcionalidad algunas veces se incluye en otras características. Consulte Características de IBM MQ para sistemas Windows.

Notas adicionales para IBM i: IBM i

1. En IBM i no puede separar el cliente nativo del servidor. La figura del servidor en la tabla es 5724H72*BASE sin Java, junto con la carga del idioma inglés (2924). Hay 22 cargas de idiomas exclusivas posibles.
2. La figura de la tabla es para el cliente nativo 5725A49 *BASE sin Java.
3. Las clases Java y JMS se pueden añadir a los enlaces de servidor y cliente. Si desea incluir estas características, añada 110 MB.
4. La adición de origen de ejemplos al cliente o servidor añade 10 MB adicionales.

5. La adición de ejemplos a las clases Java y JMS añade 5 MB adicionales.

Requisitos de espacio de disco para Continuous Delivery

V 9.4.0 CD

Tabla 9. Requisitos de espacio de disco de IBM MQ para multiplataformas para Continuous Delivery

Plataforma/release de CD	Instalación de cliente "1" en la página 116	Instalación de servidor "2" en la página 116	Instalación completa "3" en la página 116
AIX **AIX**			
V 9.4.0 IBM MQ 9.4.0	355 MB	390 MB	1440 MB
Linux **Linux para x86-64 (64 bits)**			
V 9.4.0 IBM MQ 9.4.0	280 MB	295 MB	1195 MB
Linux **Linux on Power Systems - Little Endian**			
V 9.4.0 IBM MQ 9.4.0	170 MB	195 MB	1075 MB
Linux **Linux para IBM Z**			
V 9.4.0 IBM MQ 9.4.0	260 MB	290 MB	1160 MB
Windows **Windows (instalación de 64 bits)** "4" en la página 116			
V 9.4.0 IBM MQ 9.4.0	300 MB	425 MB	1785 MB

Notas:

1. Una instalación de cliente incluye los siguientes componentes:
   - Tiempo de ejecución
   - Cliente
2. Una instalación de servidor incluye los siguientes componentes:
   - Tiempo de ejecución
   - Servidor
3. Una instalación completa incluye todos los componentes disponibles.
4. Windows No todos los componentes listados aquí son características instalables en sistemas Windows; su funcionalidad algunas veces se incluye en otras características. Consulte Características de IBM MQ para sistemas Windows.

Conceptos relacionados

Componentes y características de IBM MQ

Multi Planificación del soporte del sistema de archivos en Multiplatforms

Los datos del gestor de colas se almacenan en el sistema de archivos. Un gestor de colas utiliza un bloqueo del sistema de archivos para evitar que varias instancias de un gestor de colas multiinstancia se activen a la vez.

Sistemas de archivos compartidos

Los sistemas de archivos compartidos permiten varios sistemas para acceder al mismo dispositivo de almacenaje físico simultáneamente. Los datos pueden corromperse si varios sistemas acceden al mismo dispositivo físico de almacenaje directamente sin aplicar control de simultaneidad o bloqueo. Los sistemas operativos proporcionan sistemas de archivos locales con control de simultaneidad y bloqueo en procesos locales; los sistemas de archivos de red proporcionan control de simultaneidad y bloqueo en sistemas distribuidos.

Históricamente, los sistemas de archivos interconectados no han respondido lo suficientemente rápido, o no han ofrecido un control de simultaneidad y bloqueo suficiente, para satisfacer los requisitos de registrar mensajes. Actualmente, los sistemas de archivos interconectados ofrecen un buen rendimiento y protocolos de sistemas de archivos de red fiables, como *RFC 3530, protocolo de Sistema de archivos de red (NFS) versión 4*, que cumplen los requisitos para registrar mensajes con fiabilidad.

Sistemas de archivos compartidos y IBM MQ

Los datos de gestor de colas para un gestor de colas multiinstancia se almacenan en un sistema de archivos de red compartidos. En sistemas AIX, Linux, and Windows, los archivos de datos y los archivos de registro del gestor de colas deben colocarse en el sistema de archivos de red compartidos.

IBM i En IBM i, se utilizan diarios en vez de archivos de registros. Estos diarios no pueden compartirse. Los gestores de colas multiinstancia en IBM i utilizan la duplicación de diarios, o diarios intercambiables, para hacer que los diarios estén disponibles entre diferentes instancias del gestor de colas.

IBM MQ utiliza el bloqueo para impedir que varias instancias del mismo gestor de colas de varias instancias estén activas al mismo tiempo. El mismo bloqueo también garantiza que dos gestores de colas distintos no puedan utilizar de forma inadvertida el mismo catálogo de archivos de datos de gestor de colas. Sólo una instancia de un gestor de colas puede tener el bloqueo al mismo tiempo. En consecuencia, IBM MQ da soporte a los datos del gestor de colas almacenados en almacenamiento de red al que se accede como sistema de archivos compartidos.

No todos lo protocolos de bloqueo de sistemas de red son sólidos y, además, un sistema de archivos pueden estar configurado para obtener rendimiento en vez de por la integridad de sus datos, por lo que debe ejecutar el mandato **amqmfsck** para probar si el sistema de archivos en red realizará correctamente el control de acceso a los registros y datos de gestor de colas. Este mandato únicamente se aplica a los sistemas UNIX, Linux y IBM i. En Windows, sólo hay un sistema de archivos de red soportado y el mandato **amqmfsck** no es necesario.

Tareas relacionadas

“Verificación del comportamiento del sistema de archivos compartidos en Multiplatforms” en la página 119

Ejecute **amqmfsck** para comprobar si un sistema de archivos compartido en AIX, Linux o IBM i cumple los requisitos para almacenar los datos del gestor de colas de un gestor de colas de varias instancias. (El único requisito para una configuración de Windows es que utilice SMB 3 para el suministro de almacenamiento compartido.)

Multi Requisitos para los sistemas de archivos compartidos en Multiplatforms

Los sistemas de archivos compartidos deben proporcionar integridad de grabación de datos, acceso exclusivo garantizado a archivos y bloqueos de releases con anomalías para poder trabajar con IBM MQ de forma fiable.

Requisitos que debe cumplir un sistema de archivos compartido

Existen tres requisitos fundamentales que debe cumplir un sistema de archivos compartido para que funciona con fiabilidad con IBM MQ:

1. Integridad de grabación de datos

   La integridad de grabación de datos se llama también a veces *Grabar en disco en reserva de memoria*. El gestor de colas debe poder sincronizar con los datos que se están confirmado satisfactoriamente en el dispositivo físico. En un sistema transaccional, necesita asegurarse de que algunas grabaciones se han confirmado correctamente antes de continuar con otro proceso.

   Más concretamente, las plataformas IBM MQ for AIX or Linux utilizan la opción de apertura *O_SYNC* y la llamada al sistema `fsync()` para forzar explícitamente las grabaciones en soportes recuperables, y la operación de grabación depende de que estas opciones funcionen correctamente.

   Atención: Linux Debe montar el sistema de archivos con la opción `async`, que todavía da soporte a la opción de grabaciones síncronas y ofrece mejor rendimiento que la opción `sync`.

   Sin embargo, tenga en cuenta que si el sistema de archivos se ha exportado desde Linux, se deberá exportar el sistema de archivos utilizando la opción `sync`.

2. Acceso exclusivo garantizado a archivos

   Para sincronizar varios gestores de cola, se necesita un mecanismo para que un gestor de colas obtenga un bloqueo exclusivo en un archivo.

3. Liberar bloqueos en caso de anomalía

   Si se produce alguna anomalía en un gestor de colas, o falla la comunicación con el sistema de archivos, los archivos que el gestor de colas bloquea tienen que desbloquearse y volver a estar disponibles para otros procesos sin esperar a que el gestor de colas vuelva a conectarse al sistema de archivos.

Un sistema de archivos compartidos debe cumplir estos requisitos para que IBM MQ funcione de forma fiable. Si no, los datos y registros del gestor de colas se corrompen cuando se utiliza un sistema de archivos compartidos en una configuración de gestor de colas multiinstancia.

Para gestores de colas de varias instancias en Microsoft Windows, se debe acceder al almacenamiento en red mediante el protocolo SMB (Server Message Block) utilizado por las redes de Microsoft Windows . El cliente SMB (Server Message Block) no cumple los requisitos de IBM MQ para bloquear la semántica en plataformas distintas de Microsoft Windows, por lo que los gestores de colas multiinstancia que se ejecutan en plataformas distintas de Microsoft Windows no deben utilizar SMB (Server Message Block) como sistema de archivos compartido.

Para los gestores de colas multiinstancia de otras plataformas soportadas, se debe acceder al almacenamiento mediante un protocolo del sistema de archivos de red que sea compatible con Posix y dé soporte al bloqueo basado en leasing. Network File System 4 cumple este requisito. Los sistemas de archivos más antiguos, tales como NFS Versión 3, que no poseen un mecanismo fiable para liberar bloqueos tras una anomalía, no se deben utilizar con los gestores de colas multiinstancia.

Comprueba si el sistema de archivos compartidos cumple los requisitos

Es necesario que el usuario compruebe si el sistema de archivos compartidos que va a utilizar cumple estos requisitos. Además debe comprobar si el sistema de archivos se ha configurado correctamente en cuestiones de fiabilidad. Los sistemas de archivos compartidos ofrecen a veces opciones de configuración para mejorar el rendimiento en cuestiones de fiabilidad.

Para obtener más información, consulte Declaración de prueba en sistemas de archivos del gestor de colas multiinstancia de IBM MQ.

En circunstancias normales IBM MQ funciona correctamente cuando los atributos se almacenan en la caché y no es necesario inhabilitar la caché, por ejemplo, estableciendo NOAC en un montaje de tipo NFS. Colocar en la memoria caché los atributos puede provocar problemas cuando varios clientes del

sistema de archivos compiten por tener acceso de escritura al mismo archivo en el servidor del sistema de archivos, ya que es posible que los atributos que se encuentran en la memoria caché utilizada por cada cliente no sean los mismos atributos que hay en el servidor. Un ejemplo de los archivos a los que se accede de esta manera son los registros de errores del gestor de colas de un gestor de colas varias instancias. Los registros de errores del gestor de colas las pueden grabar tanto una instancia de gestor de colas activa como una de reserva, y los atributos de archivo colocados en la memoria caché pueden provocar que los registros de errores crezcan más de lo previsto, antes de que se produzca el aplazamiento de los archivos.

Si desea obtener ayuda para comprobar el sistema de archivos que ejecuta la tarea, consulte Verificación del comportamiento del sistema de archivos compartidos. Esta tarea comprueba que el sistema de archivos deseado cumple los requisitos 2 y 3. Es necesario que el usuario verifique el requisito 1 en la documentación del sistema de archivos compartidos o examinado los datos del registro del disco.

Los fallos de disco puede provocar errores al grabar en el disco, sobre los que IBM MQ informa como captura de datos en primer error (FFDC). Puede ejecutar el comprobador de sistema de archivos de su sistema operativo para comprobar si en el sistema de archivos compartidos existe algún fallo de disco. Por ejemplo:

- Linux AIX En AIX and Linux, el comprobador del sistema de archivos se denomina fsck.
- Windows En las plataformas Windows, el comprobador del sistema de archivos se llama CHKDSK, o SCANDISK.

Seguridad del servidor NFS

Notas:

- No puede utilizar las opciones **nosuid** o **noexec** para un punto de montaje que se utiliza para contener el directorio de instalación de IBM MQ . Esto se debe a que IBM MQ incluye programas ejecutables setuid/setgid y no se debe impedir que se ejecuten correctamente.
- Cuando coloca datos del gestor de colas sólo en un servidor del sistema de archivos de red (NFS), puede utilizar las tres opciones siguientes con el mandato de montaje para que el sistema sea seguro, sin que ello afecte negativamente a la ejecución del gestor de colas:

  noexec
  Con esta opción, no se pueden ejecutar archivos binarios en el NFS, lo que impide que un usuario remoto ejecute código no deseado en el sistema.

  nosuid
  Con esta opción, no se pueden utilizar los bits set-user-identifier y set-group-identifier, lo que impide que un usuario remoto obtenga mayores privilegios.

  nodev
  Con esta opción, no se pueden utilizar ni definir dispositivos especiales de bloque o caracteres, lo que impide a un usuario remoto salir de una cárcel chroot.

IBM i Linux AIX *Verificación del comportamiento del sistema de archivos compartidos en Multiplatforms*

Ejecute **amqmfsck** para comprobar si un sistema de archivos compartido en AIX, Linux o IBM i cumple los requisitos para almacenar los datos del gestor de colas de un gestor de colas de varias instancias. (El único requisito para una configuración de Windows es que utilice SMB 3 para el suministro de almacenamiento compartido.)

Antes de empezar

Necesita tener un servidor con almacenamiento en red y otros dos conectados que tengan instalado IBM MQ. Debe tener derechos de administrador (root) para configurar el sistema de archivos y ser un administrador de IBM MQ para ejecutar **amqmfsck**.

Acerca de esta tarea

"Requisitos para los sistemas de archivos compartidos en Multiplatforms" en la página 117 describe los requisitos para el sistema de archivos utilizando un sistema de archivos compartidos con gestores de colas multiinstancia. La nota técnica de IBM MQ Declaración de prueba para sistemas de archivos de gestor de colas multiinstancia de IBM MQ lista los sistemas de archivos compartidos con los que ya se ha probado IBM. El procedimiento en esta tarea describe cómo probar un sistema de archivos para ayudarle a evaluar si un sistema de archivos no listado mantiene la integridad de los datos.

La anomalía de un gestor de colas multiinstancia se puede desencadenar mediante anomalías de hardware o software, incluidos problemas de red que impiden que el gestor de colas escriba datos o archivos de registro. Principalmente, está interesado en provocar anomalías en el servidor de archivos. Pero también debe hacer que los servidores de IBM MQ fallen, para probar si los bloqueos se han liberado correctamente. Para estar seguros en un sistema de archivos compartidos, pruebe todas las anomalías siguientes, así como cualquier otra que sea específica del entorno:

1. Cierre del sistema operativo en el servidor de archivos incluida la sincronización entre los discos.
2. Detención del sistema operativo en el servidor de archivos sin sincronización entre discos.
3. Pulsación del botón de restablecimiento en cada uno de los servidores.
4. Extracción del cable de red de cada uno de los servidores.
5. Extracción del cable de alimentación de cada uno de los servidores.
6. Cierre de cada uno de los servidores.

Cree el directorio en un almacenamiento en red que vaya a utilizar para compartir registros y datos del gestor de colas. El propietario del directorio debe ser un administrador de IBM MQ o, en otras palabras, un miembro del grupo `mqm` en AIX and Linux. El usuario que ejecute las pruebas debe tener derechos de administrador de IBM MQ.

Utilice el ejemplo de exportación y montaje de un sistema de archivos en Creación de un gestor de colas de varias instancias en Linux o Creación de un gestor de colas de varias instancias utilizando la duplicación de diario y NetServer en IBM i como ayuda para configurar el sistema de archivos. Sistemas de archivos diferentes necesitan configurarse de forma diferente. Lea la documentación del sistema de archivos.

Nota: Ejecute el programa de ejemplo de IBM MQ MQI client **amqsfhac** en paralelo con **amqmfsck** para demostrar que un gestor de colas mantiene la integridad del mensaje durante una anomalía.

Procedimiento

En cada una de las comprobaciones, provoque todos los errores de la lista anterior mientras el comprobador del sistema de archivos está en ejecución. Si tiene la intención de ejecutar **amqsfhac** al mismo tiempo que **amqmfsck**, realice la tarea, "Ejecución de amqsfhac para probar la integridad del mensaje" en la página 125 paralelamente a esta tarea.

1. Monte el directorio exportado en los dos servidores de IBM MQ.

   En el servidor del sistema de archivos, cree un directorio compartido `shared`, y un subdirectorio para guardar los datos para gestores de colas multiinstancia, `qmdata`. Para ver un ejemplo de configuración de un directorio compartido para gestores de colas de varias instancias en Linux, consulte Creación de un gestor de colas de varias instancias en Linux
2. Compruebe el comportamiento del sistema de archivos básico.

   En un servidor de IBM MQ, ejecute el comprobador del sistema de archivos sin parámetros.

   En el servidor 1 IBM MQ:

   ```
   amqmfsck /shared/qmdata
   ```
3. Compruebe escribir al mismo tiempo en el mismo directorio desde ambos servidores de IBM MQ.

   En los dos servidores de IBM MQ, ejecute el comprobador del sistema de archivos simultáneamente con la opción `-c`.

En el servidor 1 IBM MQ:

```
amqmfsck -c /shared/qmdata
```

en el servidor 2 IBM MQ:

```
amqmfsck -c /shared/qmdata
```

4. Compruebe la espera y la liberación de bloqueos en ambos servidores de IBM MQ.

   En los dos servidores de IBM MQ, ejecute el comprobador del sistema de archivos simultáneamente con la opción `-w`.

   En el servidor 1 IBM MQ:

   ```
   amqmfsck -w /shared/qmdata
   ```

   en el servidor 2 IBM MQ:

   ```
   amqmfsck -w /shared/qmdata
   ```

5. Compruebe la integridad de los datos.

   a) Formatee el archivo de pruebas.

   Cree un gran archivo en el directorio que se está probando. El archivo se formatea para que se puedan completar correctamente las fases posteriores. El archivo debe ser lo suficientemente grande para tener el tiempo suficiente para interrumpir la segunda fase para simular la anomalía. Intente el valor predeterminado de 262144 páginas (1 GB). El programa reduce automáticamente este valor predeterminado en sistemas de archivos lentos para que el formateo se complete en aproximadamente 60 segundos.

   En el servidor 1 IBM MQ:

   ```
   amqmfsck -f /shared/qmdata
   ```

   El servidor responde con los mensajes siguientes:

   ```
   Formatting test file for data integrity test.

   Test file formatted with 262144 pages of data.
   ```

   b) Escriba datos en el archivo de prueba utilizando el comprobador del sistema de archivos mientras se produce un error.

   Ejecute el programa de prueba en dos servidores al mismo tiempo. Iniciar el programa de prueba en el servidor que va a experimentar la anomalía y, a continuación, inicie el programa de prueba en el servidor de que va a sobrevivir a la anomalía. Cause la anomalía que está investigando.

   El primer programa de pruebas se detiene con un mensaje de error. El segundo programa de prueba obtiene el bloqueo en el archivo de prueba y escribe datos en el archivo de prueba empezando por el primer programa de prueba que ha notificado el error. Deje que el segundo programa de prueba se ejecute hasta el final.

Tabla 10. Ejecute la comprobación de la integridad de los datos en dos servidores al mismo tiempo

Servidor 1 de IBM MQ	**Servidor 2 de IBM MQ**
`amqmfsck -a /shared/qmdata`	

Tabla 10. Ejecute la comprobación de la integridad de los datos en dos servidores al mismo tiempo (continuación)

Servidor 1 de IBM MQ	**Servidor 2 de IBM MQ**
	`amqmfsck -a /shared/qmdata`

`Please start this program on a second machine with the same parameters.`
 	
`File lock acquired.`
 	
`Start a second copy of this program with the same parameters on another server.`
 	
`Writing data into test file.`
 	
`To increase the effectiveness of the test, interrupt the writing by ending the process, temporarily breaking the network connection to the networked storage, rebooting the server or turning off the power.`
	`Waiting for lock...`
	
	`Waiting for lock...`
	
	`Waiting for lock...`
	
	`Waiting for lock...`
	
	`Waiting for lock...`
	
	`Waiting for lock...`

Turn the power off here.	

	`File lock acquired.`
	
	`Reading test file`
	
	`Checking the integrity of the data read.`
	
	`Appending data into the test file after data already found.`
	
	`The test file is full of data. It is ready to be inspected for data integrity.`

El tiempo de la prueba depende del comportamiento del sistema de archivos. Por ejemplo, normalmente un sistema de archivos tarda entre 30 y 90 segundos en liberar los bloqueos de archivo obtenidos por el primer programa tras una interrupción de la alimentación. Si tiene muy poco tiempo para introducir la anomalía antes de que el primer programa de prueba haya llenado el archivo, utilice la opción `-x` de **amqmfsck** para suprimir el archivo de prueba. Vuelva a intentar la prueba desde el principio con un archivo de prueba más grande.

c) Verifique la integridad de los datos en el archivo de prueba.

en el servidor 2 IBM MQ:

```
amqmfsck -i /shared/qmdata
```

El servidor responde con los mensajes siguientes:

```
File lock acquired
```

```
Reading test file checking the integrity of the data read.

The data read was consistent.

The tests on the directory completed successfully.
```

6. Suprima los archivos de prueba.

   en el servidor 2 IBM MQ:

```
amqmfsck -x /shared/qmdata
Test files deleted.
```

   El servidor responde con el mensaje:

```
Test files deleted.
```

Resultados

El mandato devuelve un código de salida de cero si la prueba finaliza correctamente; de lo contrario un código que no es cero.

Ejemplos

El primer conjunto de tres ejemplos muestran el mandato que produce una salida mínima.

Prueba correcta del bloqueo de archivos básico en un servidor

```
> amqmfsck /shared/qmdata
The tests on the directory completed successfully.
```

Error de prueba del bloqueo de archivos básico en un servidor

```
> amqmfsck /shared/qmdata
AMQ6245: Error Calling 'write()[2]' on file '/shared/qmdata/amqmfsck.lck' error '2'.
```

Prueba correcta del bloqueo en dos servidores

Tabla 11. Prueba correcta del bloqueo en dos servidores

Servidor 1 de IBM MQ	Servidor 2 de IBM MQ

`> amqmfsck -w /shared/qmdata`
`Please start this program on a second`
`machine with the same parameters.`
`Lock acquired.`
`Press Return`
`or terminate the program to release the lock.`	

	`> amqmfsck -w /shared/qmdata` `Waiting for lock...`
`[ Return pressed ]` `Lock released.`	

	`Lock acquired.`
	`The tests on the directory completed`
	`successfully`

El segundo conjunto de tres ejemplos muestra los mismos mandatos utilizados en modalidad detallada.

Prueba correcta del bloqueo de archivos básico en un servidor

```
> amqmfsck -v /shared/qmdata
System call: stat("/shared/qmdata")'
System call: fd = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fchmod(fd, 0666)
System call: fstat(fd)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: write(fd)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd1 = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fcntl(fd1, F_SETLK, F_RDLCK)
System call: fd2 = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fcntl(fd2, F_SETLK, F_RDLCK)
System call: close(fd2)
System call: write(fd1)
System call: close(fd1)
The tests on the directory completed successfully.
```

Error de prueba del bloqueo de archivos básico en un servidor

```
> amqmfsck -v /shared/qmdata
System call: stat("/shared/qmdata")
System call: fd = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fchmod(fd, 0666)
System call: fstat(fd)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: write(fd)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fcntl(fd, F_SETLK, F_RDLCK)
System call: fdSameFile = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fcntl(fdSameFile, F_SETLK, F_RDLCK)
System call: close(fdSameFile)
System call: write(fd)
AMQxxxx: Error calling 'write()[2]' on file '/shared/qmdata/amqmfsck.lck', errno 2
(Permission denied).
```

Prueba correcta del bloqueo en dos servidores

Tabla 12. Bloqueo correcto en dos servidores en modalidad detallada

Servidor 1 de IBM MQ	**Servidor 2 de IBM MQ**

`> amqmfsck -wv /shared/qmdata`
`Calling 'stat("/shared/qmdata")'`
`Calling 'fd = open("/shared/qmdata/`
`amqmfsck.lkw",`
`O_EXCL | O_CREAT | O_RDWR, 0666)'`
`Calling 'fchmod(fd, 0666)'`
`Calling 'fstat(fd)'`
`Please start this program on a second`
`machine with the same parameters.`
`Calling 'fcntl(fd, F_SETLK, F_WRLCK)'`
`Lock acquired.`
`Press Return`
`or terminate the program to release the lock.`	

	`> amqmfsck -wv /shared/qmdata`
	`Calling 'stat("/shared/qmdata")'`
	`Calling 'fd = open("/shared/qmdata/`
	`amqmfsck.lkw",`
	`O_EXCL | O_CREAT | O_RDWR,0666)'`
	`Calling 'fd = open("/shared/qmdata/amqmfsck.lkw,`
	`O_RDWR, 0666)'`
	`Calling 'fcntl(fd, F_SETLK, F_WRLCK)`
	`'Waiting for lock...`

Tabla 12. Bloqueo correcto en dos servidores en modalidad detallada (continuación)

Servidor 1 de IBM MQ	Servidor 2 de IBM MQ

`[ Return pressed ]`
`Calling 'close(fd)'`
`Lock released.`	

	`Calling 'fcntl(fd, F_SETLK, F_WRLCK)'`
	`Lock acquired.`
	`The tests on the directory completed successfully`

Referencia relacionada

Programas de ejemplo de alta disponibilidad

Multi

Ejecución de amqsfhac para probar la integridad del mensaje

Ejecute el programa de ejemplo de IBM MQ MQI client **amqsfhac** en paralelo con **amqmfsck** para demostrar que un gestor de colas mantiene la integridad del mensaje durante una anomalía.

Antes de empezar

Se necesitan cuatro servidores para esta prueba. Dos servidores para el gestor de colas multiinstancia, uno para el sistema de archivos y uno para ejecutar **amqsfhac** como una aplicación de IBM MQ MQI client.

Siga el paso "1" en la página 120 en "Verificación del comportamiento del sistema de archivos compartidos en Multiplatforms" en la página 119 para configurar el sistema de archivos para un gestor de colas multiinstancia.

Acerca de esta tarea

El programa de ejemplo de IBM MQ MQI client **amqsfhac** comprueba que un gestor de colas que utiliza almacenamiento en red mantiene integridad de datos tras una anomalía. Ejecute **amqsfhac** en paralelo con **amqmfsck** para demostrar que un gestor de colas mantiene la integridad de los mensajes durante una anomalía.

Procedimiento

1. Cree un gestor de colas multiinstancia en otro servidor, `QM1`, utilizando el sistema de archivos que ha creado en el paso "1" en la página 120 en Procedimiento.

   Consulte Crear un gestor de colas multiinstancia.
2. Inicie el gestor de colas en ambos servidores haciendo que estén altamente disponibles.

   En el servidor 1:

   ```
   strmqm -x QM1
   ```

   En el servidor 2:

   ```
   strmqm -x QM1
   ```
3. Configure la conexión con el cliente para ejecutar**amqsfhac**.

   a) Utilice el procedimiento de *Verificación de una instalación de IBM MQ* para la plataforma, o plataformas, que la empresa utiliza para configurar una conexión de cliente o los scripts de ejemplo de Ejemplos de clientes reconectables.

b) Modifique el cliente de canal para que tenga dos direcciones IP, correspondientes a los dos servidores que ejecutan QM1.

En el script de ejemplo, modifique:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME('LOCALHOST(2345)') QMNAME(QM1) REPLACE
```

A:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME('server1(2345),server2(2345)') QMNAME(QM1) REPLACE
```

donde server1 y server2 son los nombres de host de los dos servidores, y 2345 es el puerto en el que está escuchando el escucha de canal. Normalmente el valor predeterminado es 1414. Puede utilizar1414 con la configuración del escucha predeterminado.

4. Cree dos colas locales en QM1 para la prueba.
Ejecute el script MQSC siguiente:

```
DEFINE QLOCAL(TARGETQ) REPLACE
DEFINE QLOCAL(SIDEQ) REPLACE
```

5. Pruebe la configuración con **amqsfhac**

```
amqsfhac QM1 TARGETQ SIDEQ 2 2 2
```

6. Pruebe la integridad del mensaje mientras prueba la integridad del sistema de archivos.

Ejecute **amqsfhac** durante el paso “5” en la página 121 de “Verificación del comportamiento del sistema de archivos compartidos en Multiplatforms” en la página 119.

```
amqsfhac QM1 TARGETQ SIDEQ 10 20 0
```

Si detiene la instancia de gestor de colas activa, **amqsfhac** se vuelve a conectar con la otra instancia del gestor de colas una vez que está activa. Reinicie la instancia del gestor de colas detenido de nuevo, para que pueda invertir la anomalía en la próxima prueba. Necesitará incrementar probablemente el número de iteraciones basadas en la experimentación con su entorno para que el programa de prueba se ejecute el tiempo suficiente para que se produzca la anomalía.

Resultados

Un ejemplo de ejecución de **amqsfhac** en el paso “6” en la página 126 se muestra en el ejemplo siguiente. En este ejemplo, la prueba se realiza satisfactoriamente.

```
Sample AMQSFHAC start
qmname = QM1
qname = TARGETQ
sidename = SIDEQ
transize = 10
iterations = 20
verbose = 0
Iteration 0
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
Resolving MQRC_CALL_INTERRUPTED
MQGET browse side tranid=14 pSideinfo->tranid=14
Resolving to committed
Iteration 7
Iteration 8
```

```
Iteration 9
Iteration 10
Iteration 11
Iteration 12
Iteration 13
Iteration 14
Iteration 15
Iteration 16
Iteration 17
Iteration 18
Iteration 19
Sample AMQSFHAC end
```

Si la prueba detecta un problema, la salida informará de la anomalía. En algunas ejecuciones de prueba, `MQRC_CALL_INTERRUPTED` puede informar de "`Resolving to backed out`". No hay ninguna diferencia en el resultado. El resultado depende de si el almacenamiento del archivo en red confirmó la escritura en disco se confirmó antes o después de que tuviera lugar la anomalía.

Referencia relacionada

amqmfsck (comprobación del sistema de archivos)

Programas de ejemplo de alta disponibilidad

Multi Compartición de archivos de IBM MQ en Multiplatforms

A algunos archivos IBM MQ sólo puede acceder un gestor de colas activo y otros son compartidos.

Los archivos de IBM MQ se dividen en archivos de programas y archivos de datos. Los archivos de programas se instalan normalmente de forma local en cada servidor que ejecuta IBM MQ. Los gestores de colas comparten acceso a archivos de datos y directorios en el directorio de datos predeterminado. Necesita acceso exclusivo a sus propios árboles de directorios de gestores de colas que se encuentran en cada directorio `qmgrs` y `log` que se muestra en la Figura 32 en la página 127.

La Figura 32 en la página 127 es una vista de nivel superior de la estructura de directorios de IBM MQ. Muestra los directorios que pueden compartirse entre los gestores de colas y remotos. Los detalles varían según la plataforma. Las líneas con puntos indican vías de acceso que pueden configurarse.

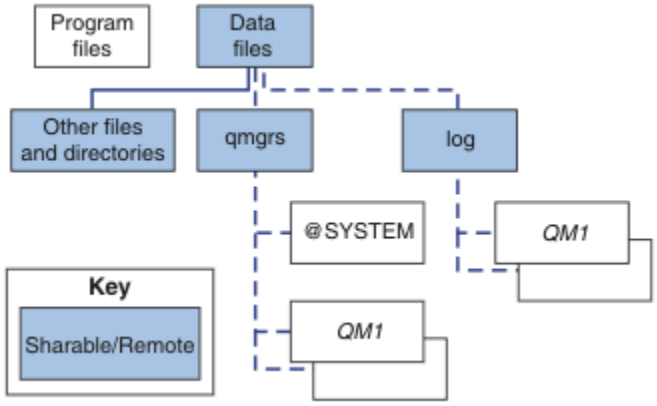


Figura 32. Visión general de la estructura de directorios de IBM MQ

Archivos de programas

El directorio de los archivos de programas se deja normalmente en la ubicación predeterminada, es local y la comparten todos los gestores de colas en el servidor.

Archivos de datos

El directorio de archivos de datos suele ser local en la ubicación predeterminada, `/var/mqm` en sistemas AIX and Linux y configurable en la instalación en Windows. Se comparte entre los gestores de colas. Puede hacer la ubicación predeterminada remota, pero no la comparta entre las diferentes instalaciones de IBM MQ. El atributo `DefaultPrefix` en la configuración IBM MQ apunta a esta vía de acceso.

qmgrs

Existen dos formas alternativas de especificar la ubicación de los datos del gestor de colas.

Utilización del atributo Prefix

El atributo **Prefix** especifica la ubicación del directorio qmgrs. IBM MQ forma el nombre del directorio del gestor de colas a partir del nombre del gestor de colas y lo crea como un subdirectorio del directorio qmgrs.

El atributo **Prefix** se encuentra en la sección QueueManager del archivo mqs.ini y se hereda del valor del atributo **DefaultPrefix** de la stanza Todos los gestores de colas. De manera predeterminada, para que la administración sea más sencilla, los gestores de colas comparten normalmente el mismo directorio qmgrs.

Si se cambia la ubicación del directorio qmgrs de cualquier gestor de colas, necesita cambiar el valor del atributo **Prefix**.

El atributo **Prefix** para el directorio QM1 en Figura 32 en la página 127 para una plataforma AIX and Linux es este:

```
Prefix=/var/mqm
```

Utilización del atributo DataPath

El atributo **DataPath** especifica la ubicación del directorio de datos del gestor de colas.

El atributo **DataPath** especifica vía de acceso completa, que incluye el nombre del directorio de datos del gestor de colas. Con el atributo **DataPath** ocurre lo contrario de lo que ocurre con el atributo **Prefix**, que especifica una vía de acceso incompleta al directorio de datos del gestor de colas.

El atributo **DataPath** , si se especifica, se encuentra en la stanza QueueManager del archivo mqs.ini . Si se ha especificado, tiene preferencia respecto al valor del atributo **Prefix**.

Si se cambia la ubicación del directorio de datos del gestor de colas de cualquier gestor de colas, debe cambiar el valor del atributo DataPath.

El atributo DataPath para el directorio QM1 en Figura 32 en la página 127, para una plataforma Linux o AIX , es el siguiente:

```
DataPath=/var/mqm/qmgrs/QM1
```

log

El directorio de registro se especifica por separado para cada gestor de colas en la stanza de registro en la configuración del gestor de colas. La configuración del gestor de colas está en qm.ini.

Subdirectorios *DataPath*/*QmgrName*/@IPCC

Los subdirectorios *DataPath*/*QmgrName*/@IPCC están en la vía de acceso del directorio compartido. Se utilizan para construir la vía de acceso de directorio para los objetos del sistema de archivos IPC. Necesitan distinguir el espacio de nombres de un gestor de colas cuando un gestor de colas se comparte entre los sistemas.

Los objetos del sistema de archivos IPC deben distinguirse por el sistema. Para cada sistema en el que se ejecute el gestor de colas, se añade un subdirectorio a la vía de acceso del directorio; consulte la Figura 33 en la página 128.

```
DataPath/QmgrName/@IPCC/esem/myHostName/
```

Figura 33. Subdirectorio IPC de ejemplo

myHostName corresponde a los primeros 20 caracteres del nombre de host que devuelve el sistema operativo. En algunos sistemas, el nombre de host podría tener hasta 64 caracteres de longitud antes del truncamiento. El valor generado de *myHostName* puede causar un problema por dos razones:

1. Los primeros 20 caracteres no son exclusivos.
2. El nombre de host lo genera un algoritmo de DHCP que no siempre asigna el mismo nombre de host a un sistema.

En estos casos, establezca *myHostName* utilizando la variable de entorno **MQS_IPC_HOST**; consulte Figura 34 en la página 129.

```
export MQS_IPC_HOST= myHostName
```

*Figura 34. Ejemplo: establecimiento de **MQS_IPC_HOST***

Otros archivos y directorios

Otros archivos y directorios, como el directorio que contiene los archivos de rastreo o el registro de errores comunes, se almacenan y guardan normalmente en el sistema de archivos locales.

Con soporte de sistemas de archivos compartidos, IBM MQ gestiona el acceso exclusivo a estos archivos utilizando bloqueos de sistema de archivos. Un bloqueo de sistema de archivos permite que sólo una instancia de un gestor de colas particular esté activa a la vez.

Cuando se inicia la primera instancia de un gestor de colas en particular, esta toma posesión de su directorio de gestor de colas. Si inicia una segunda instancia, sólo puede tomar posesión si se ha detenido la primera instancia. Si se sigue ejecutando el primer gestor de colas, la segunda instancia no se puede iniciar, e informa al gestor de colas que se está ejecutando en otro sitio. Si se ha detenido el primer gestor de colas, el segundo toma posesión de los archivos del gestor de colas y pasa a ser el gestor de colas que se está ejecutando.

Se puede automatizar el procedimiento del segundo gestor de colas para que sustituya al primero. Inicie un gestor de colas con la opción `strmqm -x`, la cual permite que otro gestor de colas sustituya al primero. El segundo gestor de colas esperaría hasta que los archivos del gestor de colas quedasen desbloqueados antes de intentar tomar posesión de los archivos del gestor de colas e iniciarse.

Linux AIX Estructura de directorios en sistemas AIX and Linux

La estructura de directorios de IBM MQ en sistemas AIX and Linux puede asignarse a diferentes sistemas de archivos para conseguir una gestión más sencilla, mejor rendimiento y mayor fiabilidad.

Utilice la estructura de directorios flexible de IBM MQ para sacar partido de los sistemas de archivos compartidos para ejecutar gestores de colas multiinstancia.

Utilice el mandato **crtmqm** *QM1* para crear la estructura de directorios que se muestra en Figura 35 en la página 130 , donde R es el release del producto. Es una estructura de directorios típica para un gestor de colas creado en un sistema IBM MQ . Se han omitido algunos directorios, archivos y valores de atributos .ini para una mayor claridad, y otro nombre del gestor de colas puede modificarse cortándose. Los nombres de los sistemas de archivos varían en diferentes sistemas.

En una instalación típica, cada gestor de colas que crea apunta a directores `log` y `qmgrs` comunes en el sistema de archivos local. En una configuración de varias instancias, los directorios `log` y `qmgrs` están en un sistema de archivos de red compartido con otra instalación de IBM MQ.

Figura 35 en la página 130 muestra la configuración predeterminada para IBM MQ v7.R en AIX donde R es el release del producto. Si desea ver ejemplos de configuraciones multiinstancia alternativas, consulte “Configuraciones de directorios de ejemplo en sistemas AIX and Linux” en la página 135.

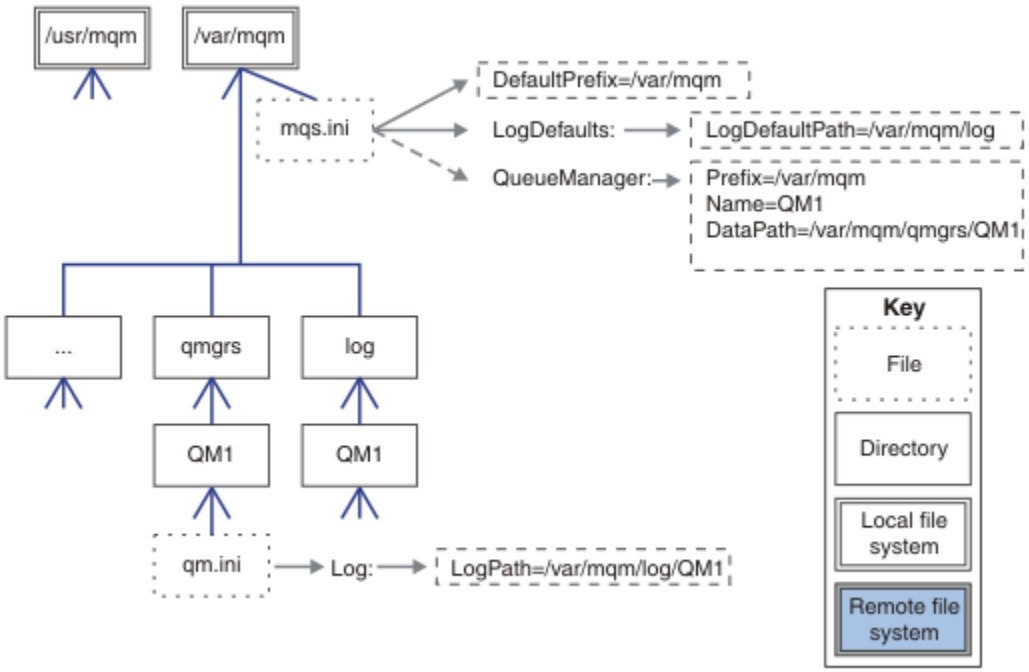


Figura 35. Ejemplo de estructura de directorios predeterminada de IBM MQ para los sistemas AIX and Linux

El producto se instala en `/usr/mqm` en AIX y `/opt/mqm` en los otros sistemas, de forma predeterminada. Los directorios de trabajo se instalan en el directorio `/var/mqm`.

Nota: Si ha creado el sistema de archivos `/var/mqm` antes de instalar IBM MQ, asegúrese de que el usuario mqm tiene permisos de directorio completos, por ejemplo, modalidad de archivo 755.

Nota: El directorio `/var/mqm/errors` debe ser un sistema de archivos independiente para evitar que los FFDC producidos por el gestor de colas rellenen el sistema de archivos que contiene `/var/mqm`.

Consulte Creación de sistemas de archivos en sistemas AIX and Linux para obtener más información.

Los directorios `log` y `qmgrs` se muestran en sus ubicaciones predeterminadas tal como se definen en los valores predeterminados de los atributos `LogDefaultPath` y `DefaultPrefix` en el archivo `mqs.ini`. Cuando se crea un gestor de colas, de forma predeterminada se crea el directorio de datos del gestor de colas en *`DefaultPrefix`*`/qmgrs` y el directorio del archivo de registro en *`LogDefaultPath`*`/log`. `LogDefaultPath` y `DefaultPrefix` sólo se dan cuando los gestores de colas y archivos de registro se crean de forma predeterminada. La ubicación real de un directorio del gestor de colas se guarda en el archivo `mqs.ini` y la ubicación del directorio del archivo de registro se guarda en el archivo `qm.ini`.

El directorio del archivo de registro para un gestor de colas se define en el archivo `qm.ini` en el atributo `LogPath`. Utilice la opción `-ld` en el mandato **`crtmqm`** para establecer el atributo `LogPath` para un gestor de colas; por ejemplo, **`crtmqm`** `-ld` *`LogPath`* `QM1` . Si se omite el parámetro `ld`, se utiliza entonces el valor de `LogDefaultPath`.

El directorio de datos del gestor de colas se define en el atributo `DataPath` de la stanza `QueueManager` del archivo `mqs.ini`. Utilice la opción `-md` en el mandato **`crtmqm`** para establecer `DataPath` para un gestor de colas; por ejemplo, **`crtmqm`** `-md` *`DataPath`* `QM1` . Si se omite el parámetro `md`, se utiliza entonces el valor del atributo `DefaultPrefix` o `Prefix`. `Prefix` tiene preferencia sobre `DefaultPrefix`.

Normalmente, se crea `QM1` especificando tanto el directorio de registro como el de datos en un mismo mandato.

```
crtmqm
```

```
-md DataPath -ld
LogPath QM1
```

Puede modificar la ubicación de un registro del gestor de colas y de los directorios de datos de un gestor de colas existente editando los atributos `DataPath` y `LogPath` en el archivo `qm.ini` cuando se detiene el gestor de colas.

La vía de acceso al directorio `errors`, como las vías de acceso a todos los demás directorios de `/var/mqm`, no se puede modificar. No obstante, los directorios se pueden montar en diferentes sistemas o unirlos simbólicamente a directorios diferentes.

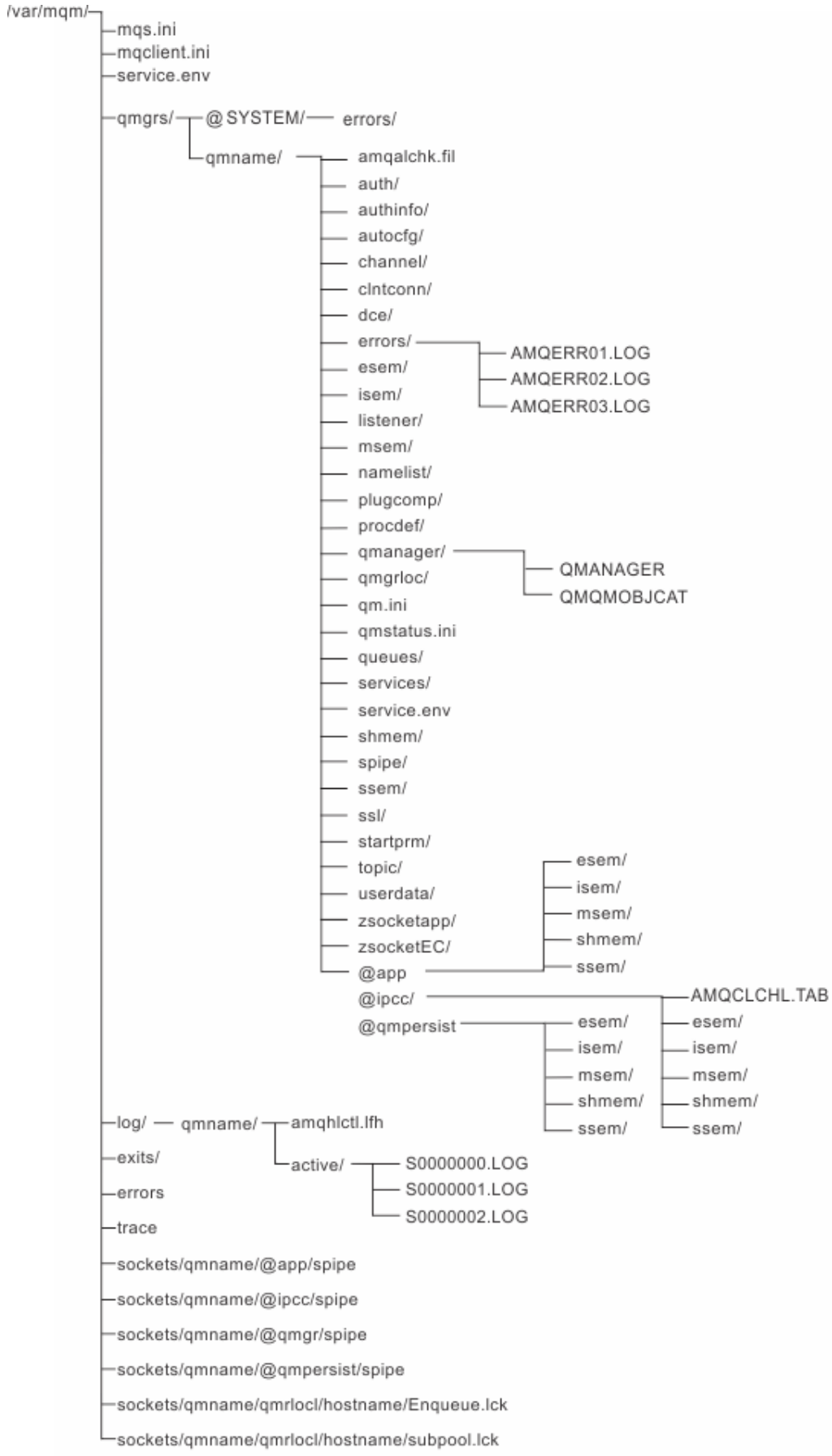
Linux AIX *Contenido de los directorios en los sistemas AIX and Linux*

Contenido de los directorios asociados a un gestor de colas.

Para obtener más información sobre la ubicación de los archivos del producto, consulte Elección de una ubicación de instalación

Para obtener más información sobre otras configuraciones de directorios, consulte “Planificación del soporte del sistema de archivos en Multiplatforms” en la página 117.

La siguiente estructura de directorios es representativa de IBM MQ después de que un gestor de colas haya estado en uso durante algún tiempo. La estructura real depende de las operaciones que se han producido en el gestor de colas.


/var/mqm/
mqs.ini
mqclient.ini
service.env
qmgrs/
@SYSTEM/
errors/
qmname/
amqalchk.fil
auth/
authinfo/
autocfg/
channel/
clntconn/
dce/
errors/
AMQERR01.LOG
AMQERR02.LOG
AMQERR03.LOG
esem/
isem/
listener/
msem/
namelist/
plugcomp/
procdef/
qmanager/
QMANAGER
QMQMOBJCAT
qmgrloc/
qm.ini
qmstatus.ini
queues/
services/
service.env
shmem/
spipe/
ssem/
ssl/
startprm/
topic/
userdata/
zsocketapp/
zsocketEC/
@app
esem/
isem/
msem/
shmem/
ssem/
@ipcc/
AMQCLCHL.TAB
esem/
isem/
msem/
shmem/
ssem/
@qmpersist
esem/
isem/
msem/
shmem/
ssem/
log/
qmname/
amqhlctl.lfh
active/
S0000000.LOG
S0000001.LOG
S0000002.LOG
exits/
errors
trace
sockets/qmname/@app/spipe
sockets/qmname/@ipcc/spipe
sockets/qmname/@qmgr/spipe
sockets/qmname/@qmpersist/spipe
sockets/qmname/qmrlocl/hostname/Enqueue.lck
sockets/qmname/qmrlocl/hostname/subpool.lck

/var/mqm/

El directorio /var/mqm contiene archivos de configuración y directorios de salida que se aplican a una instalación de IBM MQ como un todo, y no a un gestor de colas individual.

Tabla 13. Contenido documentado del directorio /var/mqm en AIX and Linux	
Nombre de directorio o archivo	**Contenido**
mqs.ini	Archivo de configuración de IBM MQ de toda la instalación que se lee cuando se inicia un gestor de colas. Vía de acceso modificable mediante la variable de entorno **AMQ_MQS_INI_LOCATION**. Asegúrese de que está establecido y exportado en el shell en el que se ejecuta el mandato **strmqm**.
mqclient.ini	Archivo de configuración de cliente predeterminado que leen los programas de IBM MQ MQI client. Vía de acceso modificable mediante la variable de entorno **MQCLNTCF**.
service.env	Contiene variables de entorno del ámbito de máquina para un proceso de servicio. Vía de acceso al archivo fija.
errors/	Registros de errores de ámbito de máquina y archivos FFST. Vía de acceso al directorio fija. Consulte también FFST: sistemas IBM MQ for UNIX y Linux.
sockets/	Contiene información de cada gestor de colas para uso exclusivo del sistema.
trace/	Archivos de rastreo. Vía de acceso al directorio fija.
web/	Directorio de servidor mqweb.
salidas/	Directorio predeterminado que contiene programas de salida de canal de usuario. Ubicación modificable en stanzas ApiExit en el archivo mqs.ini.
exits64/	

/var/mqm/qmgrs/*qmname*/

/var/mqm/qmgrs/*qmname*/ contiene directorios y archivos para un gestor de colas. La instancia del gestor de colas bloquea el directorio para que tenga un acceso exclusivo. La vía de acceso al directorio se puede modificar directamente en el archivo mqs.ini o utilizando la opción **md** del mandato **crtmqm**.

Tabla 14. Contenido documentado del directorio /var/mqm/qmgrs/qmname en AIX and Linux	
Nombre de directorio o archivo	**Contenido**
qm.ini	Archivo de configuración del gestor de colas que se lee cuando se inicia un gestor de colas.

Tabla 14. Contenido documentado del directorio /var/mqm/qmgrs/qmname en AIX and Linux (continuación)	
Nombre de directorio o archivo	**Contenido**
errors/	Anotaciones de errores en el ámbito del gestor de colas. *qmname* = `@system` contiene mensajes relacionados con el canal para un gestor de colas desconocido o no disponible.
@ipcc/ **AMQCLCHL.TAB**	Tabla de control de canales de cliente predeterminada que el servidor de IBM MQ ha creado y que los programas de cliente IBM MQ MQI client leen. Vía de acceso modificable mediante las variables de entorno **MQCHLLIB** y **MQCHLTAB**.
qmanager	Archivo de objeto del gestor de colas: `QMANAGER` Catálogo de objetos del gestor de colas: `QMQMOBJCAT`
authinfo/	Los objetos definidos dentro del gestor de colas están asociados a un archivo en estos directorios. El nombre de archivo coincide aproximadamente con el nombre de definición; consulte Visión general de los nombres de archivo de IBM MQ.
canal/	
clntconn/	
listener/	
lista de nombres/	
procdef/	
colas/	
services/	
temas/	
...	Otros directorios utilizados por IBM MQ, como @ipcc, para modificar sólo con IBM MQ.
datos de usuario/	Se puede utilizar para almacenar el estado persistente de las aplicaciones (lo puede utilizar RDQM al mover gestores de colas a distintos nodos; consulte Almacenamiento del estado de aplicaciones persistentes.)
DataPath**autocfg	Se utiliza para la configuración automática

/var/mqm/log/*qmname*/

`/var/mqm/log/`*qmname*`/` contiene los archivos de registro del gestor de colas. La instancia del gestor de colas bloquea el directorio para que tenga un acceso exclusivo. La vía de acceso se puede modificar en el archivo `qm.ini` o mediante la opción **ld** del mandato **crtmqm**.

Tabla 15. Contenido documentado del directorio /var/mqm/log/qmname en AIX and Linux	
Nombre de directorio o archivo	**Contenido**
amqhlctl.lfh	Archivo de control de anotaciones.
active/	Este directorio contiene los archivos de anotaciones numerados del modo siguiente: S0000000.LOG, S0000001.LOG, S0000002.LOG y así sucesivamente.

/opt/mqm

/opt/mqm es, de forma predeterminada, el directorio de instalación en la mayoría de las plataformas. Consulte “Requisitos de espacio de disco en Multiplatforms” en la página 114 para obtener más información sobre la cantidad de espacio necesario para el directorio de instalación en la plataforma, o plataformas, que la empresa utiliza.

Linux AIX *Configuraciones de directorios de ejemplo en sistemas AIX and Linux*

Ejemplos de otras configuraciones de sistemas de archivos en sistemas AIX and Linux.

Se puede personalizar la estructura de directorios de IBM MQ de diferentes formas para lograr objetivos diferentes.

- Coloque los directorios qmgrs y log en sistemas de archivos compartidos remotos para configurar un gestor de colas de varias instancias.
- Utilizar sistemas de archivos separados para lo directorios de registros y datos y ubique los directorios en diferentes discos para mejorar el rendimiento reduciendo el conflicto de E/S.
- Utilizar dispositivos de almacenamiento más rápidos para directorios que tienen un mayor efecto en el rendimiento. La latencia de dispositivos físicos es normalmente un factor más importante en el rendimiento de mensajería persistente que en un dispositivo que se monta de forma local o remota. La siguiente lista muestra qué directorios se ven más o menos afectados por el rendimiento.
  1. log
  2. qmgrs
  3. Otros directorios, incluido /usr/mqm
- Cree los directorios qmgrs y log en los sistemas de archivos que se asignan al almacenamiento con una buena resistencia, como por ejemplo una matriz de discos redundante, por ejemplo.
- Es mejor almacenar los registros de errores comunes en var/mqm/errors, localmente, en lugar de en un sistema de archivos de red, de modo que se pueda registrar el error relacionado con el sistema de archivos de red.

La Figura 36 en la página 136 es una plantilla a partir de la cual pueden derivarse estructuras de directorios de IBM MQ alternativas. En la plantilla, las líneas con puntos representan vías de acceso que se pueden configurar. En los ejemplos, las líneas de puntos se sustituyen por líneas sólidas que corresponden a la información de configuración almacenada en la variable de entorno AMQ_MQS_INI_LOCATION y en los archivos mqs.ini y qm.ini.

Nota: La información de vía de acceso se muestra tal como aparece en los archivos mqs.ini o qm.ini. Si proporciona parámetros de vía de acceso en el mandato **crtmqm**, omita el nombre del directorio del gestor de colas: IBM MQ añadirá el nombre del gestor de colas a la vía de acceso.

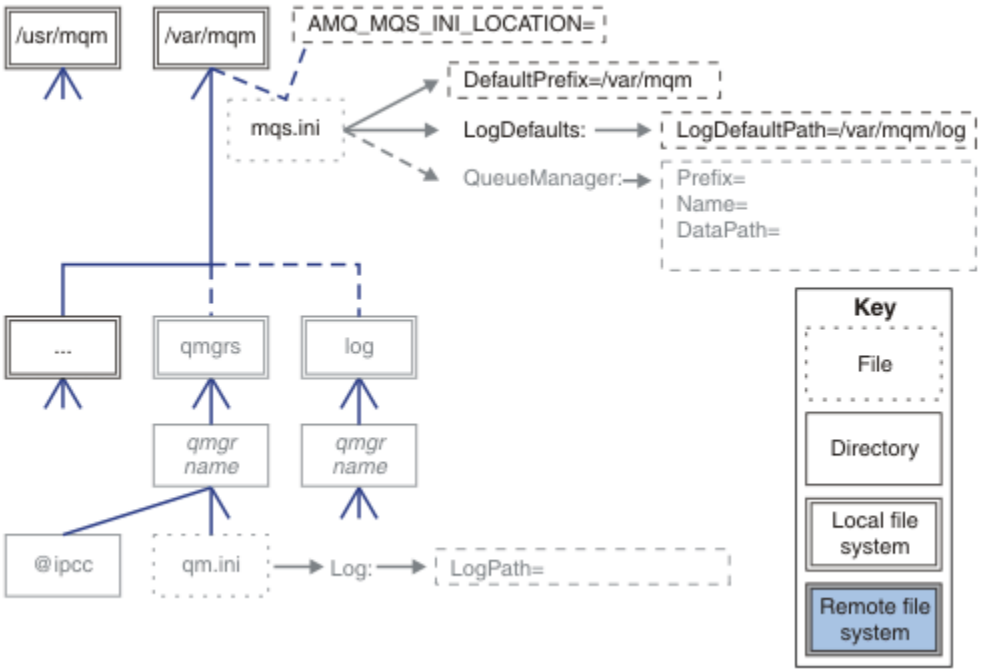


Figura 36. Plantilla modelo de estructura de directorios

Estructura de directorios típica para IBM MQ

La Figura 37 en la página 137 es la estructura de directorios predeterminada creada en la IBM MQ emitiendo el mandato **crtmqm** *QM1*.

El archivo `mqs.ini` tiene una stanza para el gestor de colas de QM1, creada haciendo referencia al valor de `DefaultPrefix`. La stanza `Log` del archivo `qm.ini` tiene un valor para `LogPath`, establecido por referencia a `LogDefaultPath` en `mqs.ini`.

Utilice los parámetros opcionales **crtmqm** para alterar temporalmente los valores predeterminados de `ViaAccesoDatos` y `ViaAccesoRegistro`.

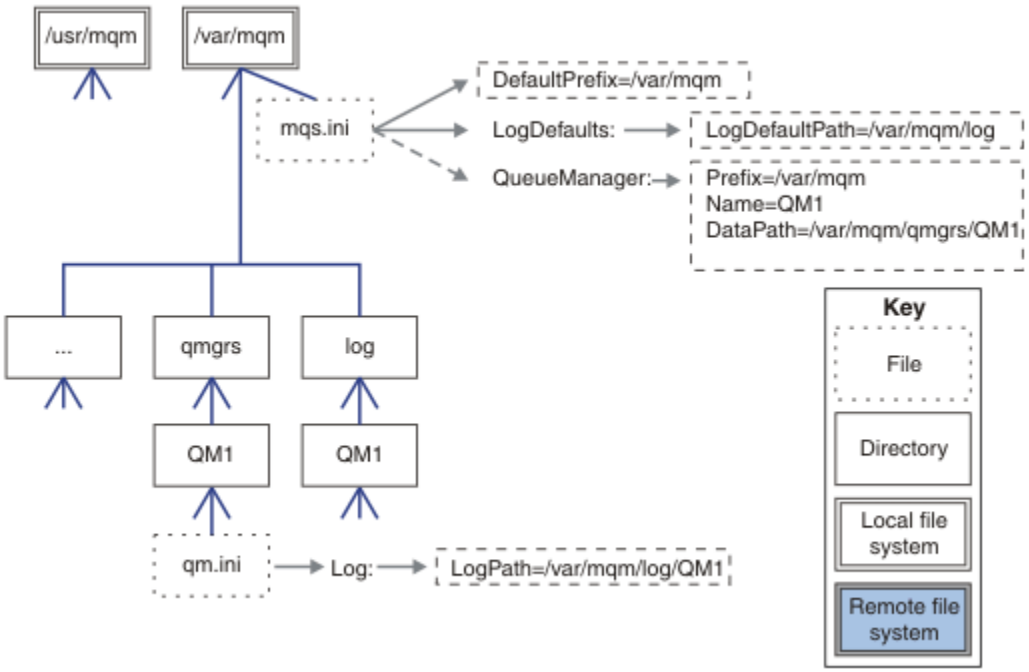


Figura 37. Ejemplo de estructura de directorios predeterminada de IBM MQ para los sistemas AIX and Linux

Compartir directorios `qmgrs` y `log` predeterminados

Una alternativa a "Compartir todo" en la página 138es compartir los directorios `qmgrs` y `log` por separado (Figura 38 en la página 137). En esta configuración, no es necesario establecer `AMQ_MQS_INI_LOCATION` ya que el valor predeterminado mqs.ini se almacena en el sistema de archivos `/var/mqm` local. Los archivos y directorios, como `mqclient.ini` y `mqserver.ini`, tampoco se comparten.

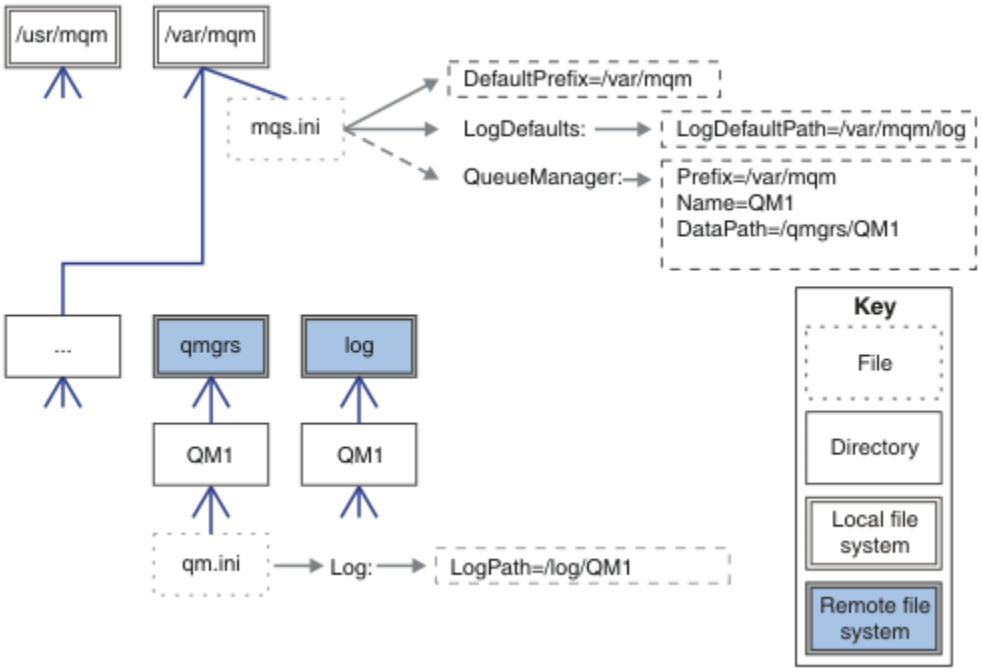


Figura 38. Compartir directorios `qmgrs` y `log`

Compartir directorios con nombre `qmgrs` y `log`

La configuración en Figura 39 en la página 138 coloca `log` y `qmgrs` en un sistema de archivos compartido remoto con nombre común denominado `/ha`. La misma configuración física puede crearse de dos formas diferentes.

1. Estableciendo `LogDefaultPath=/ha` y ejecutando el mandato **`crtmqm`** `-md` *`/ha/qmgrs`* `QM1`. El resultado es exacto al que se muestra en Figura 39 en la página 138.
2. Dejando sin modificar las vías de acceso predeterminadas y ejecutando el mandato **`crtmqm`** `-ld` *`/ha/log`* `-md` *`/ha/qmgrs`* `QM1`.

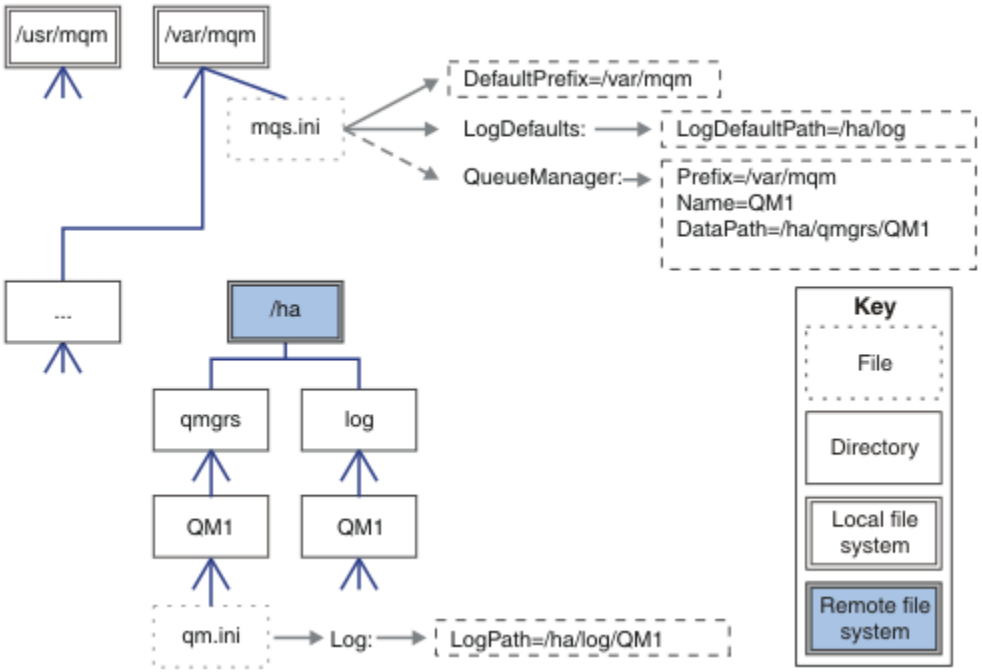


Figura 39. Compartir directorios con nombre `qmgrs` y `log`

Compartir todo

La Figura 40 en la página 139 es un ejemplo de configuración simple para un sistema con almacenamiento de archivos interconectado rápido.

Monte `/var/mqm` como un sistema de archivos compartido remoto. De forma predeterminada, cuando inicia QM1, busca `/var/mqm`, lo encuentra en el sistema de archivos compartido y lee el archivo `mqs.ini` en `/var/mqm`. En vez de utilizar un archivo único `/var/mqm/mqs.ini` para los gestores de colas en todos los servidores, puede establecer la variable de entorno `AMQ_MQS_INI_LOCATION` para que cada servidor apunte a archivos `mqs.ini` diferentes.

Nota: El contenido del archivo de error genérico en `/var/mqm/errors/` se comparte entre gestores de colas en distintos servidores.

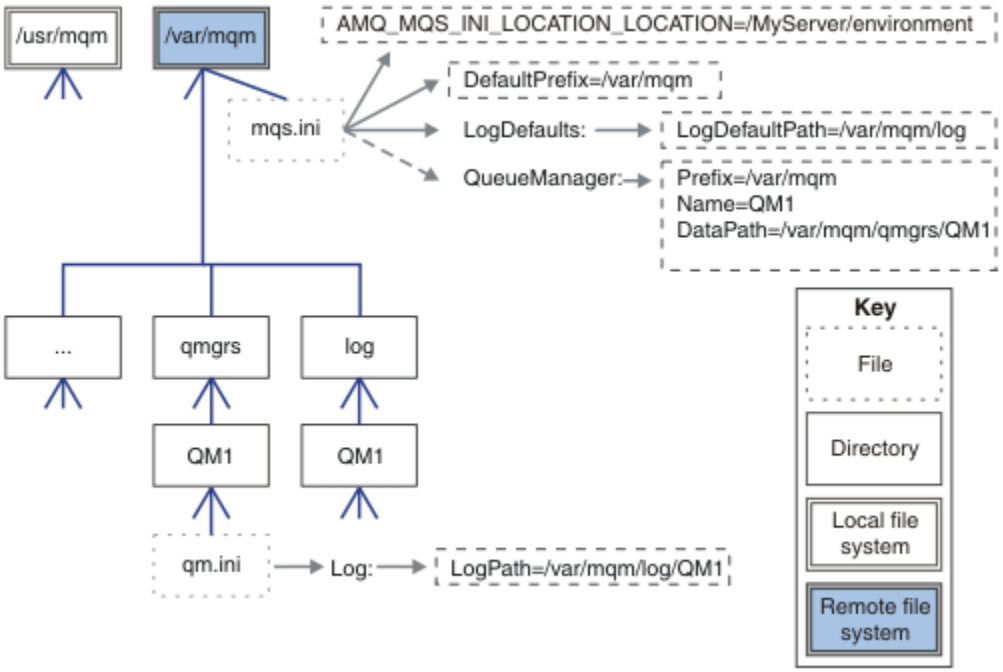


Figura 40. Compartir todo

Tenga en cuenta que no puede utilizarlo para los gestores de colas multiinstancia. La razón es que es necesario que cada host de un gestor de colas de varias instancias tenga su propia copia local de `/var/mqm` para realizar un seguimiento de los datos locales, como los semáforos y la memoria compartida. Estas entidades no pueden compartirse entre distintos hosts.

Windows Estructura de directorios en sistemas Windows

Cómo encontrar información de configuración de gestores de colas y directorios en Windows.

Los directorios predeterminados de la instalación de IBM MQ for Windows son:

Directorio de programas
```
C:\Archivos de programa\IBM\MQ
```
Directorio de datos
```
C:\ProgramData \IBM \MQ
```

Importante: Windows Para las instalaciones de Windows, los directorios son los que se han indicado, a menos que exista una instalación anterior del producto que aún contenga entradas de registro o gestores de colas, o ambos. En esta situación, la instalación nueva utiliza la antigua ubicación del directorio de datos. Para obtener más información, consulte Ubicaciones del directorio de datos y de programas.

Si desea saber qué directorio de instalación y qué directorio de datos se están utilizando, ejecute el mandato dspmqver .

El directorio de instalación se lista en el campo **InstPath** y el directorio de datos se listan en el campo **DataPath**.

Cuando se ejecuta el mandato **dspmqver** se muestra, por ejemplo, la información siguiente:

```
>dspmqver
Name:        IBM MQ
Version:     9.0.0.0
Level:       p900-L160512.4
BuildType:   IKAP - (Production)
Platform:    IBM MQ for Windows (x64 platform)
Mode:        64-bit
O/S:         Windows 7 Professional x64 Edition, Build 7601: SP1
```

```
InstName:    Installation1
InstDesc:
Primary:     Yes
InstPath:    C:\Program Files\IBM\MQ
DataPath:    C:\ProgramData\IBM\MQ
MaxCmdLevel: 900
LicenseType: Production
```

Gestores de colas multiinstancia

Para configurar un gestor de colas multiinstancia, los directorios de datos y registros deben colocarse en un almacén en red, preferiblemente en un servidor diferente a cualquiera de los servidores que estén ejecutando instancias del gestor de colas.

Se proporcionan dos parámetros en el mandato **crtmqm**, **-md** y **-ld**, para facilitar la especificación de la ubicación de los datos del gestor de colas y los directorios de registros. Al especificar el parámetro **-md** se cuadriplica el efecto:

1. La `mqs.ini` stanza `QueueManager\`*QmgrName* contiene una nueva variable, *DataPath*, que apunta al directorio de datos del gestor de colas. A diferencia de la variable *Prefix*, la de vía de acceso incluye el nombre del directorio de gestor de colas.
2. La información de configuración del gestor de colas almacenada en el archivo mqs.ini se reduce a *Name*, *Prefix*, *Directory* y *DataPath*.

Windows *Contenido de directorio*

Lista la ubicación y el contenido de los directorios de IBM MQ.

Una configuración de IBM MQ tiene tres conjuntos principales de archivos y directorios:

1. Archivos ejecutables y otros de sólo lectura que solamente se actualizan cuando se aplica el mantenimiento. Por ejemplo:
   - El archivo readme
   - Los archivos del plug-in y de ayuda de IBM MQ Explorer
   - Archivos de licencias

   Estos archivos se describen en Tabla 16 en la página 140.
2. Archivos y directorios potencialmente modificables específicos de un gestor de colas particular. Estos archivos y directorios se describen en Tabla 17 en la página 141.
3. Archivos y directorios específicos de cada gestor de colas del servidor. Estos archivos y directorios se describen en Tabla 18 en la página 142.

Directorios y archivos de recursos

Los directorios y archivos de recursos contienen todo el código ejecutable y recursos para ejecutar un gestor de colas. La variable, *FilePath*, en la clave de registro de configuración de IBM MQ específica de la instalación, contiene la vía de acceso a los directorios de recursos.

Tabla 16. Directorios y archivos en el directorio `FilePath`

Vía de acceso del archivo	**Contenido**
FilePath`\bin`	Mandatos y DLL
FilePath`\bin64`	Mandatos y DLL (64 bits)
FilePath`\conv`	Tablas de conversión de datos
FilePath`\doc`	Archivos de ayuda del asistente
FilePath`\MQExplorer`	Explorer y plug-ins de Eclipse de ayuda del Explorer
FilePath`\gskit8`	Kit de seguridad global

Tabla 16. Directorios y archivos en el directorio `FilePath` (continuación)	
Vía de acceso del archivo	**Contenido**
FilePath\java	Recursos de Java , incluido JRE
FilePath\licenses	Información de licencia
FilePath\Non_IBM_License	Información de licencia
FilePath\properties	Utilizado de forma interna
FilePath\Tivoli	
FilePath\tools	Ejemplos y recursos de despliegue
FilePath\web	Se describe en la estructura de archivos de los componentes de instalación de IBM MQ Console y REST API para los archivos no editables.
FilePath\Uninst	Utilizado de forma interna
FilePath\README.TXT	Archivo Readme

Directorios no específicos del gestor de colas

Algunos directorios contienen archivos, como archivos de rastreo o registro de errores, que no son específicos de un gestor de colas. La variable *DefaultPrefix* contiene la vía de acceso a esos directorios. *DefaultPrefix* forma parte de la stanza `AllQueueManagers`.

Tabla 17. Directorios y archivos en el directorio `DefaultPrefix`	
Vía de acceso del archivo	**Contenido**
DefaultPrefix\config	Utilizado de forma interna
DefaultPrefix\conv	Archivo de control de conversión de `ccsid_part2.tbl` y `ccsid.tbl data`, descrito en Conversión de datos
DefaultPrefix\errors	Registros de errores del gestor de colas no, `AMQERR` *nn*`.LOG`
DefaultPrefix\exits	Programas de salida de canal
DefaultPrefix\exits64	Programas de salida de canal (64 bits)
DefaultPrefix\ipc	No utilizado
DefaultPrefix\qmgrs	Se describe en Tabla 18 en la página 142
DefaultPrefix\trace	Archivos de rastreo
DefaultPrefix\web	Se describe en la estructura de archivos de los componentes de instalación de IBM MQ Console y REST API para los archivos editables de usuario
DefaultPrefix\amqmjpse.txt	Utilizado de forma interna

Directorios del gestor de colas

Cuando se crea un gestor de colas, se crea un conjunto nuevo de directorio específico del gestor de colas.

Si crea un gestor de colas con el parámetro **-md** *filepath* , la vía de acceso se almacena en la variable *DataPath* en la stanza del gestor de colas del archivo mqs.ini . Si crea un gestor de colas sin establecer el parámetro **-md** *filepath* , los directorios del gestor de colas se crean en la vía de acceso almacenada en *DefaultPrefix*, y la vía de acceso se copia en la variable *Prefix* en la stanza del gestor de colas del archivo mqs.ini .

Tabla 18. Directorios y archivos en directorios DataPath y Prefix\qmgrs\QmgrName	
Vía de acceso del archivo	**Contenido**
DataPath\@ipcc	Ubicación predeterminada para AMQCLCHL.TAB, la tabla de conexión de cliente.
DataPath\authinfo	Utilizado internamente.
DataPath\channel	
DataPath\clntconn	
DataPath\errors	Registros de errores, AMQERR *nn*.LOG
DataPath\listener	Utilizado internamente.
DataPath\namelist	
DataPath\plugcomp	
DataPath\procdef	
DataPath\qmanager	
DataPath\queues	
DataPath\services	
DataPath\ssl	
DataPath\startprm	
DataPath\topic	
DataPath\active	
DataPath\active.dat	
DataPath\amqalchk.fil	
DataPath\master	
DataPath\master.dat	
DataPath\qm.ini	Configuración del gestor de colas
DataPath\qmstatus.ini	Estado del gestor de colas
DataPath\userdata	Se puede utilizar para almacenar el estado persistente de las aplicaciones.
Prefix\qmgrs*QmgrName*	Utilizado de forma interna
Prefix\qmgrs\@SYSTEM	No utilizado
Prefix\qmgrs\@SYSTEM\errors	
*DataPath**autocfg*	Se utiliza para la configuración automática

IBM i Estructura de directorios en IBM i

Se ofrece una descripción de IFS y se indica la estructura de directorios IFS de IBM MQ para servidor, cliente y Java.

El sistema de archivos integrado (IFS) forma parte de IBM i que da soporte a una corriente de entrada/salida y una gestión de almacenamiento parecidas a la de un sistema personal, sistemas operativos AIX and Linux, a la vez que proporciona una estructura integral de toda la información almacenada en el servidor.

En IBM i, los nombres de directorio empiezan por el carácter & (ampersand) en lugar del carácter @ (at). Por ejemplo, @system en IBM i es &system.

Sistema de archivos raíz IFS para el servidor de IBM MQ

Al instalar el servidor de IBM MQ para IBM i, se crean los siguientes directorios en el sistema de archivos raíz IFS.

ProdData:

Visión general

```
QIBM
   '-- ProdData
      '-- mqm
      '-- doc
      '-- inc
      '-- lib
      '-- samp
      '-- licenses
      '-- LicenseDoc
      '-- 5724H72_V8R0M0
```

/QIBM/ProdData/mqm
Los subdirectorios de este directorio contienen todos los datos del producto, por ejemplo, las clases CC+, los archivos de formato de rastreo y los archivos de licencia. Los datos de este directorio se suprimen y reemplazan cada vez que se instala el producto.

/QIBM/ProdData/mqm/doc
En este subdirectorio se instala una Consulta de mandatos para los mandatos CL que se proporciona en formato HTML.

/QIBM/ProdData/mqm/inc
Los archivos de cabecera para compilar los programas C o C++.

/QIBM/ProdData/mqm/lib
Archivos auxiliares utilizados por MQ.

/QIBM/ProdData/mqm/samp
Más ejemplos.

/QIBM/ProdData/mqm/licenses
Archivos de licencia. Los dos archivos para cada idioma se denominan LA_ *xx* y LI_ *xx* donde *xx* es el identificador de idioma de 2 caracteres para cada idioma suministrado.

También el directorio siguiente almacena archivos de acuerdos de licencia:

/QIBM/ProdData/LicenseDoc/5724H72_V8R0M0
Archivos de licencia. Los archivos se denominan 5724H72_V8R0M0_ *xx* donde *xx* es el identificador de idioma de 2 o 5 caracteres para cada idioma suministrado.

UserData:

Visión general

```
QIBM
   '-- UserData
      '-- mqm
      '-- errors
      '-- trace
      '-- qmgrs
      '-- &system
      '-- qmgrname1
```

```
      '-- qmgrname2
      '-- and so on
```

/QIBM/UserData/mqm
Los subdirectorios de este directorio contienen todos los datos de usuario relacionados con los gestores de colas.

Cuando se instala el producto, se crea un archivo mqs.ini en el directorio /QIBM/UserData/mqm/ (a menos que ya exista en él como resultado de una instalación anterior).

Cuando se crea un gestor de colas, se crea un archivo qm.ini en el directorio /QIBM/UserData/mqm/qmgrs/ *NOMBREGC*/ (donde *NOMBREGC* es el nombre del gestor de colas).

Los datos de los directorios se conservan cuando se suprime el producto.

Sistema de archivos raíz IFS para IBM MQ MQI client

Al instalar el servidor de IBM MQ MQI client for IBM i, se crean los siguientes directorios en el sistema de archivos raíz IFS:

ProdData:

Visión general

```
QIBM
   '-- ProdData
      '-- mqm
      '-- lib
```

/QIBM/ProdData/mqm
Los subdirectorios de este directorio contienen todos los datos del producto. Los datos de este directorio se suprimen y reemplazan cada vez que se sustituye el producto.

UserData:

Visión general

```
QIBM
   '-- UserData
      '-- mqm
      '-- errors
      '-- trace
```

/QIBM/UserData/mqm
Los subdirectorios de este directorio contienen todos los datos de usuario.

Sistema de archivos raíz IFS para IBM MQ Java

Cuando instala IBM MQ Java en IBM i, los directorios siguientes se crean en el sistema de archivos raíz IFS:

ProdData:

Visión general

```
QIBM
   '-- ProdData
      '-- mqm
      '-- java
      '--samples
      '-- bin
      '-- lib
```

/QIBM/ProdData/mqm/java
Los subdirectorios de este directorio contienen todos los datos del producto, incluidas las clases Java. Los datos de este directorio se suprimen y reemplazan cada vez que se sustituye el producto.

/QIBM/ProdData/mqm/java/samples
Los subdirectorios de este directorio contienen las clases y los datos de ejemplo de Java.

Bibliotecas creadas por las instalaciones de cliente y servidor

La instalación del cliente o servidor de IBM MQ crea las siguientes bibliotecas:

- QMQM

  Biblioteca del producto.
- QMQMSAMP

  Biblioteca de ejemplos (si opta por instalar los ejemplos).
- QMxxxx

  Sólo servidor.

  Cada vez que crea un gestor de colas, IBM MQ crea automáticamente una biblioteca asociada, con un nombre como QMxxxx donde xxxx se deriva del nombre del gestor de colas. Esta biblioteca contiene objetos específicos del gestor de colas, incluyendo diarios y sus receptores asociados. De forma predeterminada, el nombre de esta biblioteca se deriva del nombre del gestor de colas, al que se le añaden a modo de prefijo los caracteres QM. Por ejemplo, si el gestor de colas se llama TEST, la biblioteca se llamará QMTEST.

  Nota: Cuando crea un gestor de colas, puede especificar el nombre de su biblioteca si lo desea. Por ejemplo:

  ```
  CRTMQM MQMNAME(TEST) MQMLIB(TESTLIB)
  ```

  Puede utilizar el mandato WRKLIB para enumerar todas las bibliotecas creadas por IBM MQ para IBM i. En las bibliotecas del gestor de colas, verá el texto QMGR: QMGRNAME. El formato del mandato es:

  ```
  WRKLIB LIB(QM*)
  ```

  Estas bibliotecas asociadas a gestores de colas se conservan cuando se suprime el producto.

Multi Planificación del soporte del sistema de archivos para MFT en Multiplatforms

Los agentes de IBM MQ Managed File Transfer MFT se pueden utilizar para transferir datos a y desde archivos en un sistema de archivos. Además, los supervisores de recursos que se ejecutan en un agente se pueden configurar para supervisar archivos en un sistema de archivos.

MFT tiene el requisito de que estos archivos se almacenen en un sistema de archivos que soporte el bloqueo. Existen dos razones para ello:

- Un agente bloquea un archivo para asegurarse de que no cambia una vez que ha empezado a leer datos de él, o a grabar datos en él.
- Los supervisores de recursos bloquean los archivos para comprobar que ningún otro proceso los está utilizando actualmente.

Los agentes y supervisores de recursos utilizan el Java método **`FileChannel.tryLock()`** para realizar el bloqueo, y el sistema de archivos debe poder bloquear los archivos cuando se le solicite que lo haga utilizando esta llamada.

Importante: Los siguientes sistemas de archivos no están soportados, ya que no cumplen los requisitos técnicos de MFT:

- GlusterFS
- NFS versión 3

Multi Elección de registro circular o lineal en Multiplatforms

En IBM MQ, puede elegir el registro circular o lineal. La información siguiente le ofrece una visión general de ambos tipos.

Ventajas de registro circular

Las principales ventajas del registro circular son que el registro circular es:

- Más fácil de administrar.

  Una vez que ha configurado el registro circular correctamente para la carga útil, no es necesaria más administración. Mientras que, para el registro lineal, las imágenes de soporte necesitan registrarse y las extensiones de registro que ya no son necesarias deben archivarse o suprimirse.
- Mejor rendimiento

  El registro circular funciona mejor que el registro lineal, porque el registro circular es capaz de volver a utilizar extensiones de registro que ya se han formateado. Considerando que el registro lineal tiene que asignar nuevas extensiones de registro y formatearlas.

Consulte Gestión de registros para obtener más información.

Ventajas del registro lineal

La principal ventaja del registro lineal es que el registro lineal proporciona protección contra más anomalías.

Ni el registro circular ni el registro lineal protegen contra un registro dañado o suprimido o los mensajes o las colas suprimidos por aplicaciones o el administrador.

El registro lineal (pero no circular) permite recuperar objetos dañados. Por lo tanto, el registro lineal proporciona protección contra los archivos de cola dañados o suprimidos, porque estas colas dañadas pueden recuperarse de un registro lineal.

Tanto el registro circular como el lineal protegen frente la pérdida de alimentación y la anomalía de comunicaciones como se describe en Recuperación de pérdida de alimentación o anomalías de comunicación.

Otras consideraciones

El hecho de que elija lineal o circular depende del grado de redundancia que necesite.

Hay un coste al elegir más redundancia, que es el registro lineal, producido por el costo de rendimiento y el coste de administración.

Consulte Tipos de registro para obtener más información.

AIX Memoria compartida en AIX

Si ciertos tipos de aplicaciones no se pueden conectar debido a una limitación de memoria en AIX, en muchos casos este problema se puede resolver estableciendo la variable de entorno EXTSHM=ON.

Algunos procesos de 32 bits en AIX pueden encontrarse con una limitación del sistema operativo que afecta a su capacidad para conectarse a gestores de colas de IBM MQ. Cada conexión estándar a IBM MQ utiliza memoria compartida pero, a diferencia de otras plataformas UNIX, AIX permite a los procesos de 32 bits adjuntar sólo 11 segmentos de memoria compartida.

La mayoría de los procesos de 32 bits no encontrarán este límite, pero es posible que las aplicaciones con altos requisitos de memoria no se puedan conectar a IBM MQ con el código de razón 2102: MQRC_RESOURCE_PROBLEM. Este error lo pueden ver los siguientes tipos de aplicaciones:

- Programas que se ejecutan en máquinas virtuales Java de 32 bits
- Programas que utilizan los modelos de memoria grandes o muy grandes
- Programas que se conectan a muchos gestores de colas o bases de datos
- Programas que se conectan a conjuntos de memoria de compartida por sí mismos

AIX ofrece una característica de memoria compartida ampliada para los procesos de 32 bits que les permite adjuntar más memoria compartida. Para ejecutar una aplicación con esta característica, exporte la variable de entorno EXTSHM=ON antes de iniciar los gestores de colas y el programa. La característica EXTSHM=ON evita este error en la mayoría de los casos, pero es incompatible con programas que utilizan la opción SHM_SIZE de la función shmctl.

Las aplicaciones de IBM MQ MQI client y todos los procesos de 64 bits no se ven afectados por esta limitación. Pueden conectarse a gestores de colas de IBM MQ, independientemente de si la variable EXTSHM se ha establecido o no.

Linux AIX IBM MQ y los recursos IPC de UNIX System V

Un gestor de colas utiliza recursos IPC. Utilice **ipcs -a** para averiguar qué recursos se están utilizando.

Esta información solo se aplica a IBM MQ en ejecución en sistemas AIX and Linux.

IBM MQ utiliza recursos de comunicación entre procesos (IPC) de System V (*semáforos* y *segmentos de memoria compartida*) para almacenar y pasar datos entre componentes del sistema. Estos recursos los utilizan las aplicaciones y procesos de gestor de colas que se conectan con el gestor de colas. Los IBM MQ MQI clients no utilizan recursos IPC, excepto el control de rastreo de IBM MQ. Utilice el UNIX mandato **ipcs -a** para obtener información completa sobre el número y el tamaño de los recursos IPC actualmente en uso en la máquina.

Linux AIX IBM MQ y prioridad de procesos en UNIX

Buenas prácticas al establecer valores *nice* de prioridad de procesos.

Esta información solo se aplica a IBM MQ en ejecución en sistemas AIX and Linux.

Si ejecuta un proceso en segundo plano, el shell que invoque ese proceso le puede asignar un valor *nice* más alto (y, por lo tanto, una prioridad inferior). Esto puede tener implicaciones en el rendimiento general de IBM MQ. En situaciones de mucha actividad, si hay muchas hebras listas para ejecutarse con una prioridad superior y algunas con una prioridad inferior, las características de planificación del sistema operativo pueden quitar tiempo de procesador a las hebras con prioridad inferior.

Una buena práctica es que los procesos iniciados independientemente asociados a gestores de colas, como **runmqlsr**, tengan los mismos valores *nice* que el gestor de colas al que están asociados. Asegúrese de que el shell no asigne un valor *nice* más alto a estos procesos en segundo plano. Por ejemplo, en ksh, utilice el valor "`set +o bgnice`" para impedir que ksh aumente el valor *nice* de los procesos en segundo plano. Puede verificar los valores *nice* de los procesos en ejecución examinando la columna *NI* de una lista "ps -efl".

Además, inicie los procesos de aplicaciones IBM MQ con el mismo valor *nice* que el gestor de colas. Si se ejecutan con distintos valores *nice*, una hebra de la aplicación podría bloquear una hebra del gestor de colas o viceversa, lo que empeoraría el rendimiento.

z/OS Planning your IBM MQ environment on z/OS

When planning your IBM MQ environment, you must consider the resource requirements for data sets, page sets, Db2, Coupling Facilities, and the need for logging, and backup facilities. Use this topic to plan the environment where IBM MQ runs.

Before you plan your IBM MQ architecture, familiarize yourself with the basic IBM MQ for z/OS concepts, see the topics in IBM MQ for z/OS concepts.

When planning your queue manager, you might need to work with different people in your organization. It is usually a good idea to involve those people early, as change control procedures can take a long time. They might also be able to tell you what parameters you need to configure IBM MQ for z/OS.

For example you might need to work with the:

- Storage administrator, to determine the high level qualifier of queue manager data sets, and to allocate enough space for queue manager data sets.
- z/OS system programmer to define the IBM MQ subsystem to z/OS and APF authorize the IBM MQ for z/OS libraries.
- Network administrator to determine which TCP/IP stack and ports should be used for IBM MQ for z/OS.
- Security administrator to set up access to queue manager data sets, security profiles for IBM MQ for z/OS resources, and TLS certificates.
- Db2 administrator to set up Db2 tables when configuring a queue sharing group.

Related concepts

IBM MQ Technical overview

Related tasks

“Planificación de una arquitectura de IBM MQ” on page 5

Cuando planifique su entorno de IBM MQ, tenga en cuenta el soporte que proporciona IBM MQ para las arquitecturas de uno o varios gestores de colas y para los estilos de mensajería de punto a punto y de publicación/suscripción. Además planifique los requisitos de recursos y su uso de los recursos de registro y copia de seguridad.

Configuring z/OS

Administering IBM MQ for z/OS

z/OS Planning for your queue manager

When you are setting up a queue manager, your planning should allow for the queue manager to grow, so that the queue manager meets the needs of your enterprise.

The best way to configure a queue manager is in steps:

1. Configure the base queue manager
2. Configure the channel initiator which does queue manager to queue manager communications, and remote client application communication
3. If you want to encrypt and protect messages, configure Advanced Message Security
4. If you want to use File Transfer over IBM MQ, configure Managed File Transfer for z/OS.
5. If you want to use the administrative or messaging REST API, or the IBM MQ Console to manage IBM MQ from a web browser, configure the mqweb server.

Some enterprises have hundreds of thousands of queue managers in their environment. You need to consider your IBM MQ network now, and in five years time.

On z/OS, some queue managers process thousands of messages a second, and log over 100 MB a second. If you expect very high volumes you may need to consider having more than one queue manager.

On z/OS, IBM MQ can run as part of a queue sharing group (QSG) where messages are stored in the Coupling Facility, and any queue manager in the queue sharing group can access the messages. If you want to run in a queue sharing group you need to consider how many queue managers you need. Typically, there is one queue manager for each LPAR. You might also have one queue manager to backup CF structures regularly.

Some changes to configuration are easy to do, such as defining a new queue. Some are harder, such as making logs and page sets bigger; and some configuration cannot be changed, such as the name of a queue manager or the queue sharing group name.

There is performance and tuning information available in the MP16 performance SupportPac .

Naming conventions

You need to have a naming convention for the queue manager data sets.

Many enterprises use the release number in the name of the load libraries, and so on. You might want to consider having an alias of `MQM.SCSQAUTH` pointing to the version currently in use, such as `MQM.V930.SCSQAUTH`, so you do not have to change CICS®, Batch, and IMS JCL when you migrate to a new version of IBM MQ.

You can use a symbolic link in z/OS UNIX System Services to reference the installation directory for the version of IBM MQ currently in use.

The data sets used by the queue manager (logs, page sets, JCL libraries) need a naming convention to simplify the creation of security profiles, and the mapping of data sets to SMS storage classes that control where the data sets are placed on disk, and the attributes they have.

Note, that putting the version of IBM MQ into the name of the page sets or logs, is not a good idea. One day you might migrate to a new version, and the data set will have the "wrong" names.

Applications

You need to understand the business applications and the best way to configure IBM MQ. For example if applications have logic to provide recovery and repeat capability, then non persistent messages might be good enough. If you want IBM MQ to handle the recovery, then you need to use persistent messages and put and get messages in syncpoint.

You need to isolate queues from different business transactions. If a queue for one business application fills up, you do not want this impacting other business applications. Isolate the queues in different page sets and buffer pools, or structures, if possible.

You need to understand the profile of messages. For many applications the queues have only a few messages. Other applications can have queues build up during the day, and be processed overnight. A queue which normally has only a few messages on it, might need to hold many hours worth of messages if there is a problem and messages are not processed. You need to size the CF structures and page sets to allow for your expected peak capacity.

Post configuration

Once you have configured your queue manager (and components) you need to plan for:

- Backing up page sets.
- Backing up definitions of objects.
- Automating the backup of any CF structures.
- Monitoring IBM MQ messages, and taking action when a problem is detected.
- Collecting the IBM MQ statistics data.
- Monitoring resource usage, such as virtual storage, and amount of data logged per hour. With this you can see if your resource usage is increasing and if you need to take actions, such as setting up a new queue manager

z/OS Planning your storage and performance requirements on z/OS

You must set realistic and achievable storage, and performance goals for your IBM MQ system. Use this topic help you understand the factors which affect storage, and performance.

This topic contains information about the storage and performance requirements for IBM MQ for z/OS. It contains the following sections:

- z/OS performance options for IBM MQ
- Determining z/OS workload management importance and velocity goals
- “Library storage” on page 150

- "System LX usage" on page 150
- "Storage configuration" on page 151
- "Disk storage" on page 156

See, "Where to find more information about storage and performance requirements" on page 156 for more information.

z/OS performance options for IBM MQ

With workload management, you define performance goals and assign a business importance to each goal. You define the goals for work in business terms, and the system decides how much resource, such as processor and storage, should be given to the work to meet its goal. Workload management controls the dispatching priority based on the goals you supply. Workload management raises or lowers the priority as needed to meet the specified goal. Thus, you need not fine-tune the exact priorities of every piece of work in the system and can focus instead on business objectives.

The three kinds of goals are:

Response time
How quickly you want the work to be processed

Execution velocity
How fast the work should be run when ready, without being delayed for processor, storage, I/O access, and queue delay

Discretionary
A category for low priority work for which there are no performance goals

Response time goals are appropriate for end-user applications. For example, CICS users might set workload goals as response time goals. For IBM MQ address spaces, velocity goals are more appropriate. A small amount of the work done in the queue manager is counted toward this velocity goal but this work is critical for performance. Most of the work done by the queue manager counts toward the performance goal of the end-user application. Most of the work done by the channel initiator address space counts toward its own velocity goal. The receiving and sending of IBM MQ messages, which the channel initiator accomplishes, is typically important for the performance of business applications using them.

Determining z/OS workload management importance and velocity goals

See "Determining z/OS workload management importance" on page 151 for more information.

Library storage

You must allocate disk storage for the product libraries. The exact figures depend on your configuration, and should include both the target and distribution libraries, as well as the SMP/E libraries.

The target libraries used by IBM MQ for z/OS use PDSE formats. Ensure that any PDSE target libraries are not shared outside a sysplex. For more information about the required libraries and their sizes and the required format, see the Program Directory. Para enlaces de descarga de los directorios de programas, consulteIBM MQ for z/OS Archivos PDF del directorio de programas .

System LX usage

Each defined IBM MQ subsystem reserves one system linkage index (LX) at IPL time, and a number of non-system linkage indexes when the queue manager is started. The system linkage index is reused when the queue manager is stopped and restarted. Similarly, distributed queuing reserves one non-system linkage index. In the unlikely event of your z/OS system having inadequate system LXs defined, you might need to take these reserved system LXs into account.

If required, the number of system LXs can be increased by setting the *NSYSLX* parameter in SYS1.PARMLIB member IEASYSxx.

z/OS *Determining z/OS workload management importance*

For full information about workload management and defining goals through the service definition, see the .z/OS product documentation.

This topic suggests how to set the z/OS workload management importance and velocity goals relative to other important work in your system. See *z/OS MVS Planning: Workload Management* for more information.

The queue manager address space needs to be defined with high priority as it provides subsystem services. The channel initiator is an application address space, but is usually given a high priority to ensure that messages being sent to a remote queue manager are not delayed. Advanced Message Security (AMS) also provides subsystem services and needs to be defined with high priority.

Use the following service classes:

The default SYSSTC service class

- VTAM and TCP/IP address spaces
- IRLM address space (IRLMPROC)

Note: The VTAM, TCP/IP, and IRLM address spaces must have a higher dispatching priority than all the DBMS address spaces, their attached address spaces, and their subordinate address spaces. Do not allow workload management to reduce the priority of VTAM, TCP/IP, or IRLM to (or below) that of the other DBMS address spaces

A high velocity goal and importance of 1 for a service class with a name that you define, such as PRODREGN, for the following:

- IBM MQ queue manager, channel initiator and AMS address spaces
- Db2 (all address spaces, except for the Db2-established stored procedures address space)
- CICS (all region types)
- IMS (all region types except BMPs)

A high velocity goal is good for ensuring that startups and restarts are performed as quickly as possible for all these address spaces.

The velocity goals for CICS and IMS regions are only important during startup or restart. After transactions begin running, workload management ignores the CICS or IMS velocity goals and assigns priorities based on the response time goals of the transactions that are running in the regions. These transaction goals should reflect the relative priority of the business applications they implement. They might typically have an importance value of 2. Any batch applications using IBM MQ should similarly have velocity goals and importance reflecting the relative priority of the business applications they implement. Typically the importance and velocity goals will be less than those for PRODREGN.

z/OS *Storage configuration*

V 9.4.0 In a 64 bit address space, there is a virtual line called "the bar" that marks the 2GB address. The bar separates storage below the 2GB address, called "below the bar", from storage above the 2GB address, called "above the bar". Storage below the bar uses 31 bit addressability, storage above the bar uses 64 bit addressability.

V 9.4.0

You can specify the limit of 31-bit storage by using the JCL REGION parameter, and the limit of 64-bit storage by using the MEMLIMIT parameter. These specified values can be overridden by z/OS exits.

Suggested storage configuration

The following table shows suggested **REGION** and **MEMLIMIT** values for the queue manager, channel initiator, and AMS address spaces. These suggestions should be used as a starting point and adjusted using the information in:

- “Queue manager storage configuration” on page 152
- “Channel initiator storage configuration from IBM MQ 9.4.0” on page 154

Table 19. Suggested definitions for REGION and MEMLIMIT

Address space	**Storage configuration**
Queue manager	REGION=0M, MEMLIMIT=3G
V 9.4.0 Channel initiator from IBM MQ 9.4.0	REGION=0M, MEMLIMIT=2G
AMS address space	REGION=0M

Managing the MEMLIMIT and REGION size

Other mechanisms, for example the **MEMLIMIT** parameter in the SMFPRMxx member of SYS1.PARMLIB or the IEFUSI exit might be used at your installation to provide a default amount of virtual storage above the bar for z/OS address spaces. See Memory management above the bar for full details about limiting storage above the bar.

z/OS *Queue manager storage configuration*

The queue manager address space is likely to be the major user of 64-bit storage in an IBM MQ installation. Each connection to the queue manager requires common storage to be allocated as described in the following text. In addition to 64-bit storage, you should allow the queue manager to use all available 31-bit storage by specifying REGION=0M on the queue manager JCL.

Common storage

Each IBM MQ for z/OS subsystem has the following approximate storage requirements:

- CSA 4KB
- ECSA 800KB, plus the size of the trace table that is specified in the **TRACTBL** parameter of the CSQ6SYSP system parameter macro. For more information, see Using CSQ6SYSP.

In addition, each concurrent logical connection to the queue manager requires about 5 KB of ECSA. When a task ends, other IBM MQ tasks can reuse this storage.

IBM MQ does not release the storage until the queue manager is shut down, so you can calculate the maximum amount of ECSA required by multiplying the maximum number of concurrent connections by 5KB. The number of concurrent logical connections is the sum of the number of:

- Tasks (TCBs) in Batch, TSO, z/OS UNIX System Services, IMS, and Db2 stored procedure address space (SPAS) regions that are connected to IBM MQ, but not disconnected.
- CICS transactions that have issued an IBM MQ request, but have not terminated
- JMS Connections, Sessions, TopicSessions or QueueSessions that have been created (for bindings connection), but not yet destroyed or garbage collected.
- Active IBM MQ channels

You can set a limit to the common storage, used by logical connections to the queue manager, with the **ACELIM** configuration parameter. The **ACELIM** control is primarily of interest to sites where Db2 stored procedures cause operations on IBM MQ queues.

When driven from a stored procedure, each IBM MQ operation can result in a new logical connection to the queue manager. Large Db2 units of work, for example due to table load, can result in an excessive demand for common storage.

ACELIM is intended to limit common storage use and to protect the z/OS system, by limiting the number of connections in the system. You should only set **ACELIM** on queue managers that have been identified

as using excessive quantities of ECSA storage. See the **ACELIM** section in *Using CSQ6SYSP* for more information.

To set a value for **ACELIM**, firstly determine the amount of storage currently in the subpool controlled by the **ACELIM** value. This information is in the SMF 115 subtype 5 records produced by statistics CLASS(3) trace.

IBM MQ SMF data can be formatted using SupportPac MP1B. The number of bytes in use in the subpool controlled by **ACELIM** is displayed in the STGPOOL DD, on the line titled *ACE/PEB*.

For more information about SMF 115 statistics records, see Interpreting IBM MQ for z/OS performance statistics.

Increase the normal value by a sufficient margin to provide space for growth and workload spikes. Divide the new value by 1024 to yield a maximum storage size in KB for use in the **ACELIM** configuration.

Private storage

The queue manager address space uses 64-bit storage for many internal control blocks. The **MEMLIMIT** parameter of the queue manager JCL defines the maximum amount of 64-bit storage available. 3GB of storage, `MEMLIMIT=3G`, is the minimum you should use, however, depending on your configuration significantly more might be required.

You should specify a specific **MEMLIMIT** value rather than `MEMLIMIT=NOLIMIT` to prevent potential problems. If you specify `NOLIMIT` or a very large value, then there is the potential to use up all of the available z/OS virtual storage, which leads to paging in your system. When increasing the value of **MEMLIMIT** you should discuss the new setting with your z/OS system programmer in case there is a system-wide limit on the amount of on storage that can be used.

If you have a large value for **MEMLIMIT** you might need to increase the size of your dump data sets as more data is captured in a dump.

You can monitor the address space storage usage from the CSQY220I message that indicates the amount of 31 and 64-bit private storage in use, and the remaining free amount.

Buffer pools

Buffer pools are a significant user of private storage in the queue manager address space. Each buffer pool size is determined at queue manager initialization time, and storage is allocated for the buffer pool when a page set that is using that buffer pool is connected. The parameter **LOCATION (ABOVE|BELOW)** is used to specify where the buffers are allocated. You can use the ALTER BUFFPOOL command to dynamically change the size of buffer pools.

When calculating a value for **MEMLIMIT** it is critical that you take into account the buffer pool sizes if they are configured with **LOCATION(ABOVE)**. You should perform the calculation as follows.

Calculate the value of **MEMLIMIT** as 2GB plus the size of the buffer pools configured with **LOCATION(ABOVE)**, rounded up to the nearest GB. Set MEMLIMIT to a minimum of 3GB and increase this as necessary when you need to increase the size of your buffer pools.

For example, for three buffer pools configured with **LOCATION(ABOVE)**, buffer pool one has 10,000 buffers, and buffer pools two and three have 50,000 buffers each. Memory usage above the bar equals 110,000 (total number of buffers) * 4096 = 450,560,000 bytes = 430MB.

All buffer pools regardless of **LOCATION** make use of 64-bit storage for control structures. As the number of buffer pools and number of buffers in those pools increase this can become significant. Each buffer requires around an additional 200 bytes of 64-bit storage. For the preceding configuration that would require: 200 * 110,000 = 22,000,000 bytes = 21MB.

Therefore, in this scenario 3GB can be used for the **MEMLIMIT**, which allows scope for growth: 21MB + 430MB + 2GB which rounds up to 3GB.

For some configurations there can be significant performance benefits to using buffer pools that have their buffers permanently backed by real storage. You can achieve this by specifying the FIXED4KB value

for the **PAGECLAS** attribute of the buffer pool. However, you should only do this if there is sufficient real storage available on the LPAR, otherwise other address spaces might be affected. For information about when you should use the FIXED4KB value for **PAGECLAS**, see IBM MQ Support Pac MP16: IBM MQ for z/OS - Capacity planning & tuning.

Making the buffer pools so large that there is MVS™ paging might adversely affect performance. You might consider using a smaller buffer pool that does not page, with IBM MQ moving the message to and from the page set.

Indexed queues

On z/OS, local queues are indexed if the queue has an **INDXTYPE** attribute that has not been set to NONE. The indexes for shared queues are held in a coupling facility, but for private queues the index is held in 64 bit storage. For each message on an indexed queue 136 bytes of data are used to index the message. For very deep queues this can result in a significant amount of 64 bit storage being allocated. For example, 10 million messages on an indexed queue will use 1.27 GB of 64 bit storage in order to maintain the index.

If you expect to have a large number of messages on indexed queues you should allow for this when setting **MEMLIMIT**. To calculate an upper limit for the amount of storage required for indexes, multiply the **MAXDEPTH** attribute for each indexed queue by 136 and sum the value. This value should be added to your existing **MEMLIMIT**.

V 9.4.0 RECOVER CFSTRUCT

From IBM MQ 9.4.0 the **RECOVER CFSTRUCT** command makes greater use of 64-bit storage. In many cases there should be spare 64-bit storage available and so use of the command does not require an increase in the value of **MEMLIMIT**. However, if you are likely to have large structure backups, containing more than a few million messages, you should increase the **MEMLIMIT** for all queue managers which might process the **RECOVER CFSTRUCT** command by 500MB.

For example if you had MEMLIMIT=3G already, you should consider using MEMLIMIT=4G as the **MEMLIMIT** parameter does not allow for decimal points.

Shared Message Data Set (SMDS) buffers and MEMLIMIT

When running messaging workloads using shared message data sets, there are two levels of optimizations that can be achieved by adjusting the DSBUFS and DSBLOCK attributes.

The amount of above bar queue manager storage used by the SMDS buffer is DSBUFS x DSBLOCK. This means that by default, 100 x 256KB (25MB) is used for each CFLEVEL(5) structure in the queue manager.

Although this value is not too high, if your enterprise, or enterprises have many CFSTRUCTs, some of them might allocate a high value of MEMLIMIT for buffer pools, and sometimes they have deep indexed queues, so in total, they might run out of storage above the bar.

V 9.4.0 z/OS *Channel initiator storage configuration from IBM MQ 9.4.0*

The channel initiator typically uses much less 64-bit storage than the queue manager. However, from IBM MQ 9.4.0 the usage has increased. In addition to 64-bit storage, you should allow the channel initiator to use all available 31-bit storage by specifying REGION=0M on the queue manager JCL.

Common storage

The channel initiator typically requires ECSA usage of up to 160KB.

31-bit private storage

The 31-bit storage available to the channel initiator limits the number of concurrent connections the CHINIT can have.

Every channel uses approximately 170KB of extended private region in the channel initiator address space. For message channels, for example, sender or receiver channels, storage is increased by message size if messages larger than 32KB are transmitted. This increased storage is freed when:

- A sending or client channel requires less than half the current buffer size for 10 consecutive messages.
- A heartbeat is sent or received.

The storage is freed for reuse within the Language Environment, however, the storage is not seen as free by the z/OS virtual storage manager. This means that the upper limit for the number of channels is dependent on message size and arrival patterns, and on limitations of individual user systems on extended private region size.

The upper limit on the number of channels is likely to be approximately 9000 on many systems because the extended region size is unlikely to exceed 1.6GB.

The channel initiator trace is written to a data space. The size of the data space storage, is controlled by the **TRAXTBL** parameter. See ALTER QMGR.

64-bit private storage

The MEMLIMIT parameter of the channel initiator JCL defines the maximum amount of 64-bit storage available. 2 GB of storage, MEMLIMIT=2 GB, is the minimum value you should use. Depending on your configuration significantly more might be required.

You should specify a sensible MEMLIMIT value rather than MEMLIMIT=NOLIMIT to prevent potential problems. If you specify NOLIMIT or a very large value, then there is the potential to use up all of the available z/OS virtual storage, leading to paging in your system. When increasing the value of MEMLIMIT you should discuss the new setting with your z/OS system programmer in case there is a system-wide limit on the amount of on storage that can be used.

If you have a large value for MEMLIMIT you might need to increase the size of your dump data sets as more data is captured in a dump.

There are two users of 64-bit storage in the channel initiator: SMF and server-connection channels.

SMF

If enabled, SMF class 4 accounting, or statistics, require 64-bit storage. A minimum of 256MB storage is required. If sufficient storage is not available, the channel initiator issues the CSQX124E message and class 4 accounting and statistics are not available.

Server-connection channels

From IBM MQ 9.4.0 server-connection channels allocate message buffers in 64-bit storage, if they are transferring messages larger than 32 KB in size.

These buffers are freed if the channels require less than half the current buffer size for 10 consecutive messages, or a heartbeat is sent or received.

The value of MEMLIMIT sets an upper limit on how many concurrent server-connection channels can run. You should use a minimum value of MEMLIMIT=2G to ensure that the same number of channels can run as in earlier versions of IBM MQ, as well as providing some capacity for growth.

You can calculate an approximate value for MEMLIMIT by working out the peak maximum number of concurrently active server-connection channels, and for those channels the maximum message size you expect them to transfer. You should use MEMLIMIT=2GB as a starting point and round up.

For example, if you set the maximum number of concurrent server-connection channels to be 2,000 and each channel to have a maximum message size of 1MB, then server-connection channels are using a maximum of just under 2GB of 64-bit storage. As this is very close to 2GB then you should round up to MEMLIMIT=3G.

z/OS *Disk storage*

Use this topic when planning your disk storage requirements for log data sets, Db2 storage, coupling facility storage, and page data sets.

Work with your storage administrator to determine where to put the queue manager data sets. For example, your storage administrator may give you specific DASD volumes, or SMS storage classes, data classes, and management classes for the different data set types.

- Log data sets must be on DASD. These logs can have high I/O activity with a small response time and do not need to be backed up.
- Archive logs can be on DASD or tape. After they have been created, they might never be read again except in an abnormal situation, such as recovering a page set from a backup. They should have a long retention date.
- Page sets might have low to medium activity and should be backed up regularly. On a high use system, they should be backed up twice a day.
- BSDS data sets should be backed up daily; they do not have high I/O activity.

All data sets are similar to those used by Db2, and similar maintenance procedures can be used for IBM MQ.

See the following sections for details of how to plan your data storage:

- **Logs and archive storage**

  "How long do I need to keep archive logs" on page 175 describes how to determine how much storage your active log and archive data sets require, depending on the volume of messages that your IBM MQ system handles and how often the active logs are offloaded to your archive data sets.
- **Db2 storage**

  "Db2 storage" on page 192 describes how to determine how much storage Db2 requires for the IBM MQ data.
- **coupling facility storage**

  "Defining coupling facility resources" on page 182 describes how to determine how large to make your coupling facility structures.
- **Page set and message storage**

  "Planning your page sets and buffer pools" on page 157 describes how to determine how much storage your page data sets require, depending on the sizes of the messages that your applications exchange, on the numbers of these messages, and on the rate at which they are created or exchanged.

z/OS *Where to find more information about storage and performance requirements*

Use this topic as a reference to find more information about storage and performance requirements.

You can find more information from the following sources:

Table 20. Where to find more information about storage requirements

Topic	Where to look
System parameters	Using CSQ6SYSP and Customizing your queue managers
Storage required to install IBM MQ	Program Directory. Para enlaces de descarga de los directorios de programas, consulteIBM MQ for z/OS Archivos PDF del directorio de programas .
IEALIMIT and IEFUSI exits	See IEALIMIT and IEFUSI in the *z/OS:MVS Installation Exits* documentation.

Table 20. Where to find more information about storage requirements (continued)	
Topic	**Where to look**
Latest information	IBM MQ SupportPac website SupportPacs para IBM MQ y otras áreas de proyecto.
Workload management and defining goals through the service definition	*z/OS MVS Planning: Workload Management*

z/OS Planning your page sets and buffer pools

Information to help you with planning the initial number, and sizes of your page data sets, and buffer pools.

This topic contains the following sections:

- "Plan your page sets" on page 157
  - Page set usage
  - Number of page sets
  - Size of page sets
  - Planning for z/OS data set encryption
- "Calculate the size of your page sets" on page 158
  - Page set zero
  - Page set 01 - 99
  - Calculating the storage requirement for messages
- "Enabling dynamic page set expansion" on page 160
- "Defining your buffer pools" on page 162

Plan your page sets

Page set usage

For short-lived messages, few pages are normally used on the page set and there is little or no I/O to the data sets except at startup, during a checkpoint, or at shutdown.

For long-lived messages, those pages containing messages are normally written out to disk. This operation is performed by the queue manager in order to reduce restart time.

Separate short-lived messages from long-lived messages by placing them on different page sets and in different buffer pools.

Number of page sets

Using several large page sets can make the role of the IBM MQ administrator easier because it means that you need fewer page sets, making the mapping of queues to page sets simpler.

Using multiple, smaller page sets has a number of advantages. For example, they take less time to back up, and I/O can be carried out in parallel during backup and restart. However, consider that this adds a significant performance cost to the role of the IBM MQ administrator, who is required to map each queue to one of a much greater number of page sets.

Define at least five page sets, as follows:

- A page set reserved for object definitions (page set zero)
- A page set for system-related messages
- A page set for performance-critical long-lived messages

- A page set for performance-critical short-lived messages
- A page set for all other messages

"Defining your buffer pools" on page 162 explains the performance advantages of distributing your messages on page sets in this way.

Size of page sets

Define sufficient space in your page sets for the expected peak message capacity. Consider for any unexpected peak capacity, such as when a build-up of messages develops because a queue server program is not running. You can do this by allocating the page set with secondary extents or, alternatively, by enabling dynamic page set expansion. For more information, see "Enabling dynamic page set expansion" on page 160. It is difficult to make a page set smaller, so it is often better to allocate a smaller page set, and allow it to expand when needed.

When planning page set sizes, consider all messages that might be generated, including non-application message data. For example, trigger messages, event messages and any report messages that your application has requested.

The size of the page set determines the time taken to recover a page set when restoring from a backup, because a large page set takes longer to restore.

Note: Recovery of a page set also depends on the time the queue manager takes to process the log records written since the backup was taken; this time period is determined by the backup frequency. For more information, see "Planning for backup and recovery" on page 194.

Note: Page sets larger than 4 GB require the use of SMS extended addressability.

Planning for z/OS data set encryption

You can apply the z/OS data set encryption feature to page sets for queue managers running at IBM MQ for z/OS 9.1.4 or later.

You must allocate these page sets with EXTENDED attributes, and a data set key label that ensures the data is AES encrypted.

See the section, confidentiality for data at rest on IBM MQ for z/OS with data set encryption. for more information.

Calculate the size of your page sets

For queue manager object definitions (for example, queues and processes), it is simple to calculate the storage requirement because these objects are of fixed size and are permanent. For messages however, the calculation is more complex for the following reasons:

- Messages vary in size.
- Messages are transitory.
- Space occupied by messages that have been retrieved is reclaimed periodically by an asynchronous process.

Large page sets of greater than 4 GB that provide extra capacity for messages if the network stops, can be created if required. It is not possible to modify the existing page sets. Instead, new page sets with extended addressability and extended format attributes, must be created. The new page sets must be the same physical size as the old ones, and the old page sets must then be copied to the new ones. If backward migration is required, page set zero must not be changed. If page sets less than 4 GB are adequate, no action is needed.

Page set zero

Page set zero is reserved for object definitions.

For page set zero, the storage required is:

```
   (maximum number of local queue definitions x 1010)
      (excluding shared queues)
+  (maximum number of model queue definitions x 746)
+  (maximum number of alias queue definitions x 338)
+  (maximum number of remote queue definitions x 434)
+  (maximum number of permanent dynamic queue definitions x 1010)
+  (maximum number of process definitions x 674)
+  (maximum number of namelist definitions x 12320)
+  (maximum number of message channel definitions x 2026)
+  (maximum number of client-connection channel definitions x 5170)
+  (maximum number of server-connection channel definitions x 2026)
+  (maximum number of storage class definitions x 266)
+  (maximum number of authentication information definitions x 1010)
+  (maximum number of administrative topic definitions x 15000)
+  (total length of topic strings defined in administrative topic definitions)
```

Divide this value by 4096 to determine the number of records to specify in the cluster for the page set data set.

You do not need to allow for objects that are stored in the shared repository, but you must allow for objects that are stored or copied to page set zero (objects with a disposition of GROUP or QMGR).

The total number of objects that you can create is limited by the capacity of page set zero. The number of local queues that you can define is limited to 524 287.

Page sets 01 - 99

For page sets 01 - 99, the storage required for each page set is determined by the number and size of the messages stored on that page set. (Messages on shared queues are not stored on page sets.)

Divide this value by 4096 to determine the number of records to specify in the cluster for the page set data set.

Calculating the storage requirement for messages

This section describes how messages are stored on pages. Understanding this can help you calculate how much page set storage you must define for your messages. To calculate the approximate space required for all messages on a page set you must consider maximum queue depth of all the queues that map to the page set and the average size of messages on those queues.

Note: The sizes of the structures and control information given in this section are liable to change between major releases. For details specific to your release of IBM MQ, refer to SupportPac MP16 - IBM MQ for z/OS Planificación de capacidad y ajuste de and IBM MQ Family-Performance Reports

You must allow for the possibility that message "gets" might be delayed for reasons outside the control of IBM MQ (for example, because of a problem with your communications protocol). In this case, the "put" rate of messages might far exceed the "get" rate. This can lead to a large increase in the number of messages stored in the page sets and a consequent increase in the storage size demanded.

Each page in the page set is 4096 bytes long. Allowing for fixed header information, each page has 4057 bytes of space available for storing messages.

When calculating the space required for each message, the first thing you must consider is whether the message fits on one page (a short message) or whether it needs to be split over two or more pages (a long message). When messages are split in this way, you must allow for additional control information in your space calculations.

For the purposes of space calculation, a message can be represented as the following:

The message header section contains the message descriptor and other control information, the size of which varies depending on the size of the message. The message data section contains all the actual message data, and any other headers (for example, the transmission header or the IMS bridge header).

A minimum of two pages are required for page set control information which, is typically less than 1% of the total space required for messages.

Short messages

A short message is defined as a message that fits on one page.

Small messages are stored one on each page.

Long messages

If the size of the message data is greater than 3596 bytes, but not greater than 4 MB, the message is classed as a long message. When presented with a long message, IBM MQ stores the message on a series of pages, and stores control information that points to these pages in the same way that it would store a short message. This is shown in Figure 41 on page 160:

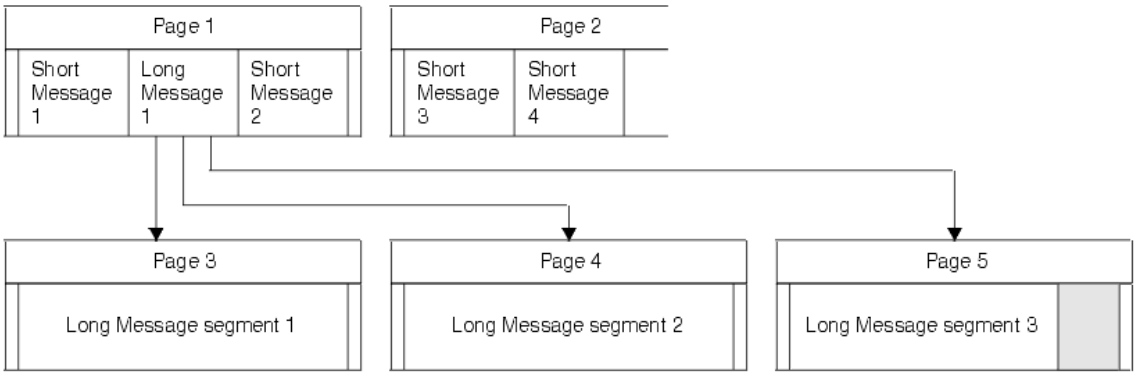


Figure 41. How IBM MQ stores long messages on page sets

Very long messages

Very long messages are messages with a size greater than 4 MB. These are stored so that each 4 MB uses 1037 pages. Any remainder is stored in the same way as a long message, as described above.

z/OS *Enabling dynamic page set expansion*

Page sets can be extended dynamically while the queue manager is running. A page set can have 123 extents, and can be spread over multiple disk volumes.

Each time a page set expands, a new data set extent is used. The queue manager continues to expand a page set when required, until the maximum number of extents has been reached, or until no more storage is available for allocation on eligible volumes.

Once page set expansion fails for one of the reasons above, the queue manager marks the page set for no further expansion attempts. This marking can be reset by altering the page set to EXPAND(SYSTEM).

Page set expansion takes place asynchronously to all other page set activity, when 90% of the existing space in the page set is allocated.

The page set expansion process formats the newly allocated extent and makes it available for use by the queue manager. However, none of the space is available for use, until the entire extent has been formatted. This means that expansion by a large extent is likely to take some time, and putting applications might 'block' if they fill the remaining 10% of the page set before the expansion has completed.

Sample thlqual.SCSQPROC(CSQ4PAGE) shows how to define the secondary extents.

To control the size of new extents, you use one of the following options of the EXPAND keyword of the DEFINE PSID and ALTER PSID commands:

- USER
- SYSTEM
- NONE

USER

Uses the secondary extent size specified when the page set was allocated. If a value was not specified, or if a value of zero was specified, dynamic page set expansion cannot occur.

Page set expansion occurs when the space in the page is 90% used, and is performed asynchronously with other page set activity.

This may lead to expansion by more than a single extent at a time.

Consider the following example: you allocate a page set with a primary extent of 100,000 pages and a secondary extent of 5000 pages. A message is put that requires 9999 pages. If the page set is already using 85,000 pages, writing the message crosses the 90% full boundary (90,000 pages). At this point, a further secondary extent is allocated to the primary extent of 100,000 pages, taking the page set size to 105,000 pages. The remaining 4999 pages of the message continue to be written. When the used page space reaches 94,500 pages, which is 90% of the updated page set size of 105,000 pages, another 5000 page extent is allocated, taking the page set size to 110,000 pages. At the end of the MQPUT, the page set has expanded twice, and 94,500 pages are used. None of the pages in the second page set expansion have been used, although they were allocated.

At restart, if a previously used page set has been replaced with a data set that is smaller, it is expanded until it reaches the size of the previously used data set. Only one extent is required to reach this size.

SYSTEM

Ignores the secondary extent size that was specified when the page set was defined. Instead, the queue manager sets a value that is approximately 10% of the current page set size. The value is rounded up to the nearest cylinder of DASD.

If a value was not specified, or if a value of zero was specified, dynamic page set expansion can still occur. The queue manager sets a value that is approximately 10% of the current page set size. The new value is rounded up depending on the characteristics of the DASD.

Page set expansion occurs when the space in the page set is approximately 90% used, and is performed asynchronously with other page set activity.

At restart, if a previously used page set has been replaced with a data set that is smaller, it is expanded until it reaches the size of the previously used data set.

NONE

No further page set expansion is to take place.

Related reference

ALTER PSID

DEFINE PSID

DISPLAYUSAGE

Defining your buffer pools

Use this topic to help plan the number of buffer pools you should define, and their settings.

This topic is divided into the following sections:

1. "Decide on the number of buffer pools to define" on page 162
2. "Decide on the settings for each buffer pool" on page 163
3. "Monitor the performance of buffer pools under expected load" on page 163
4. "Adjust buffer pool characteristics" on page 163

Decide on the number of buffer pools to define

You should define four buffer pools initially:

Buffer pool 0

Use for object definitions (in page set zero) and performance critical, system related message queues, such as the SYSTEM.CHANNEL.SYNCQ queue and the SYSTEM.CLUSTER.COMMAND.QUEUE and SYSTEM.CLUSTER.REPOSITORY.QUEUE queues.

However it is important to consider point "7" on page 164 in *Adjust buffer pool characteristics* if a large number of channels, or clustering, is to be used.

Use the remaining three buffer pools for user messages.

Buffer pool 1

Use for important long-lived messages.

Long-lived messages are those that remain in the system for longer than two checkpoints, at which time they are written out to the page set. If you have many long-lived messages, this buffer pool should be relatively small, so that page set I/O is evenly distributed (older messages are written out to DASD each time the buffer pool becomes 85% full).

If the buffer pool is too large, and the buffer pool never gets to 85% full, page set I/O is delayed until checkpoint processing. This might affect response times throughout the system.

If you expect few long-lived messages only, define this buffer pool so that it is sufficiently large to hold all these messages.

Buffer pool 2

Use for performance-critical, short-lived messages.

There is normally a high degree of buffer reuse, using few buffers. However, you should make this buffer pool large to allow for unexpected message accumulation, for example, when a server application fails.

Buffer pool 3

Use for all other (typically, performance noncritical) messages.

Queues such as the dead-letter queue, SYSTEM.COMMAND.* queues and SYSTEM.ADMIN.* queues can also be mapped to buffer pool 3.

Where virtual storage constraints exist, and buffer pools need to be smaller, buffer pool 3 is the first candidate for size reduction.

You might need to define additional buffer pools in the following circumstances:

- If a particular queue is known to require isolation, perhaps because it exhibits different behavior at various times.
  - Such a queue might either require the best performance possible under the varying circumstances, or need to be isolated so that it does not adversely affect the other queues in a buffer pool.
  - Each such queue can be isolated into its own buffer pool and page set.
- You want to isolate different sets of queues from each other for class-of-service reasons.

– Each set of queues might then require one, or both, of the two types of buffer pools 1 or 2, as described in Suggested definitions for buffer pool settings, necessitating creation of several buffer pools of a specific type.

Decide on the settings for each buffer pool

If you are using the four buffer pools described in “Decide on the number of buffer pools to define” on page 162, then Suggested definitions for buffer pool settings gives two sets of values for the size of the buffer pools.

The first set is suitable for a test system, the other for a production system or a system that will become a production system eventually. In all cases define your buffer pools with the **LOCATION**(*ABOVE*) attribute

Table 21. Suggested definitions for buffer pool settings

Definition setting	Test system	Production system
BUFFPOOL 0	1 050 buffers	50 000 buffers
BUFFPOOL 1	1 050 buffers	20 000 buffers
BUFFPOOL 2	1 050 buffers	50 000 buffers
BUFFPOOL 3	1 050 buffers	20 000 buffers

If you need more than the four suggested buffer pools, select the buffer pool (1 or 2) that most accurately describes the expected behavior of the queues in the buffer pool, and size it using the information in Suggested definitions for buffer pool settings.

Ensure that your MEMLIMIT is set high enough, so that all the buffer pools can be located above the bar.

Monitor the performance of buffer pools under expected load

You can monitor the usage of buffer pools by analyzing buffer pool performance statistics. In particular, you should ensure that the buffer pools are large enough so that the values of QPSTSOS, QPSTSTLA, and QPSTDMC remain at zero.

For further information, see Buffer manager data records.

Adjust buffer pool characteristics

Use the following points to adjust the buffer pool settings from “Decide on the settings for each buffer pool” on page 163, if required.

Use the performance statistics from “Monitor the performance of buffer pools under expected load” on page 163 as guidance.

1. If you are migrating from an earlier version of IBM MQ, only change your existing settings if you have more real storage available.
2. In general, bigger buffer pools are better for performance, and buffer pools can be much bigger if they are above the bar.

   However, at all times you should have sufficient real storage available so that the buffer pools are resident in real storage. It is better to have smaller buffer pools that do not result in paging, than big ones that do.

   Additionally, there is no point having a buffer pool that is bigger than the total size of the page sets that use it, although you should take into account page set expansion if it is likely to occur.
3. Aim for one page set per buffer pool, as this provides better application isolation.
4. If you have sufficient real storage, such that your buffer pools will never be paged out by the operating system, consider using page-fixed buffers in your buffer pool.

This is particularly important if the buffer pool is likely to undergo much I/O, as it saves the CPU cost associated with page-fixing the buffers before the I/O, and page-unfixing them afterwards.

5. There are several benefits to locating buffer pools above the bar even if they are small enough to fit below the bar. These are:
   - 31 bit virtual storage constraint relief - for example more space for common storage.
   - If the size of a buffer pool needs to be increased unexpectedly while it is being heavily used, there is less impact and risk to the queue manager, and its workload, by adding more buffers to a buffer pool that is already above the bar, than moving the buffer pool to above the bar and then adding more buffers.
6. Tune buffer pool zero and the buffer pool for short-lived messages (buffer pool 2) so that the 15% free threshold is never exceeded (that is, QPSTCBSL divided by QPSTNBUF is always greater than 15%). If more than 15% of buffers remain free, I/O to the page sets using these buffer pools can be largely avoided during normal operation, although messages older than two checkpoints are written to page sets.

   Attention: The optimum value for these parameters is dependent on the characteristics of the individual system. The values given are intended only as a guideline and might not be appropriate for your system.
7. SYSTEM.* queues which get very deep, for example SYSTEM.CHANNEL.SYNCQ, might benefit from being placed in their own buffer pool, if sufficient storage is available.

IBM MQ SupportPac MP16 - IBM MQ for z/OS Planificación de capacidad y ajuste de provides further information about tuning buffer pools.

z/OS Planning your logging environment

Use this topic to plan the number, size and placement of the logs, and log archives used by IBM MQ.

Logs are used to:

- Write recovery information about persistent messages
- Record information about units of work using persistent messages
- Record information about changes to objects, such as define queue
- Backup CF structures

and for other internal information.

The IBM MQ logging environment is established using the system parameter macros to specify options, such as: whether to have single or dual active logs, what media to use for the archive log volumes, and how many log buffers to have.

These macros are described in Create the bootstrap and log data sets and Tailor your system parameter module.

Note: If you are using queue sharing groups, ensure that you define the bootstrap and log data sets with SHAREOPTIONS(2 3).

This section contains information about the following topics:

z/OS *Log data set definitions*

Use this topic to decide on the most appropriate configuration for your log data sets.

This topic contains information to help you answer the following questions:

- Should your installation use single or dual logging?
- How many active log data sets do you need?
- “How large should the active logs be?” on page 166
- Active log placement

- “Active log encryption with z/OS data set encryption” on page 167

Should your installation use single or dual logging?

In general you should use dual logging for production, to minimize the risk of losing data. If you want your test system to reflect production, both should use dual logging, otherwise your test systems can use single logging.

With single logging data is written to one set of log data sets. With dual logging data is written to two sets of log data sets, so in the event of a problem with one log data set, such as the data set being accidentally deleted, the equivalent data set in the other set of logs can be used to recover the data.

With dual logging you require twice as much DASD as with single logging.

If you are using dual logging, then also use dual BSDSs and dual archiving to ensure adequate provision for data recovery.

Dual active logging adds a small performance cost.

Attention: Use of disk mirroring technologies, such as Metro Mirror, are not necessarily a replacement for dual logging and dual BSDS. If a mirrored data set is accidentally deleted, both copies are lost.

If you use persistent messages, single logging can increase maximum capacity by 10-30% and can also improve response times.

Single logging uses 2 - 310 active log data sets, whereas dual logging uses 4 - 620 active log data sets to provide the same number of active logs. Thus single logging reduces the amount of data logged, which might be important if your installation is I/O constrained.

How many active log data sets do you need?

The number of logs depends on the activities of your queue manager. For a test system with low throughput, three active log data sets might be suitable. For a high throughput production system you might want the maximum number of logs available, so, if there is a problem with offloading logs you have more time to resolve the problems.

You must have at least three active log data sets, but it is preferable to define more. For example, if the time taken to fill a log is likely to approach the time taken to archive a log during peak load, define more logs.

Note: Page sets and active log data sets are eligible to reside in the extended addressing space (EAS) part of an extended address volumes (EAV) and an archive log dataset can also reside in the EAS.

You should also define more logs to offset possible delays in log archiving. If you use archive logs on tape, allow for the time required to mount the tape.

Consider having enough active log space to keep a day's worth of data, in case the system is unable to archive because of lack of DASD or because it cannot write to tape. If all the active logs fill up, then IBM MQ is unable to process persistent messages or transactions. It is very important to have enough active log space.

It is possible to dynamically define new active log data sets as a way of minimizing the effect of archive delays or problems. New data sets can be brought online rapidly, using the **DEFINE LOG** command to avoid queue manager 'stall' due to lack of space in the active log.

If you want to define more than 31 active log data sets, you must configure your logging environment to use a version 2 format BSDS. Once a version 2 format BSDS is in use, up to 310 active log data sets can be defined for each log copy ring. See “Planning to increase the maximum addressable log range” on page 176 for information on how you convert to a version 2 format BSDS.

You can tell whether your queue manager is using a version 2 or higher BSDS, either by running the print log map utility (CSQJU004), or from the CSQJ034I message issued during queue manager initialization.

An end of log RBA range of FFFFFFFFFFFFFFFF, in the CSQJ034I message, indicates that a version 2, or higher, format BSDS is in use. An end of log RBA range of 0000FFFFFFFFFFFF, in the CSQJ034I message, indicates that a version 1 format BSDS is in use.

When a queue manager is using a version 2, or higher, format BSDS, it is possible to use the **DEFINE LOG** command to dynamically add more than 31 active log data sets to a log copy ring.

How large should the active logs be?

The maximum supported active log size, when archiving to disk or to tape, is 4 GB.

You should create active logs of at least 1 GB in size for production and test systems.

Important: You need to be careful when allocating data sets, because IDCAMS rounds up the size you allocate.

To allocate a 3 GB log specify one of the following options:

- Cylinders(4369)
- Megabytes(3071)
- TRACKS(65535)
- RECORD(786420)

Any one of these allocates 2.99995 GB.

To allocate a 4GB log specify one of the following options:

- Cylinders(5825)
- Megabytes(4095)
- TRACKS(87375)
- RECORD(1048500)

Any one of these allocates 3.9997 GB.

When using striped data sets, where the data set is spread across multiple volumes, the specified size value is allocated on each DASD volume used for striping. So, if you want to use 4 GB logs and four volumes for striping, you should specify:

- CYLinders(1456)
- Megabytes(1023)

Setting these attributes allocates 4*1456 = 5824 Cylinders or 4 * 1023 = 4092 Megabytes.

Note: Striping is supported when using extended format data sets. This is usually set by the storage manager.

See Increasing the size of the active log for information on carrying out the procedure.

Active log placement

You should work with your storage management team to set up storage pools for the queue managers. You need to consider:

- A naming convention, so the queue managers use the correct SMS definitions.
- Space required for active and archive logs. Your storage pool should have enough space for the active logs from a whole day.
- Performance and resilience to failures.

For performance reasons you should consider striping your active log data sets. The I/O is spread across multiple volumes and reduces the I/O response times, leading to higher throughput. See the preceding text for information about allocating the size of the active logs when using striping.

You should review the I/O statistics using reports from RMF or a similar product. Perform the review of these statistics monthly (or more frequently) for the IBM MQ data sets, to ensure there are no delays due to the location of the data sets.

In some situations, there can be much IBM MQ page set I/O, and this can impact the IBM MQ log performance if they are located on the same DASD.

If you use dual logging, ensure that each set of active and archive logs is kept apart. For example, allocate them on separate DASD subsystems, or on different devices.

This reduces the risk of them both being lost if one of the volumes is corrupted or destroyed. If both copies of the log are lost, the probability of data loss is high.

When you create a new active log data, set you should preformat it using CSQJUFMT. If the log is not preformatted, the queue manager formats the log the first time it is used, which impacts the performance.

With older DASD with large spinning disks, you had to be careful which volumes were used to get the best performance.

With modern DASD, where data is spread over many PC sized disks, you do not need to worry so much about which volumes are used.

Your storage manager should be checking the enterprise DASD to review and resolve any performance problems. For availability, you might want to use one set of logs on one DASD subsystem, and the dual logs on a different DASD subsystem.

Active log encryption with z/OS data set encryption

You can apply the z/OS data set encryption feature to active log data sets for queue managers running at IBM MQ for z/OS 9.1.4 or later.

You must allocate these active log data sets with EXTENDED attributes, and a data set key label that ensures the data is AES encrypted.

See the section, confidentiality for data at rest on IBM MQ for z/OS with data set encryption. for more information.

z/OS *Using MetroMirror with IBM MQ*

IBM Metro Mirror, previously known as Synchronous Peer to Peer Remote Copy (PPRC), is a synchronous replication solution between two storage subsystems, where write operations are completed on both the primary and secondary volumes before the write operation is considered to be complete. Metro Mirror can be used in environments that require no data loss in the event of a storage subsystem failure.

Supported data set types

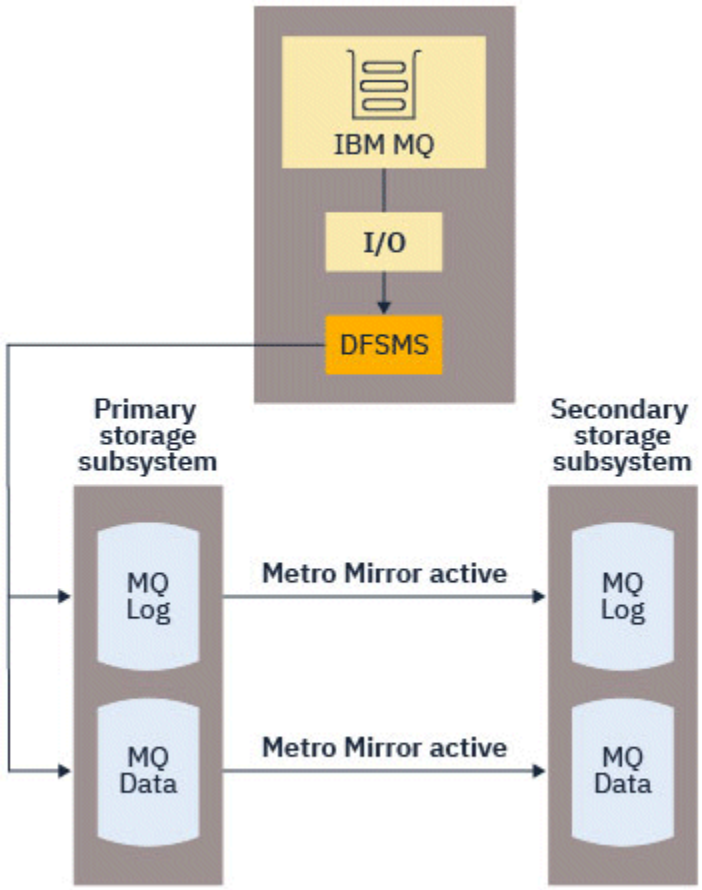
All of the following IBM MQ data set types can be replicated using Metro Mirror. However, exactly which ones are replicated depends on the availability requirements of your enterprise:

- Active logs
- Archive logs
- Bootstrap data set (BSDS)
- Page sets
- Shared message data set (SMDS)
- Data sets used for configuration, for example, in the CSQINP* DD cards on the MSTR JCL

Using zHyperWrite with IBM MQ active logs

When a write is made to a data set that is replicated using Metro Mirror, the write is first made to the primary volume, and then replicated to the secondary volume. This replication is done by the storage subsystem and is transparent to the application that issued the write, for example IBM MQ.

This process is illustrated in the following diagram.

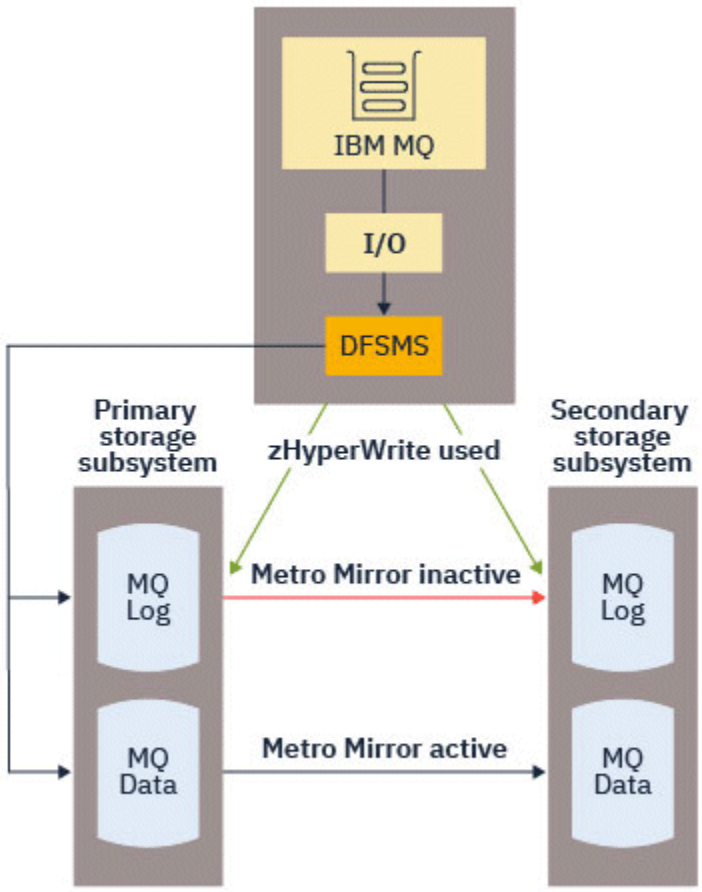


Because both writes to the primary and secondary storage subsystems need to complete before the write returns to IBM MQ, use of Metro Mirror can have a performance impact. You need to balance this performance impact against the availability benefits of using Metro Mirror.

The IBM MQ active logs are most sensitive to the performance impact of using Metro Mirror. IBM MQ allows use of zHyperWrite with the active logs to help reduce this performance impact.

zHyperWrite is a storage subsystem technology that works with z/OS to reduce the performance impact of writes made to data sets that are replicated using Metro Mirror. When zHyperWrite is used, the write to the primary and secondary volumes are issued in parallel at the Data Facility Storage Management Subsystem (DFSMS) level, instead of sequentially at the storage subsystem level, thereby reducing the performance impact.

The following diagram illustrates zHyperWrite being used for the active logs, and Metro Mirror being used for the other IBM MQ data set types. Note that if a zHyperWrite write fails, DFSMS will transparently reissue the write using Metro Mirror.



zHyperWrite on IBM MQ, is supported only on the active log data sets.

In order to use zHyperWrite with the active logs, you need to:

- Configure IBM MQ to use zHyperWrite, and
- The active logs need to be on zHyperWrite capable volumes

You can configure IBM MQ to use zHyperWrite by using one of the following methods:

- Specify ZHYWRITE(YES) in the system parameter module.
- Issue the command SET LOG ZHYWRITE(YES).

Set the following conditions for active log data sets to be on zHyperWrite capable volumes:

- Enable the volumes for Metro Mirror, and the volumes support zHyperWrite
- Ensure that the volumes are HyperSwap enabled
- Specify HYPERWRITE=YES in the IECIOSxx parameter

V 9.4.0 Prior to IBM MQ 9.4.0, if all the preceding conditions are met, then writes to the active logs are enabled for zHyperWrite. If one, or more, of these conditions are not met, IBM MQ writes to the active logs as normal, and Metro Mirror replicates the writes if it is configured.

V 9.4.0 From IBM MQ 9.4.0, if ZHYWRITE(YES) is specified, then IBM MQ always attempts to use zHyperWrite when writing to the active logs, regardless of whether the logs are on zHyperWrite capable volumes. If the logs are not on zHyperWrite capable volumes then Metro Mirror replicates the writes if it is configured. There are no negative effects of attempting to use zHyperWrite if the logs are not on zHyperWrite capable volumes

Notes:

- IBM MQ does not require that all active log data sets are on zHyperWrite capable volumes.

  If IBM MQ detects that some active log data sets are on zHyperWrite capable volumes, and others are not, it issues message CSQJ166E and carries on processing.
- IBM MQ checks whether active log data sets are zHyperWrite capable when the data sets are first opened.

  Log data sets are opened either at queue manager start up, or when dynamically adding using the DEFINE LOG command. If the log data sets are made zHyperWrite capable while a queue manager has them open, the queue manager will not detect this until it has been restarted.

You can use the output of the DISPLAY LOG command to indicate whether the current active log data sets are zHyperWrite capable. The following example shows that both of the data sets are zHyperWrite capable. If the queue manager has been configured with ZHYWRITE(YES), writes to these logs would be enabled for zHyperWrite:

```
Copy %Full zHyperWrite DSName
 1      4  CAPABLE     MQTST.SUBSYS.MQDL.LOGCOPY1.DS001
 2      4  CAPABLE     MQTST.SUBSYS.MQDL.LOGCOPY2.DS001
```

z/OS V 9.4.0 ***Rendimiento de registro más rápido con zHyperLink***

zHyper La tecnología de enlace está diseñada para reducir la latencia de entrada/salida (E/S) al proporcionar una ruta de comunicación rápida, confiable y directa entre la CPU y el dispositivo de E/S.

Visión general dezHyper Enlace

El enlace zHyperpuede mejorar el rendimiento del registro activo y reducir el tiempo de transacción de IBM MQ hasta 3.5 veces. Este objetivo se logra instalandozHyper Adaptadores de enlace en elz/OS anfitrión, seleccioneIBM hardware de almacenamiento y conectarlos usandozHyper Cables de enlace. Esto crea una conexión punto a punto entre la CPU y el dispositivo de E/S, lo que reduce el tiempo de respuesta de E/S hasta 10 veces, en comparación conIBM FICON® de alto rendimiento (zHPF). Este tiempo de respuesta bajo se consigue utilizando solicitudes de E/S síncronas.

Las ventajas de las E/S síncronas sobre las E/S asíncronas

ElIBM MQ La tarea del registrador consiste en un bucle que espera el siguiente dato que debe escribirse en el registro. Cuando esos datos están disponibles, el registrador programa la escritura, espera a que se complete y luego pasa al siguiente dato.

La E/S tradicional es más lenta que la CPU, por lo que es más eficiente realizar la E/S de forma asincrónica para liberar la CPU para otras tareas. Por lo tanto, la E/S asíncrona tradicional requiere que se suspenda la tarea del registrador hasta que se complete la escritura. Cuando se completa la escritura, la tarea del registrador debe esperar a que una CPU esté disponible, lo que agrega un breve retraso de reenvío, así como retrasos causados por la repoblación de la memoria caché de la CPU.

zHyper Link proporciona tiempos de E/S mucho más rápidos, que están más cerca de la velocidad de la CPU, por lo tanto, conzHyper El enlace y la E/S se pueden realizar de forma sincrónica, lo que significa que la tarea del registrador no se suspende durante la operación de escritura, lo que elimina los retrasos relacionados con el reenvío y la caché.

Mientras se produce la escritura, la tarea del registrador sigue utilizando activamente la CPU, lo que aumenta el uso de la CPU en comparación con la E/S tradicional.

Si el gestor de colas intenta utilizarzHyper Enlace, y elzHyper La escritura del enlace falla, por ejemplo debido a problemas de configuración, y entonces el administrador de colas recurre de forma transparente a la E/S tradicional.

Requisitos mínimos de hardware

- IBM z14 o posterior

- DS8880 o posterior

Requisitos de software

- zHyperLink Express está soportado en z/OS 2.3 o posterior.
- La imagen de z/OS debe ejecutarse en una LPAR, no como invitado en IBM z/VM®.
- zHyperLink requiere que se habilite IBM z High-Performance FICON (zHPF).

Utilización de zHyperLink con registros activos de IBM MQ

Para usarzHyper Para vincular con los registros activos de un administrador de colas, debe:

- ConfigurarIBM MQ usarzHyper Enlace, y
- Asegúrese de que los registros activos estén activadoszHyper Vincular volúmenes capaces.

  Ver Empezar conIBMzHyper Enlace paraz/OS para más información.

Puedes configurarIBM MQ usarzHyper Enlace utilizando uno de los siguientes métodos:

- Especifique ZHYLINK(YES) en los parámetros de registro.
- Emita el mandato SET LOG ZHYLINK(YES).

Notas:

- El enlace zHyperrequiere que zHyperWrite esté activado. Esto significa que para utilizar ZHYLINK, ZHYWRITE también debe estar activado en los parámetros de registro. Cuando sólo se especifica ZHYLINK (YES) cuando se establece ZHYWRITE (NO) en el gestor de colas, el parámetro ZHYWRITE se altera temporalmente automáticamente en YES.
- Si se intenta establecer explícitamente ZHYLINK (YES) con ZHYWRITE (NO), se producirá una finalización anómala del mandato SET LOG.
- Si se establece ZHYLINK=YES en los ZPRM, se altera temporalmente ZHYWRITE en YES.

Si tiene algún problema, consulte Resolución de problemas de zHyperEnlace para obtener más información.

IBM MQ no requiere que todos los conjuntos de datos de registro activo estén en volúmenes con capacidad de enlace zHyper, pero se le recomienda que lo haga. Si IBM MQ detecta que algunos conjuntos de datos de registro activo están en volúmenes con capacidad de enlace zHypery otros no, emite el mensaje CSQJ601E y continúa el proceso.

IBM MQ comprueba si los conjuntos de datos de registro activo tienen capacidad de enlace zHypercuando los conjuntos de datos se abren por primera vez. Los conjuntos de datos de registro se abren al iniciar el gestor de colas o al añadir dinámicamente utilizando el mandato DEFINE LOG . Si los conjuntos de datos de registro tienen capacidad de enlace zHypermientras un gestor de colas los tiene abiertos, el gestor de colas no lo detecta hasta que se ha reiniciado.

Si se especifica ZHYLINK (YES), IBM MQ siempre intenta utilizar el enlace zHyperal grabar en los registros activos, independientemente de si los registros están en volúmenes con capacidad de enlace zHyper. No hay efectos negativos al intentar utilizar el enlace zHypersi los registros no están en volúmenes con capacidad de enlace zHyper.

Puede utilizar la salida del mandato DISPLAY LOG para indicar el estado de zHyperLink para los conjuntos de datos de registro activo actuales:

```
Copy %Full zHyperWrite    Encrypted          DSName
  1     81 YES          NO                MQTST.SUBSYS.MQDL.LOGCOPY1.DS001
  2     81 YES            NO                MQTST.SUBSYS.MQDL.LOGCOPY2.DS001
Copy zHyperLink
  1  YES
  2  YES
```

El estado del enlace zHyperes uno de los siguientes:

SÍ
El enlace zHyperestá habilitado en el gestor de colas y se intentará en todas las grabaciones.

NO
El enlace zHyperno está habilitado en el gestor de colas y el conjunto de datos **no está** en zHypervolúmenes con capacidad de enlace.

CAPABLE
El enlace zHyperno está habilitado en el gestor de colas y el conjunto de datos **está** en un volumen con capacidad de enlace zHyper.

Hay varias estadísticas SMF adicionales para monitorear y comprenderzHyper Rendimiento del enlace; verzHyper Estadísticas de enlaces para detalles.

Sesiones de escritura

Cuando usaszHyper Enlace, uno o mászHyper Las sesiones de escritura de enlaces se establecen con el DASD. El DASD actual admite un máximo de 64 sesiones de escritura simultáneas, por lo que debe considerar cuidadosamente qué administradores de colas habilitazHyper Enlace y si otros subsistemas, como por ejemploDb2 también están usandozHyper Enlace para escribir al mismo DASD. Si se queda sin sesiones de escritura disponibles, el administrador de colas vuelve automáticamente a utilizar E/S asincrónicas tradicionales.

Puedes calcular el número dezHyper Enlace las sesiones de escritura de la siguiente manera:

```
Number of log copies (either 1 or 2) * number of stripes per log copy * 2
if Metro Mirror (PPRC) is used.
```

Por lo tanto, un administrador de colas en modo de registro único con una franja y sinMetro Mirror utiliza una única sesión de escritura. Un administrador de colas en modo de registro dual, con dos franjas y PPRC utiliza 8 sesiones de escritura.

Nota: MientrasMetro Mirror da como resultado que se utilicen el doble de sesiones de escritura, esas sesiones de escritura se dividen equitativamente entre los dos DASD reflejados.

z/OS *Planning your log archive storage*

Use this topic to understand the different ways of maintaining your archive log data sets.

You can place archive log data sets on standard-label tapes, or DASD, and you can manage them by data facility hierarchical storage manager (DFHSM). Each z/OS logical record in an archive log data set is a VSAM control interval from the active log data set. The block size is a multiple of 4 KB.

Archive log data sets are dynamically allocated, with names chosen by IBM MQ. The data set name prefix, block size, unit name, and DASD sizes needed for such allocations are specified in the system parameter module. You can also choose, at installation time, to have IBM MQ add a date and time to the archive log data set name.

It is not possible to specify with IBM MQ, specific volumes for new archive logs, but you can use Storage Management routines to manage this. If allocation errors occur, offloading is postponed until the next time offloading is triggered.

If you specify dual archive logs at installation time, each log control interval retrieved from the active log is written to two archive log data sets. The log records that are contained in the pair of archive log data sets are identical, but the end-of-volume points are not synchronized for multivolume data sets.

Should your archive logs reside on tape or DASD?

When deciding whether to use tape or DASD for your archive logs, there are a number of factors that you should consider:

- Review your operating procedures before deciding about tape or disk. For example, if you choose to archive to tape, there must be enough tape drive when they are required. After a disaster, all subsystems might want tape drives and you might not have as many free tape drives as you expect.
- During recovery, archive logs on tape are available as soon as the tape is mounted. If DASD archives have been used, and the data sets migrated to tape using hierarchical storage manager (HSM), there is a delay while HSM recalls each data set to disk. You can recall the data sets before the archive log is used. However, it is not always possible to predict the correct order in which they are required.
- When using archive logs on DASD, if many logs are required (which might be the case when recovering a page set after restoring from a backup) you might require a significant quantity of DASD to hold all the archive logs.
- In a low-usage system or test system, it might be more convenient to have archive logs on DASD to eliminate the need for tape mounts.
- Both issuing a RECOVER CFSTRUCT command, and backing out a persistent unit of work, result in the log being read backwards. Tape drives with hardware compression perform badly on operations that read backwards. Plan sufficient log data on DASD to avoid reading backwards from tape.

Archiving to DASD offers faster recoverability but is more expensive than archiving to tape. If you use dual logging, you can specify that the primary copy of the archive log go to DASD and the secondary copy go to tape. This increases recovery speed without using as much DASD, and you can use the tape as a backup.

See “Changing the storage medium for archive logs” on page 174 for details of how you archive your logs from tape to DASD, and how you carry out the reverse process.

Archiving to tape

If you choose to archive to a tape device, IBM MQ can extend to a maximum of 20 volumes.

If you are considering changing the size of the active log data set so that the set fits on one tape volume, note that a copy of the BSDS is placed on the same tape volume as the copy of the active log data set. Adjust the size of the active log data set downward to offset the space required for the BSDS on the tape volume.

If you use dual archive logs on tape, it is typical for one copy to be held locally, and the other copy to be held off-site for use in disaster recovery.

Archiving to DASD volumes

IBM MQ requires that you catalog all archive log data sets allocated on non-tape devices (DASD). If you choose to archive to DASD, the CATALOG parameter of the CSQ6ARVP macro must be YES. If this parameter is NO, and you decide to place archive log data sets on DASD, you receive message CSQJ072E each time an archive log data set is allocated, although IBM MQ still catalogs the data set.

If the archive log data set is held on DASD, the archive log data sets can extend to another volume; multivolume is supported.

If you choose to use DASD, make sure that the primary space allocation (both quantity and block size) is large enough to contain either the data coming from the active log data set, or that from the corresponding BSDS, whichever is the larger of the two.

This minimizes the possibility of unwanted z/OS X'B37' or X'E37' abend codes during the offload process. The primary space allocation is set with the PRIQTY (primary quantity) parameter of the CSQ6ARVP macro.

Archive log data sets can exist on large or extended-format sequential data sets. SMS ACS routines now use DSNTYPE(LARGE) or DSNTYPE(EXT).

IBM MQ supports allocation of archive logs as extended format data sets. When extended format is used, the maximum archive log size is increased from 65535 tracks to the maximum active log size of 4GB. Archive logs are eligible for allocation in the extended addressing space (EAS) of extended address volumes (EAV).

Where the required hardware and software levels are available, allocating archive logs to a data class defined with COMPACTION using zEDC might reduce the disk storage required to hold archive logs. For more information, see IBM MQ for z/OS: Reducing storage occupancy with IBM zEnterprise Data Compression (zEDC) and zEnterprise Data Compression (zEDC) for more information.

The z/OS data set encryption feature can be applied to archive logs for queue managers running on IBM MQ. These archive logs must be allocated through Automatic Class Selection (ACS) routines to a data class defined with EXTENDED attributes, and a data set key label that ensures the data is AES encrypted.

Using SMS with archive log data sets

If you have MVS/DFP storage management subsystem (DFSMS) installed, you can write an Automatic Class Selection (ACS) user-exit filter for your archive log data sets, which helps you convert them for the SMS environment.

Such a filter, for example, can route your output to a DASD data set, which DFSMS can manage. You must exercise caution if you use an ACS filter in this manner. Because SMS requires DASD data sets to be cataloged, you must make sure the CATALOG DATA field of the CSQ6ARVP macro contains YES. If it does not, message CSQJ072E is returned; however, the data set is still cataloged by IBM MQ.

For more information about ACS filters, see Data sets that DFSMShsm dynamically allocates during aggregate backup processing.

z/OS *Changing the storage medium for archive logs*

The procedure for changing the storage medium used by archive logs.

About this task

This task describes how to change the storage medium used for archive logs, for example moving from archiving to tape to archiving to DASD.

You have a choice of how to make the changes:

1. Make the changes only using the CSQ6ARVP macro so that they are applied from the next time the queue manager restarts.
2. Make the changes using the CSQ6ARVP macro, and dynamically using the SET ARCHIVE command. This means that the changes apply from the next time the queue manager archives a log file, and persist after the queue manager restarts.

Procedure

1. Changing so archive logs are stored on DASD instead of tape:
   a) Read the section "Archiving to DASD volumes" on page 173 and review the CSQ6ARVP parameters.
   b) Make changes to the following parameters in CSQ6ARVP
      - Update the UNIT and, if necessary, the UNIT2 parameters.
      - Update the BLKSIZE parameter, as the optimal setting for DASD differs from tape.
      - Set the PRIQTY and SECQTY parameters to be large enough to hold the largest of the active log or BSDS.
      - Set the CATALOG parameter to be YES.
      - Confirm the ALCUNIT setting is what you want. You should use BLK, because it is independent of the device type.
      - Set the ARCWTOR parameter to NO if it is not already.
2. Changing so archive logs are stored on tape instead of DASD:
   a) Read the section "Archiving to tape" on page 173, and review the CSQ6ARVP parameters.
   b) Make changes to the following parameters in CSQ6ARVP:

- Update the UNIT and, if necessary, the UNIT2 parameters.
- Update the BLKSIZE parameter, as the optimal setting for tape differs from DASD.
- Confirm the ALCUNIT setting is what you want. You should use BLK, because it is independent of the device type.
- Review the setting of the ARCWTOR parameter.

z/OS *How long do I need to keep archive logs*

Use the information in this section to help you plan your backup strategy.

You specify how long archive logs are kept in days , using the ARCRETN parameter in USING CSQ6ARVP or the SET SYSTEM command. After this period the data sets can be deleted by z/OS.

You can manually delete archive log data sets when they are no longer needed.

- The queue manager might need the archive logs for recovery.

  The queue manager can only keep the most recent 1000 archives in the BSDS, When the archive logs are not in the BSDS they cannot be used for recovery, and are only of use for audit, analysis, or replay type purposes.
- You might want to keep the archive logs so that you can extract information from the logs. For example, extracting messages from the log, and reviewing which user ID put or got the message.

The BSDS contains information on logs and other recovery information. This data set is a fixed size. When the number of archive logs reaches the value of MAXARCH in CSQ6LOGP, or when the BSDS fills up, the oldest archive log information is overwritten.

There are utilities to remove archive log entries from the BSDS, but in general, the BSDS wraps and overlays the oldest archive log record.

When is an archive log needed

You need to back up your page sets regularly. The frequency of backups determines which archive logs are needed in the event of losing a page set.

You need to back up your CF structures regularly. The frequency of backups determines which archive logs are needed in the event of losing data in the CF structure.

The archive log might be needed for recovery. The following information explains when the archive log might be needed, where there are problems with different IBM MQ resources.

Loss of a page set

You must recover your system from your backup and restart the queue manager.

You need the logs from when the backup was taken, as well as up to three log data sets prior to the backup being taken.

All LPARs lose connectivity to a CF structure, or the structure is unavailable

Use the RECOVER CFSTRUCT command to recover the structure.

Structure recovery requires the logs from all queue managers that have accessed the structure since the last backup (back to the time when the backup was taken) plus the structure backup itself in the log of the queue manager that took the backup.

If you have been doing frequent backups of the CF structures, the data should be in active logs, and you should not need archive logs.

If there is no recent backup of the CF structure, you might need archive logs.

Note: All non persistent messages will be lost; all persistent messages will be re-created by performing the following tasks:

1. Reading the last CF structure backup from the log
2. Reading the logs from all queue managers that have used the structure

3. Merging updates since the backup

Administration structure rebuild

If you need to rebuild the administration structure, the information is read from the last checkpoint of the log for each queue manager in the QSG.

If a queue manager is not active, another queue manager in the QSG reads the log.

You should not need archive logs.

Loss of an SMDS data set

If you lose an SMDS data set, or the data set gets corrupted, the data set becomes unusable and the status for it is set to FAILED. The CF structure is unchanged.

In order to restore the SMDS data set, you need to:

1. Redefine the SMDS data set, and
2. Recover the CF structure by issuing the RECOVER CFSTRUCT command.

   Note: All non persistent messages on the CF structure will be lost; all persistent messages will be restored.

The requirement for queue manager logs is the same as for recovering from a structure that is unavailable.

z/OS *Planning to increase the maximum addressable log range*

You can increase the maximum addressable log range by configuring your queue manager to use a larger log relative byte address (RBA).

The log RBA size was increased from IBM MQ for z/OS 8.0. For an overview of this change, see Larger log Relative Byte Address.

Queue managers created at IBM MQ 9.3.0 or later, have 8 byte log RBA enabled by default and, therefore, do not require conversion.

You can convert your queue managers to use 8 byte log RBA values at any time. A queue sharing group can contain some queue managers with 8 byte log RBA enabled, and some queue managers with 6 byte log RBA enabled.

Undoing the change

The change cannot be backed out.

How long does it take?

The change requires a queue manager restart. Stop the queue manager, run the CSQJUCNV utility against the bootstrap data set (BSDS), or data sets, to create new data sets, rename these bootstrap data sets, and restart the queue manager. The CSQJUCNV utility usually takes a few seconds to run.

What impact does this have?

- With 8 byte log RBA in use, every write of data to the log data sets has additional bytes. Therefore, for a workload consisting of persistent messages there is a small increase in the amount of data written to the logs.
- Data written to a page set, or coupling facility (CF) structure, is not affected.

Related tasks

Implementing the larger log Relative Byte Address

z/OS Planning your channel initiator

The channel initiator provides communications between queue managers, and runs in its own address space.

There are two types of connections:

1. Application connections to a queue manager over a network. These are known as client channels.
2. Queue manager to queue manager connections. These are known as MCA channels.

Listeners

A channel listener program listens for incoming network requests and starts the appropriate channel when that channel is needed. To process inbound connections the channel initiator needs at least one IBM MQ listener task configured. A listener can either be a TCP listener, or a LU 6.2 listener.

Each listener requires a TCP port or LU name.

Note that you can have more than one listener for each channel initiator.

TCP/IP

A channel initiator can operate with more than one TCP stack on the same z/OS image. For example, one TCP stack could be for internal connections, and another TCP stack for external connections.

When you define an output channel:

1. You set the destination host and port of the connection. This can be either:
   - an IP address, for example `10.20.4.6`
   - a host name, for example `mvs-prod.myorg.com`

   If you use a host name to specify the destination, IBM MQ uses the Domain Name System (DNS) to resolve the IP address of the destination.
2. If you are using multiple TCP stacks you can specify the **LOCLADDR** parameter on the channel definition, which specifies the IP Stack address to be used.

You should plan to have a highly available DNS server, or DNS servers. If the DNS is not available, outbound channels might not be able to start, and channel authentication rules that map an incoming connection using a host name cannot be processed.

APPC and LU 6.2

If you are using APPC, the channel initiator needs an LU name, and configuration in APPC.

Queue sharing groups

To provide a single system image, and allow an incoming IBM MQ connection request to go to any queue manager in the queue sharing group, you need to do some configuration. For example:

1. A hardware network router. This router has one IP address seen by the enterprise, and can route the initial request to any queue manager connected to this hardware.
2. A Virtual IP address (VIPA). An enterprise wide IP address is specified, and that address can be routed to any one of the TCP stacks in a sysplex. The TCP stack can then route it to any listening queue manager in the sysplex.

Protecting IBM MQ traffic

You can configure IBM MQ to use TLS connections to protect data on the wire. To use TLS you need to use digital certificates and key rings.

You also need to work with the personnel at the remote end of the channel, to ensure that you have compatible IBM MQ definitions and compatible certificates.

You can control which connections can connect to IBM MQ and the user ID, based on

- IP address
- Client user ID
- Remote queue manager, or
- Digital certificate (see Channel Authentication Records)

It is also possible to restrict client applications by ensuring that they supply a valid user ID and password (see Connection Authentication).

You can get the channel initiator working, and then configure each channel to use TLS, one at a time.

Monitoring the channel initiator

There are MQSC commands that give information about the channel initiator and channels:

- The DISPLAY CHINIT command gives information about the channel initiator, and active listeners.
- The DISPLAY CHSTATUS command displays the activity and status of a channel.

The channel initiator can also produce SMF records with information about the channel initiator tasks and channel activity. See “Planning for channel initiator SMF data” on page 179 for more information.

The channel initiator emits messages to the job log when channels start and stop. Automation in your enterprise can use these messages to capture status. As some channels are active for only a few seconds, many messages can be produced. You can suppress these messages either by using the z/OS message processing facility, or by setting **EXCLMSG** with the SET SYSTEM command.

Configuring your IBM MQ channel definitions

When you have many queue managers connected together it can be hard to manage all the object definitions. Using IBM MQ clustering can simplify this.

You specify two queue managers as full repositories. Other queue managers need one connection to, and one connection from, one of the repositories. When connections to other queue managers are needed, the queue manager creates and starts channels automatically.

If you are planning to have a large number of queue managers in a cluster, you should plan to have queue managers that act as dedicated repositories and have no application traffic.

See “Planificación de sus gestores de colas y clústeres distribuidos” on page 20 for more information.

Actions before you configure the channel initiator

1. Decide if you are using TCP/IP or APPC.
2. If you are using TCP, allocate at least one port for IBM MQ.
3. If you need a a DNS server, configure the server to be highly available if required.
4. If you are using APPC, allocate an LU name, and configure APPC.

Actions after you have configured the channel initiator, before you go into production

1. Plan what connections you will have:
   a. Client connections from remote applications.
   b. MCA channels to and from other queue managers. Typically you have a channel to and from each remote queue manager.
2. Set up clustering, or join an existing clustering environment.

3. Consider whether you need to use multiple TCP stacks, VIPA, or an external router for availability in front of the channel initiator.
4. If you are planning on using TLS:
   a. Set up the key ring
   b. Set up certificates
5. If you are planning on using channel authentication:
   a. Decide the criteria for mapping inbound sessions to MCA user IDs
   b. Enable reverse DNS lookup by setting the queue manager parameter **REVDNS**
   c. Review security. For example, delete the default channels, and specify user IDs with only the necessary authority in the **MCAUSER** attribute for a channel.
6. Capture the accounting and statistics SMF records produced by the channel initiator and post process them.
7. Automate the monitoring of job log messages.
8. If necessary, tune your network environment to improve throughput. With TCP, large send and receive buffers improve throughput. You can force MQ to use specific TCP buffer sizes using the commands:

   ```
   RECOVER QMGR(TUNE CHINTCPRBDYNSZ nnnnn)
   RECOVER QMGR(TUNE CHINTCPSBDYNSZ nnnnn)
   ```

   which sets the SO_RCVBUF, and SO_SNDBUF, for the channels to the size in bytes specified in nnnnn.

Related concepts

“Planning for your queue manager” on page 148
When you are setting up a queue manager, your planning should allow for the queue manager to grow, so that the queue manager meets the needs of your enterprise.

z/OS Planning for channel initiator SMF data

You need to plan the implementation of collecting SMF data for the channel initiator.

The channel initiator produces two types of record:

- Statistics data with information about the channel initiator and the tasks within it.
- Channel accounting data with information similar to the DISPLAY CHSTATUS command.

You start collecting statistics data using the command:

```
START TRACE(STAT) CLASS(4)
```

and stop it using the command:

```
STOP TRACE(STAT) CLASS(4)
```

You start collecting accounting data using the command:

```
START TRACE(ACCTG) CLASS(4)
```

and stop it using the command:

```
STOP TRACE(ACCTG) CLASS(4)
```

You can control which channels have accounting data collected for using the **STATCHL** attribute on the channel definition or the queue manager.

- For client channels, you must set **STATCHL** at the queue manager level.

- For automatically defined cluster sender channels, you can control the collection of accounting data with the **STATACLS** queue manager attribute.

The default value of **STATCHL** for the queue manager is OFF. In order to collect channel accounting data you must change the value of **STATCHL** from the default on either the queue manager or channel definition, in addition to starting class 4 accounting trace.

The SMF records are produced when:

- From IBM MQ for z/OS 9.3.0 onwards, the time interval indicated by the CSQ6SYSP **STATIME** or **ACCTIME** parameters has elapsed; or, if **STATIME** or **ACCTIME** is zero on the SMF data collection broadcast. The requests to collect SMF data for the channel initiator and the queue manager are synchronized.
- A `STOP TRACE(ACCTG) CLASS(4)` or `STOP TRACE(STAT) CLASS(4)` command is issued, or
- The channel initiator is shut down. At this point any SMF data is written out.

If a channel stops during the SMF interval, accounting data is written to SMF the next time the SMF processing runs. If a client connects, does some work and disconnects, then reconnects and disconnects, there are two sets of channel accounting data produced.

The statistics data normally fits into one SMF record, however, multiple SMF records might be created if a large number of tasks are in use.

Accounting data is gathered for each channel for which it is enabled, and normally fits into one SMF record. However, multiple SMF records might be created if a large number of channels are active.

The cost of collecting the channel initiator SMF data is small. Typically the increase in CPU usage is under a few percent, and often within measurement error.

Before you use this function you need to work with your z/OS systems programmer to ensure that SMF has the capacity for the additional records, and that they change their processes for extracting SMF records to include the new SMF data.

For channel initiator statistics data, the SMF record type is 115 and sub-type 231.

For channel initiator accounting data, the SMF record type is 116 and sub-type 10.

You can write your own programs to process this data, or use the SupportPac MP1B that contains a program, MQSMF, for printing the data, and creating data in Comma Separated Values (CSV) format suitable for importing into a spread sheet.

If you are experiencing issues with capturing channel initiator SMF data, see Dealing with issues when capturing SMF data for the channel initiator (CHINIT) for further information.

Related tasks
Interpreting IBM MQ performance statistics
Troubleshooting channel accounting data

z/OS Planning your z/OS TCP/IP environment

To get the best throughput through your network, you must use TCP/IP send and receive buffers with a size of 64 KB, or greater. With this size, the system optimizes its buffer sizes.

See What is Dynamic Right Sizing for High Latency Networks? for more information.

You can check your system buffer size by using the following Netstat command, for example:

```
TSO NETSTAT ALL (CLIENT csq1CHIN
```

The results display much information, including the following two values:

```
ReceiveBufferSize: 0000065536
SendBufferSize: 0000065536
```

65536 is 64 KB. If your buffer sizes are less than 65536, you must work with your network team to increase the **TCPSENDBFRSIZE** and **TCPRCVBUFRSIZE** values in the PROFILE DDName in the TCPIP procedure. For example, you might use the following command:

```
TCPCONFIG TCPSENDBFRSZE 65536 TCPRCVBUFRSIZE 65536
```

If you are unable to change your system-wide **TCPSENDBFRSIZE** or **TCPRCVBUFRSIZE** settings, contact your IBM Software Support center.

z/OS Planning your queue sharing group (QSG)

The easiest way to implement a shared queuing environment, is to configure a queue manager, add that queue manager to a QSG, then add other queue managers to the QSG.

A queue sharing group uses Db2 tables to store configuration information. There is one set of tables used by all QSGs that share the same Db2 data sharing group.

Shared queue messages are stored in a structure in a coupling facility (CF). Each QSG has its own set of CF structures. You need to configure the structures to meet your needs.

Messages over 63KB in size cannot be stored in the CF. You need to use either Shared Message Data Sets (SMDS) or Db2 for these messages.

Message profiles and capacity planning

You should understand the message profile of your shared queue messages. The following are examples of factors that you need to consider:

- Average, and maximum message size
- The typical queue depth, and exception queue depth. For example, you might need to have enough capacity to hold messages for a whole day, and the typical queue depth is under 100 messages.

If the message profile changes, you can increase the size of the structures, or implement SMDS, at a later date.

If you want to be able to handle a large peak volume of messages, you can configure IBM MQ to offload messages to SMDS when the usage of the structure reaches user specified thresholds.

You need to decide if you want to duplex the CF structures. This is controlled by the CF structure definition in the CFRM policy:

1. A duplexed structure uses two coupling facilities. If there is a problem with one CF, there is no interruption to the service, and the structure can be rebuilt on a third CF, if one is available. Duplexed structures can significantly impact the performance of operations on shared queues.
2. If the structure is not duplexed, then a problem with the CF means that shared queues on structures in that CF will become unavailable until the structure can be rebuilt in another CF.

   IBM MQ can be configured to automatically rebuild structures in another CF in this case. Persistent messages will be recovered from the logs of the queue managers.

Note that it is easy to change the CF definitions.

You can define a structure so that it can hold nonpersistent messages only, or so that it can hold persistent and nonpersistent messages.

Structures that can hold persistent messages need to be backed up periodically. Back up your CF structures at least every hour to minimize the time needed to recover the structure in the event of a failure. The backup is stored in the log data set of the queue manager performing the backup.

If you are expecting to have a high throughput of messages on your shared queues, it is best practice to have a dedicated queue manager for backing up the CF structures. This reduces the time needed to recover the structures, as a less data needs to be read from queue manager logs.

Channels

To provide a single system image for applications connecting into an IBM MQ QSG, you can define shared input channels. If these are set up, then a connection coming into the queue sharing group environment, can go to any queue manager in the QSG.

You might need to set up a network router, or Virtual IP address (VIPA) for these channels.

You can define shared output channels. A shared output channel instance can be started from any queue manager in the QSG.

See Shared channels for more information.

Security

You protect IBM MQ resources using an external security manager. If you are using RACF®, the RACF profiles are prefixed with the queue manager name. For example, a queue named APPLICATION.INPUT would be protected using a profile in the MQQUEUE class named `qmgrName.APPLICATION.INPUT.`

When using a queue sharing group you can continue to protect resources with profiles prefixed with the queue manager name, or you can prefix profiles with the queue sharing group name. For example `qsgName.APPLICATION.INPUT`.

You should aim to use profiles prefix with the queue sharing group name because this means there is a single definition for all queue managers, saving you work, and preventing a mismatch in definitions between queue managers.

Related concepts

"Planning for your queue manager" on page 148
When you are setting up a queue manager, your planning should allow for the queue manager to grow, so that the queue manager meets the needs of your enterprise.

z/OS Planning your coupling facility and offload storage environment

Use this topic when planning the initial sizes, and formats of your coupling facility (CF) structures, and shared message data set (SMDS) environment or Db2 environment.

This section contains information about the following topics:

- "Defining coupling facility resources" on page 182
  - Deciding your offload storage mechanism
  - Planning your structures
  - Planning the size of your structures
  - Mapping shared queues to structures
- "Planning your shared message data set (SMDS) environment" on page 188
- "Planning your Db2 environment" on page 191

Defining coupling facility resources

If you intend to use shared queues, you must define the coupling facility structures that IBM MQ will use in your CFRM policy. To do this you must first update your CFRM policy with information about the structures, and then activate the policy.

Your installation probably has an existing CFRM policy that describes the coupling facilities available. The Administrative data utility is used to modify the contents of the policy based on textual statements you provide. You must add statements to the policy that defines the names of the new structures, the coupling facilities that they are defined in, and what size the structures are.

The CFRM policy also determines whether IBM MQ structures are duplexed and how they are reallocated in failure scenarios. Shared queue recovery contains recommendations for configuring CFRM for resilience to failures that affect the coupling facility.

Deciding your offload storage environment

The message data for shared queues can be offloaded from the coupling facility and stored in either a Db2 table or in an IBM MQ managed data set called a *shared message data set* (SMDS). Messages which are too large to store in the coupling facility (that is, larger than 63 KB) must always be offloaded, and smaller messages can optionally be offloaded to reduce coupling facility space usage.

For more information, see Specifying offload options for shared messages.

Planning your structures

A queue sharing group (QSG) requires a minimum of two structures to be defined. The first structure, known as the administrative structure, is used to coordinate IBM MQ internal activity across the queue sharing group. No user data is held in this structure. It has a fixed name of *qsg-name*CSQ_ADMIN (where *qsg-name* is the name of your queue sharing group). Subsequent structures are known as application structures, and are used to hold the messages on IBM MQ shared queues. Each structure can hold up to 512 shared queues.

An application structure named *qsg-name*CSQSYSAPPL is used for system queues. Defining this structure is optional, but it is required in order to use certain features. By default, the SYSTEM.QSG.CHANNEL.SYNCQ and SYSTEM.QSG.UR.RESOLUTION.QUEUE queues are defined on the *qsg-name*CSQSYAPPL structure.

Using multiple structures

A queue sharing group can connect to up to 64 coupling facility structures. One of these structures must be the administration structure. If it is defined, another of these structures might be the *qsg-name*CSQSYSAPPL structure. You can use up to 63 (62 if *qsg-name*CSQSYSAPPL is defined) structures for message data. You might choose to use multiple application structures for any of the following reasons:

- You have some queues that are likely to hold a large number of messages and so require all the resources of an entire coupling facility.
- You have a requirement for a large number of shared queues, so they must be split across multiple structures because each structure can contain only 512 queues.
- RMF reports on the usage characteristic of a structure suggest that you should distribute the queues it contains across a number of coupling facilities.
- You want some queue data to held in a physically different coupling facility from other queue data for data isolation reasons.
- Recovery of persistent shared messages is performed using structure level attributes and commands, for example BACKUP CFSTRUCT. To simplify backup and recovery, you could assign queues that hold nonpersistent messages to different structures from those structures that hold persistent messages.

When choosing which coupling facilities to allocate the structures in, consider the following points:

- Your data isolation requirements.
- The volatility of the coupling facility (that is, its ability to preserve data through a power outage).
- Failure independence between the accessing systems and the coupling facility, or between coupling facilities.
- The level of coupling facility control code (CFCC) installed on the coupling facility (IBM MQ requires Level 9 or higher).

Planning the size of your structures

The administrative structure

The administrative structure (*qsg-name*CSQ_ADMIN) must be large enough to contain 1000 list entries for each queue manager in the queue sharing group. When a queue manager starts, the structure is checked to see if it is large enough for the number of queue managers currently *defined* to the queue sharing group. Queue managers are considered as being defined to the queue sharing group if they have been added by the CSQ5PQSG utility. You can check which queue managers are defined to the group with the MQSC DISPLAY GROUP command.

Note: When calculating the size of the structure, you should allow for the size of large units of work, in addition to the number of queue managers in the queue sharing group.

Table 22 on page 184 shows the minimum required size for the administrative structure for various numbers of queue managers defined in the queue sharing group. These sizes were established for a CFCC level 14 coupling facility structure; for higher levels of CFCC, they probably need to be larger.

Table 22. Minimum administrative structure sizes

Number of queue managers defined in queue sharing group	Required storage
1	6144 KB
2	6912 KB
3	7976 KB
4	8704 KB
5	9728 KB
6	10496 KB
7	11520 KB
8	12288 KB
9	13056 KB
10	14080 KB
11	14848 KB
12	15616 KB
13	16640 KB
14	17408 KB
15	18176 KB
16	19200 KB
17	19968 KB
18	20736 KB
19	21760 KB
20	22528 KB
21	23296 KB
22	24320 KB
23	25088 KB

Table 22. Minimum administrative structure sizes (continued)	
Number of queue managers defined in queue sharing group	**Required storage**
24	25856 KB
25	27136 KB
26	27904 KB
27	28672 KB
28	29696 KB
29	30464 KB
30	31232 KB
31	32256 KB

When you add a queue manager to an existing queue sharing group, the storage requirement might have increased beyond the size recommended in Table 22 on page 184. If so, use the following procedure to estimate the required storage for the *qsg-name*CSQ_ADMIN structure:

1. Issue MQSC command **DISPLAY CFSTATUS(CSQ_ADMIN)** on an existing member of the queue sharing group.
2. Extract the ENTSMAX information for the CSQ_ADMIN structure.
3. If this number is less than 1000 times the total number of queue managers you want to define in the queue sharing group, increase the structure size.

Application structures

The size of the application structures required to hold IBM MQ messages depends on the likely number and size of the messages to be held on a structure concurrently.

The graph in Figure 42 on page 186 shows how large you should make your CF structures to hold the messages on your shared queues. To calculate the allocation size you need the following information:

- The average size of messages on your queues.
- The total number of messages likely to be stored in the structure.

Find the number of messages along the horizontal axis. Select the curve that corresponds to your message size and determine the required value from the vertical axis. For example, for 200 000 messages of length 1 KB gives a value in the range 256 through 512 MB.

Table 23 on page 186 provides the same information in tabular form.

Figure 42. Calculating the size of a coupling facility structure

Use this table to help calculate how large to make your coupling facility structures:

Table 23. Calculating the size of a coupling facility structure

Number of messages	1 KB	2 KB	4 KB	8 KB	16 KB	32 KB	63 KB
100	6 MB	6 MB	7 MB	7 MB	8 MB	10 MB	14 MB
1000	8 MB	9 MB	12 MB	17 MB	27 MB	48 MB	88 MB
10000	25 MB	38 MB	64 MB	115 MB	218 MB	423 MB	821 MB
100000	199 MB	327 MB	584 MB	1097 MB	2124 MB	4177 MB	8156 MB

Your CFRM policy should include the following statements:

- `INITSIZE` is the size in KB that the structure is allocated with when the first queue manager connects to it.
- `SIZE` is the maximum size that the structure can attain.
- `FULLTHRESHOLD` sets the percentage value of the threshold at which z/OS issues message IXC585E to indicate that the structure is getting full.

A best practice is to ensure that INITSIZE and SIZE are within a factor of 2. For example, with the figures determined previously, you might include the following statements:

```
STRUCTURE NAME(structure-name)
INITSIZE(value from graph in KB, that is, multiplied by 1024)
SIZE(something larger)
FULLTHRESHOLD(85)
```

```
STRUCTURE NAME(QSG1APPLICATION1)
INITSIZE(262144) /* 256 MB */
SIZE(524288)     /* 512 MB */
FULLTHRESHOLD(85)
```

If the structure use reaches the threshold where warning messages are issued, intervention is required. You might use IBM MQ to inhibit MQPUT operations to some of the queues in the structure to prevent applications from writing more messages, start more applications to get messages from the queues, or quiesce some of the applications that are putting messages to the queue.

Alternatively, you can use z/OS facilities to alter the structure size in place. The following z/OS command:

```
SETXCF START,ALTER,STRNAME=structure-name,SIZE=newsize
```

alters the size of the structure to *newsize*, where *newsize* is a value that is less than the value of SIZE specified on the CFRM policy for the structure, but greater than the current coupling facility size.

You can monitor the use of a coupling facility structure with the MQSC DISPLAY CFSTATUS command.

If no action is taken and a queue structure fills up, an MQRC_STORAGE_MEDIUM_FULL return code is returned to the application. If the administration structure becomes full, the exact symptoms depend on which processes experience the error, but they might include the following problems:

- No responses to commands.
- Queue manager failure as a result of problems during commit processing.

The CSQSYSAPPL structure

The *qsg-name*CSQSYSAPPL structure is an application structure for system queues. Table 3 demonstrates an example of how to estimate the message data sizes for the default queues defined on the *qsg-name*CSQSYSAPPL structure.

Table 24. Table showing CSQSYSAPPL usage against sizing.

***qsg-name*CSQSYSAPPL usage**	**Sizing**
SYSTEM.QSG.CHANNEL.SYNCQ	2 messages of 500 bytes per active instance of a shared channel
SYSTEM.QSG.UR.RESOLUTION.QUEUE	1000 messages of 2 KB

The suggested initial structure definition values are as follows:

```
STRUCTURE NAME(qsg-nameCSQSYSAPPL)
INITSIZE(20480)        /* 20 MB */
SIZE(30720)            /* 30 MB */
FULLTHRESHOLD(85)
```

These values can be adjusted depending on your use of shared channels and group units of recovery.

Mapping shared queues to structures

To define an application structure to IBM MQ, use the DEFINE CFSTRUCT command. When you define a structure to IBM MQ, do not include the QSG name prefix in the structure name. For example, to define an application structure to IBM MQ that has the name *qsg-name*APPLICATION1 in the CFRM policy, issue the following command:

```
DEFINE CFSTRUCT(APPLICATION1)
```

The CFSTRUCT attribute of the queue definition is used to map the queue to a structure. Specify the name of the CF structure without the QSG name prefix in this attribute. For example, the following command defines a shared queue on the APPLICATION1 structure:

```
DEFINE QLOCAL(myqueue) QSGDISP(SHARED) CFSTRUCT(APPLICATION1)
```

z/OS *Planning your shared message data set (SMDS) environment*

If you are using queue sharing groups with SMDS offloading, IBM MQ needs to connect to a group of shared message data sets. Use this topic to help understand the data set requirements, and configuration required to store IBM MQ message data.

A *shared message data set* (described by the keyword SMDS) is a data set used by a queue manager to store offloaded message data for shared messages stored in a coupling facility structure.

Note: When defining SMDS data sets for a structure, you must have one for each queue manager.

When this form of data offloading is enabled, the **CFSTRUCT** requires an associated group of shared message data sets, one data set for each queue manager in the queue sharing group. The group of shared message data sets is defined to IBM MQ using the **DSGROUP** parameter on the **CFSTRUCT** definition. Additional parameters can be used to supply further optional information, such as the number of buffers to use and expansion attributes for the data sets.

Each queue manager can write to the data set which it owns, to store shared message data for messages written through that queue manager, and can read all of the data sets in the group.

A list describing the status and attributes for each data set associated with the structure is maintained internally as part of the **CFSTRUCT** definition, so each queue manager can check the definition to find out which data sets are currently available.

This data set information can be displayed using the **DISPLAY CFSTATUS TYPE(SMDS)** command to display current status and availability, and the **DISPLAY SMDS** command to display the parameter settings for the data sets associated with a specified **CFSTRUCT**.

Individual shared message data sets are effectively identified by the combination of the owning queue manager name (usually specified using the **SMDS** keyword) and the **CFSTRUCT** structure name.

This section describes the following topics:

- The DSGROUP parameter
- The DSBLOCK parameter
- Shared message data set characteristics
- Shared message data set space management
- Access to shared message data sets
- Creating a shared message data set
- Shared message data set performance and capacity considerations
- Activating a shared message data set

See DEFINE CFSTRUCT for details of these parameters.

For information on managing your shared message data sets, see Managing shared message data sets for further details.

The DSGROUP parameter

The **DSGROUP** parameter on the **CFSTRUCT** definition identifies the group of data sets in which large messages for that structure are to be stored. Additional parameters may be used to specify the logical block size to be used for space allocation purposes and values for the buffer pool size and automatic data set expansion options.

The **DSGROUP** parameter must be set up before offloading to data sets can be enabled.

- If a new **CFSTRUCT** is being defined at **CFLEVEL(5)** and the option **OFFLOAD(SMDS)** is specified or assumed, then the **DSGROUP** parameter must be specified on the same command.
- If an existing **CFSTRUCT** is being altered to increase the **CFLEVEL** to **CFLEVEL(5)** and the option **OFFLOAD(SMDS)** is specified or assumed, then the **DSGROUP** parameter must be specified on the same command if it is not already set.

The DSBLOCK parameter

Space within each data set is allocated to queues as logical blocks of a fixed size (usually 256 KB) specified using the **DSBLOCK** parameter on the **CFSTRUCT** definition, then allocated to individual messages as ranges of pages of 4 KB (corresponding to the physical block size and control interval size) within each logical block. The logical block size also determines the maximum amount of message data that can be read or written in a single I/O operation, which is the same as the buffer size for the SMDS buffer pool.

A larger value of the **DSBLOCK** parameter can improve performance for very large messages by reducing the number of separate I/O operations. However, a smaller value decreases the amount of buffer storage required for each active request. The default value for the **DSBLOCK** parameter is 256 KB, which provides a reasonable balance between these requirements, so specifying this parameter might not normally be necessary.

Shared message data set characteristics

A shared message data set is defined as a VSAM linear data set (LDS). Each offloaded message is stored in one or more blocks in the data set. The stored data is addressed directly by information in the coupling facility entries, like an extended form of virtual storage. There is no separate index or similar control information stored in the data set itself.

The direct addressing scheme means that for messages which fit into one block, only a single I/O operation is needed to read or write the block. When a message spans more than one block, the I/O operations for each block can be fully overlapped to minimize elapsed time, provided that sufficient buffers are available.

The shared message data set also contains a small amount of general control information, consisting of a header in the first page, which includes recovery and restart status information, and a space map checkpoint area which is used to save the free block space map at queue manager normal termination.

Shared message data set space management

As background information for capacity, performance and operational considerations, it might be useful to understand the concepts of how space in shared message data sets is managed by the queue managers.

Free space in each shared message data set is tracked by its owning queue manager using a space map which indicates the number of pages in use within each logical block. The space map is maintained in main storage while the data set is open and saved in the data set when it is closed normally. (In recovery situations the space map is automatically rebuilt by scanning the messages in the coupling facility structure to find out which data set pages are currently in use).

When a shared message with offloaded message data is being written, the queue manager allocates a range of pages for each message block. If there is a partly used current logical block for the specified queue, the queue manager allocates space starting at the next free page in that block, otherwise it allocates a new logical block. If the whole message does not fit within the current logical block, the queue

manager splits the message data at the end of the logical block and allocates a new logical block for the next message block. This is repeated until space has been allocated for the whole message. Any unused space in the last logical block is saved as the new current logical block for the queue. When the data set is closed normally, any unused pages in current logical blocks are returned to the space map before it is saved.

When a shared message with offloaded message data has been read and is ready to be deleted, the queue manager processes the delete request by transferring the coupling facility entry for the message to a clean-up list monitored by the owning queue manager (which may be the same queue manager). When entries arrive on this list, the owning queue manager reads and deletes the entries and returns the freed ranges of pages to the space map. When all used pages in a logical block have been freed the block becomes available for reuse.

Access to shared message data sets

Each shared message data set must be on shared direct access storage which is accessible to all queue managers in the queue sharing group.

During normal running, each queue manager opens its own shared message data set for read/write access, and opens any active shared message data sets for other queue managers for read-only access, so it can read messages stored by those queue managers. This means that each queue manager userid requires at least UPDATE access to its own shared message data set and READ access to all other shared message data sets for the structure.

If it is necessary to recover shared message data sets using **RECOVER CFSTRUCT**, the recovery process can be executed from any queue manager in the queue sharing group. A queue manager which may be used to perform recovery processing requires UPDATE access to all data sets that it may need to recover

Creating a shared message data set

Each shared message data set should normally be created before the corresponding **CFSTRUCT** definition is created or altered to enable the use of this form of message offloading, as the **CFSTRUCT** definition changes will normally take effect immediately, and the data set will be required as soon as a queue manager attempts to access a shared queue which has been assigned to that structure. A sample job to allocate and pre-format a shared message data set is provided in SCSQPROC(CSQ4SMDS). The job must be customized and run to allocate a shared message data set for each queue manager which uses a CFSTRUCT with OFFLOAD(SMDS).

If the queue manager finds that offload support has been enabled and tries to open its shared message data set but it has not yet been created, the shared message data set will be flagged as unavailable. The queue manager will then be unable to store any large messages until the data set has been created and the queue manager has been notified to try again, for example using the **START SMDSCONN** command.

A shared message data set is created as a VSAM linear data set using an Access Method Services **DEFINE CLUSTER** command. The definition must specify **SHAREOPTIONS(2 3)** to allow one queue manager to open it for write access and any number of queue managers to read it at the same time. The default control interval size of 4 KB must be used. If the data set may need to expand beyond 4 GB, it must be defined using an SMS data class which has the VSAM extended addressability attribute. A shared message data set is eligible to reside in the extended addressing space (EAS) part of an extended address volumes (EAV).

Each shared message data set can either be empty or pre-formatted to binary zeros (using **CSQJUFMT** or a similar utility such as the sample job SCSQPROC(CSQ4SMDS)), before its initial use. If it is empty or only partly formatted when it is opened, the queue manager automatically formats the remaining space to binary zeros.

Shared message data set performance and capacity considerations

Each shared message data set is used to store offloaded data for shared messages written to the associated **CFSTRUCT** by the owning queue manager, from regions within the same system. Each message that is offloaded takes up to 768 bytes of CF storage, made up of 256 bytes for the entry

and 512 bytes for the two elements of header and descriptor. Each offloaded message is stored in one or more pages (physical blocks of size 4 KB) in the data set.

The data set space required for a given number of offloaded messages can therefore be estimated by rounding up the overall message size (including the descriptor) to the next multiple of 4 KB and then multiplying by the number of messages.

As for a page set, when a shared message data set is almost full, it can optionally be expanded automatically. The default behavior for this automatic expansion can be set using the **DSEXPAND** parameter on the **CFSTRUCT** definition. This setting can be overridden for each queue manager using the **DSEXPAND** parameter on the **ALTER SMDS** command. Automatic expansion is triggered when the data set reaches 90% full and more space is required. If expansion is allowed but an expansion attempt is rejected by VSAM because no secondary space allocation was specified when the data set was defined, expansion is retried using a secondary allocation of 20% of the current size of the data set.

Provided that the shared message data set is defined with the extended addressability attribute, the maximum size is only limited by VSAM considerations to a maximum of 16 TB or 59 volumes. This is significantly larger than the 64 GB maximum size of a local page set.

Activating a shared message data set

When a queue manager has successfully connected to an application coupling facility structure, it checks whether that structure definition specifies offloading using an associated **DSGROUP** parameter. If so, the queue manager allocates and opens its own shared message data set for write access, then it opens for read access any existing shared message data sets owned by other queue managers.

When a shared message data set is opened for the first time (before it has been recorded as active within the queue sharing group), the first page will not yet contain a valid header. The queue manager fills in header information to identify the queue sharing group, the structure name and the owning queue manager.

After the header has been completed, the queue manager registers the new shared message data set as active and broadcasts an event to notify any other active queue managers about the new data set.

Every time a queue manager opens a shared message data set it validates the header information to ensure that the correct data set is still being used and that it has not been damaged.

z/OS *Planning your Db2 environment*

If you are using queue sharing groups, IBM MQ needs to attach to a Db2 subsystem that is a member of a data sharing group. Use this topic to help understand the Db2 requirements used to hold IBM MQ data.

IBM MQ needs to know the name of the data sharing group that it is to connect to, and the name of a Db2 subsystem (or Db2 group) to connect to, to reach this data sharing group. These names are specified in the QSGDATA parameter of the CSQ6SYSP system parameter macro (described in Using CSQ6SYSP).

Within the data sharing group, shared Db2 tables are used to hold:

- Configuration information for the queue sharing group.
- Properties of IBM MQ shared and group objects.
- Optionally, data relating to offloaded IBM MQ messages.

IBM MQ provides a single set of sample jobs for defining the necessary Db2 table spaces, tables, and indexes. These jobs make use of Universal Table Spaces (UTS). Earlier versions of the product had two sets of jobs, one for UTS, and one for older types of table space, which have been deprecated by the most recent versions of Db2.

IBM MQ can still be used with older types of table space, and this might be appropriate if you already have an existing queue sharing group. However, if you are creating a new queue sharing group, it should use UTS.

Db2 V12 Function level 508 provides a non disruptive migration process for migrating multi-table table spaces to universal table spaces. You can use this approach to migrate the multi-table table spaces, used

by existing queue sharing groups, to universal table spaces without taking an outage of the whole queue sharing group.

In Db2 V13, use the MOVE TABLE option of the ALTER TABLESPACE statement. See Moving tables from multi-table table spaces to partition-by-growth table spaces for more information.

By default Db2 uses the user ID of the person running the jobs as the owner of the Db2 resources. If this user ID is deleted then the resources associated with it are deleted, and so the table is deleted. Consider using a group ID to own the tables, rather than an individual user ID. You can do this by adding `GROUP=groupname` onto the JOB card, and specifying `SET CURRENT SQLID='groupname'` before any SQL statements.

IBM MQ uses the RRS Attach facility of Db2. This means that you can specify the name of a Db2 group that you want to connect to. The advantage of connecting to a Db2 group attach name (rather than a specific Db2 subsystem), is that IBM MQ can connect (or reconnect) to any available Db2 subsystem on the z/OS image that is a member of that group. There must be a Db2 subsystem that is a member of the data sharing group active on each z/OS image where you are going to run a queue-sharing IBM MQ subsystem, and RRS must be active.

Db2 storage

For most installations, the amount of Db2 storage required is about 20 or 30 cylinders on a 3390 device. However, if you want to calculate your storage requirement, the following table gives some information to help you determine how much storage Db2 requires for the IBM MQ data. The table describes the length of each Db2 row, and when each row is added to or deleted from the relevant Db2 table. Use this information together with the information about calculating the space requirements for the Db2 tables and their indexes in the *Db2 for z/OS Installation Guide*.

Table 25. Planning your Db2 storage requirements

Db2 table name	Length of row	A row is added when:	A row is deleted when:
CSQ.ADMIN_B_QSG	252 bytes	A queue sharing group is added to the table with the ADD QSG function of the CSQ5PQSG utility.	A queue sharing group is removed from the table with the REMOVE QSG function of the CSQ5PQSG utility. (All rows relating to this queue sharing group are deleted automatically from all the other Db2 tables when the queue sharing group record is deleted.)
CSQ.ADMIN_B_QMGR	Up to 3828 bytes	A queue manager is added to the table with the ADD QMGR function of the CSQ5PQSG utility.	A queue manager is removed from the table with the REMOVE QMGR function of the CSQ5PQSG utility.
CSQ.ADMIN_B_STRUCTURE	1454 bytes	The first local queue definition, specifying the QSGDISP(SHARED) attribute, that names a previously unknown structure within the queue sharing group is defined.	The last local queue definition, specifying the QSGDISP(SHARED) attribute, that names a structure within the queue sharing group is deleted.
CSQ.ADMIN_B_SCST	342 bytes	A shared channel is started.	A shared channel becomes inactive.
CSQ.ADMIN_B_SSKT	254 bytes	A shared channel that has the NPMSPEED(NORMAL) attribute is started.	A shared channel that has the NPMSPEED(NORMAL) attribute becomes inactive.

Table 25. Planning your Db2 storage requirements (continued)

Db2 table name	Length of row	A row is added when:	A row is deleted when:
CSQ.ADMIN_B_STRBACKUP	514 bytes	A new row is added to the CSQ.ADMIN_B_STRUCTURE table. Each entry is a dummy entry until the BACKUP CFSTRUCT command is run, which overwrites the dummy entries.	A row is deleted from the CSQ.ADMIN_B_STRUCTURE table.
CSQ.OBJ_B_AUTHINFO	3400 bytes	An authentication information object with QSGDISP(GROUP) is defined.	An authentication information object with QSGDISP(GROUP) is deleted.
CSQ.OBJ_B_QUEUE	Up to 3707 bytes	• A queue with the QSGDISP(GROUP) attribute is defined. • A queue with the QSGDISP(SHARED) attribute is defined. • A model queue with the DEFTYPE(SHAREDYN) attribute is opened.	• A queue with the QSGDISP(GROUP) attribute is deleted. • A queue with the QSGDISP(SHARED) attribute is deleted. • A dynamic queue with the DEFTYPE(SHAREDYN) attribute is closed with the DELETE option.
CSQ.OBJ_B_NAMELIST	Up to 15127 bytes	A namelist with the QSGDISP(GROUP) attribute is defined.	A namelist with the QSGDISP(GROUP) attribute is deleted.
CSQ.OBJ_B_CHANNEL	Up to 14127 bytes	A channel with the QSGDISP(GROUP) attribute is defined.	A channel with the QSGDISP(GROUP) attribute is deleted.
CSQ.OBJ_B_STGCLASS	Up to 2865 bytes	A storage class with the QSGDISP(GROUP) attribute is defined.	A storage class with the QSGDISP(GROUP) attribute class is deleted.
CSQ.OBJ_B_PROCESS	Up to 3347 bytes	A process with the QSGDISP(GROUP) attribute is defined.	A process with the QSGDISP(GROUP) attribute is deleted.
CSQ.OBJ_B_TOPIC	Up to 14520 bytes	A topic object with QSGDISP(GROUP) attribute is defined.	A topic object with QSGDISP(GROUP) attribute is deleted.
CSQ.EXTEND_B_QMGR	Less than 430 bytes	A queue manager is added to the table with the ADD QMGR function of the CSQ5PQSG utility.	A queue manager is removed from the table with the REMOVE QMGR function of the CSQ5PQSG utility.
CSQ.ADMIN_B_MESSAGES	87 bytes	For large message PUT (1 per BLOB).	For large message GET (1 per BLOB).

Table 25. Planning your Db2 storage requirements (continued)			
Db2 table name	**Length of row**	**A row is added when:**	**A row is deleted when:**
CSQ.ADMIN_MSGS_BAUX1 CSQ.ADMIN_MSGS_BAUX2 CSQ.ADMIN_MSGS_BAUX3 CSQ.ADMIN_MSGS_BAUX4		These 4 tables contain message payload for large messages added into one of these 4 tables for each BLOB of the message. BLOBS are up to 511 KB in length, so if the message size is > 711 KB, there will be multiple BLOBs for this message.	

The use of large numbers of shared queue messages of size greater than 63 KB can have significant performance implications on your IBM MQ system. For more information, see SupportPac MP16, Capacity Planning and Tuning for IBM MQ for z/OS, at: SupportPacs for IBM MQ and other project areas.

z/OS Planning for backup and recovery

Developing backup and recovery procedures at your site is vital to avoid costly and time-consuming losses of data. IBM MQ provides means for recovering both queues and messages to their current state after a system failure.

This topic contains the following sections:

- "Recovery procedures" on page 194
- "Tips for backup and recovery" on page 195
- "Recovering page sets" on page 197
- "Recovering CF structures" on page 198
- "Achieving specific recovery targets" on page 198
- "Backup considerations for other products" on page 200
- "Recovery and CICS" on page 200
- "Recovery and IMS" on page 201
- "Preparing for recovery on an alternative site" on page 201
- "Example of queue manager backup activity" on page 201

Recovery procedures

Develop the following procedures for IBM MQ:

- Creating a point of recovery.
- Backing up page sets.
- Backing up CF structures.
- Recovering page sets.
- Recovering from out-of-space conditions (IBM MQ logs and page sets).
- Recovering CF structures.

See Administración IBM MQ for z/OS for information about these.

Become familiar with the procedures used at your site for the following:

- Recovering from a hardware or power failure.
- Recovering from a z/OS component failure.

- Recovering from a site interruption, using off-site recovery.

z/OS Tips for backup and recovery

Use this topic to understand some backup and recovery tasks.

The queue manager restart process recovers your data to a consistent state by applying log information to the page sets. If your page sets are damaged or unavailable, you can resolve the problem using your backup copies of your page sets (if all the logs are available). If your log data sets are damaged or unavailable, it might not be possible to recover completely.

Consider the following points:

- Periodically take backup copies
- Do not discard archive logs you might need
- Do not change the DDname to page set association

Periodically take backup copies

A *point of recovery* is the term used to describe a set of backup copies of IBM MQ page sets and the corresponding log data sets required to recover these page sets. These backup copies provide a potential restart point in the event of page set loss (for example, page set I/O error). If you restart the queue manager using these backup copies, the data in IBM MQ is consistent up to the point that these copies were taken. Provided that all logs are available from this point, IBM MQ can be recovered to the point of failure.

The more recent your backup copies, the quicker IBM MQ can recover the data in the page sets. The recovery of the page sets is dependent on all the necessary log data sets being available.

In planning for recovery, you need to determine how often to take backup copies and how many complete backup cycles to keep. These values tell you how long you must keep your log data sets and backup copies of page sets for IBM MQ recovery.

When deciding how often to take backup copies, consider the time needed to recover a page set. The time needed is determined by the following:

- The amount of log to traverse.
- The time it takes an operator to mount and remove archive tape volumes.
- The time it takes to read the part of the log needed for recovery.
- The time needed to reprocess changed pages.
- The storage medium used for the backup copies.
- The method used to make and restore backup copies.

In general, the more frequently you make backup copies, the less time recovery takes, but the more time is spent making copies.

For each queue manager, you should take backup copies of the following:

- The archive log data sets
- The BSDS copies created at the time of the archive
- The page sets
- Your object definitions
- Your CF structures

To reduce the risk of your backup copies being lost or damaged, consider:

- Storing the backup copies on different storage volumes to the original copies.
- Storing the backup copies at a different site to the original copies.

- Making at least two copies of each backup of your page sets and, if you are using single logging or a single BSDS, two copies of your archive logs and BSDS. If you are using dual logging or BSDS, make a single copy of both archive logs or BSDS.

Before moving IBM MQ to a production environment, fully test and document your backup procedures.

Backing up your page sets

You need to back up page sets regularly. Some enterprises back up the page sets twice a day.

You need the active and archive logs since a backup to be able to recover using the backup. You need enough log data to go back four checkpoints if the backup was taken when the queue manager was running.

You can use ADRDSSU FastReplication to back up page sets, and you can do this while the queue manager is active. Note that you need to ensure there is enough space in the storage pool.

Backing up your object definitions

Create backup copies of your object definitions. To do this, use the MAKEDEF feature of the COMMAND function of the utility program (described in Using the COMMAND function of CSQUTIL).

You should do this whenever you take backup copies of your queue manager data sets, and keep the most current version.

Backing up your coupling facility structures

If you have set up any queue sharing groups, even if you are not using them, you must take periodic backups of your CF structures. To do this, use the IBM MQ BACKUP CFSTRUCT command. You can use this command only on CF structures that are defined with the RECOVER(YES) attribute. If any CF entries for persistent shared messages refer to offloaded message data stored in a shared message data set (SMDS) or Db2, the offloaded data is retrieved and backed up together with the CF entries. Shared message data sets should not be backed up separately.

It is recommended that you take a backup of all your CF structures about every hour, to minimize the time it takes to restore a CF structure.

You could perform all your CF structure backups on a single queue manager, which has the advantage of limiting the increase in log use to a single queue manager. Alternatively, you could perform backups on all the queue managers in the queue sharing group, which has the advantage of spreading the workload across the queue sharing group. Whichever strategy you use, IBM MQ can locate the backup and perform a RECOVER CFSTRUCT from any queue manager in the queue sharing group. The logs of all the queue managers in the queue sharing group need to be accessed to recover the CF structure.

Backing up your message security policies

If you are using Advanced Message Security to create a backup of your message security policies, create a backup using the message security policy utility (CSQ0UTIL) to run **dspmqspl** with the -export parameter, then save the policy definitions that are output to the EXPORT DD.

You should create a backup of your message security policies whenever you take backup copies of your queue manager data sets, and keep the most current version.

Do not discard archive logs you might need

IBM MQ might need to use archive logs during restart. You must keep sufficient archive logs so that the system can be fully restored. IBM MQ might use an archive log to recover a page set from a restored backup copy. If you have discarded that archive log, IBM MQ cannot restore the page set to its current state. When and how you discard archive logs is described in Discarding archive log data sets.

You can use the `/cpf DIS USAGE TYPE(ALL)` command to display the log RBA, and log range sequence number (LRSN) that you need to recover your queue manager's page sets and the queue sharing group's structures. You should then use the print log map utility (CSQJU004) to print bootstrap data set (BSDS) information for the queue manager to locate the logs containing the log RBA.

For CF structures, you need to run the CSQJU004 utility on each queue manager in the queue sharing group to locate the logs containing the LRSN. You need these logs and any later logs to be able to recover the page sets and structures.

Do not change the DDname to page set association

IBM MQ associates page set number 00 with DDname CSQP0000, page set number 01 with DDname CSQP0001, and so on, up to CSQP0099. IBM MQ writes recovery log records for a page set based on the DDname that the page set is associated with. For this reason, you must not move page sets that have already been associated with a PSID DDname.

z/OS Recovering page sets

Use this topic to understand the factors involved when recovering pages sets, and how to minimize restart times.

A key factor in recovery strategy concerns the time for which you can tolerate a queue manager outage. The total outage time might include the time taken to recover a page set from a backup, or to restart the queue manager after an abnormal termination. Factors affecting restart time include how frequently you back up your page sets, and how much data is written to the log between checkpoints.

To minimize the restart time after an abnormal termination, keep units of work short so that, at most, two active logs are used when the system restarts. For example, if you are designing an IBM MQ application, avoid placing an MQGET call that has a long wait interval between the first in-syncpoint MQI call and the commit point because this might result in a unit of work that has a long duration. Another common cause of long units of work is batch intervals of more than 5 minutes for the channel initiator.

You can use the DISPLAY THREAD command to display the RBA of units of work and to help resolve the old ones.

How often must you back up a page set?

Frequent page set backup is essential if a reasonably short recovery time is required. This applies even when a page set is very small or there is a small amount of activity on queues in that page set.

If you use persistent messages in a page set, the backup frequency should be in hours rather than days. This is also the case for page set zero.

To calculate an approximate backup frequency, start by determining the target total recovery time. This consists of the following:

1. The time taken to react to the problem.
2. The time taken to restore the page set backup copy.

   If you use SnapShot backup/restore, the time taken to perform this task is a few seconds. For information about SnapShot, see the *DFSMSdss Storage Administration Guide*.
3. The time the queue manager requires to restart, including the additional time needed to recover the page set.

   This depends most significantly on the amount of log data that must be read from active and archive logs since that page set was last backed up. All such log data must be read, in addition to that directly associated with the damaged page set.

   Note: When using *fuzzy backup* (where a snapshot is taken of the logs and page sets while a unit of work is active), it might be necessary to read up to three additional checkpoints, and this might result in the need to read one or more additional logs.

When deciding on how long to allow for the recovery of the page set, the factors that you need to consider are:

- The rate at which data is written to the active logs during normal processing depends on how messages arrive in your system, in addition to the message rate.

  Messages received or sent over a channel result in more data logging than messages generated and retrieved locally.
- The rate at which data can be read from the archive and active logs.

  When reading the logs, the achievable data rate depends on the devices used and the total load on your particular DASD subsystem.

  With most tape units, it is possible to achieve higher data rates for archived logs with a large block size. However, if an archive log is required for recovery, all the data on the active logs must be read also.

z/OS Recovering CF structures

Use this topic to understand the recovery process for CF structures.

At least one queue manager in the queue sharing group must be active to process a RECOVER CFSTRUCT command. CF structure recovery does not affect queue manager restart time, because recovery is performed by an already active queue manager.

The recovery process consists of two logical steps that are managed by the RECOVER CFSTRUCT command:

1. Locating and restoring the backup.
2. Merging all the logged updates to persistent messages that are held on the CF structure from the logs of all the queue managers in the queue sharing group that have used the CF structure, and applying the changes to the backup.

The second step is likely to take much longer because a lot of log data might need to be read. You can reduce the time taken if you take frequent backups, or if you recover multiple CF structures at the same time, or both.

The queue manager performing the recovery locates the relevant backups on all the other queue managers' logs using the data in Db2 and the bootstrap data sets. The queue manager replays these backups in the correct time sequence across the queue sharing group, from just before the last backup through to the point of failure.

The time it takes to recover a CF structure depends on the amount of recovery log data that must be replayed, which in turn depends on the frequency of the backups. In the worst case, it takes as long to read a queue manager's log as it did to write it. So if, for example, you have a queue sharing group containing six queue managers, an hour's worth of log activity could take six hours to replay. In general it takes less time than this, because reading can be done in bulk, and because the different queue manager's logs can be read in parallel. As a starting point, we recommend that you back up your CF structures every hour.

All queue managers can continue working with non-shared queues and queues in other CF structures while there is a failed CF structure. If the administration structure has also failed, at least one of the queue managers in the queue sharing group must be started before you can issue the RECOVER CFSTRUCT command.

Backing up CF structures can require considerable log writing capacity, and can therefore impose a large load on the queue manager doing the backup. Choose a lightly loaded queue manager for doing backups; for busy systems, add an additional queue manager to the queue sharing group and dedicate it exclusively for doing backups.

z/OS Achieving specific recovery targets

Use this topic for guidance on how you can achieve specific recovery target times by adjusting backup frequency.

If you have specific recovery targets to achieve, for example, completion of the queue manager recovery and restart processing in addition to the normal startup time within *xx* seconds, you can use the following calculation to estimate your backup frequency (in hours):

```
Formula (A)
                    Required restart time * System recovery log read rate
                          (in secs)                  (in MB/sec)
Backup frequency  = ------------------------------------------------------
   (in hours)              Application log write rate (in MB/hour)
```

Note: The examples given next are intended to highlight the need to back up your page sets frequently. The calculations assume that most log activity is derived from a large number of persistent messages. However, there are situations where the amount of log activity is not easily calculated. For example, in a queue sharing group environment, a unit of work in which shared queues are updated in addition to other resources might result in UOW records being written to the IBM MQ log. For this reason, the Application log write rate in Formula (A) can be derived accurately only from the observed rate at which the IBM MQ logs fill.

For example, consider a system in which IBM MQ MQI clients generate a total load of 100 persistent messages a second. In this case, all messages are generated locally.

If each message is of user length 1 KB, the amount of data logged each hour is approximately:

```
100 * (1 + 1.3) KB * 3600 = approximately 800 MB

where
      100           = the message rate a second
      (1 + 1.3) KB  = the amount of data logged for
                      each 1 KB of persistent messages
```

Consider an overall target recovery time of 75 minutes. If you have allowed 15 minutes to react to the problem and restore the page set backup copy, queue manager recovery and restart must then complete within 60 minutes (3600 seconds) applying formula (A). Assuming that all required log data is on RVA2-T82 DASD, which has a recovery rate of approximately 2.7 MB a second, this necessitates a page set backup frequency of at least every:

```
3600 seconds * 2.7 MB a second / 800 MB an hour = 12.15 hours
```

If your IBM MQ application day lasts approximately 12 hours, one backup each day is appropriate. However, if the application day lasts 24 hours, two backups each day is more appropriate.

Another example might be a production system in which all the messages are for request-reply applications (that is, a persistent message is received on a receiver channel and a persistent reply message is generated and sent down a sender channel).

In this example, the achieved batch size is one, and so there is one batch for every message. If there are 50 request replies a second, the total load is 100 persistent messages a second. If each message is 1 KB in length, the amount of data logged each hour is approximately:

```
50((2 * (1+1.3) KB) + 1.4 KB + 2.5 KB) * 3600 = approximately 1500 MB

where:
 50                 = the message pair rate a second
 (2 * (1 + 1.3) KB) = the amount of data logged for each message pair
 1.4 KB             = the overhead for each batch of messages
                      received by each channel
 2.5 KB             = the overhead for each batch of messages sent
                      by each channel
```

To achieve the queue manager recovery and restart within 30 minutes (1800 seconds), again assuming that all required log data is on RVA2-T82 DASD, this requires that page set backup is carried out at least every:

```
1800 seconds * 2.7 MB a second / 1500 MB an hour = 3.24 hours
```

Periodic review of backup frequency

Monitor your IBM MQ log usage in terms of MB an hour. Periodically perform this check and amend your page set backup frequency if necessary.

z/OS Backup considerations for other products

If you are using IBM MQ with CICS or IMS then you must also consider the implications for your backup strategy with those products. The data facility hierarchical storage manager (DFHSM) manages data storage, and can interact with the storage used by IBM MQ.

Backup and recovery with DFHSM

The data facility hierarchical storage manager (DFHSM) does automatic space-availability and data-availability management among storage devices in your system. If you use it, you need to know that it moves data to and from the IBM MQ storage automatically.

DFHSM manages your DASD space efficiently by moving data sets that have not been used recently to alternative storage. It also makes your data available for recovery by automatically copying new or changed data sets to tape or DASD backup volumes. It can delete data sets, or move them to another device. Its operations occur daily, at a specified time, and allow for keeping a data set for a predetermined period before deleting or moving it.

You can also perform all DFHSM operations manually. For more information on DFHSM, see the z/OS DFSMS product documentation. If you use DFHSM with IBM MQ, note that DFHSM does the following:

- Uses cataloged data sets.
- Operates on page sets and logs.
- Supports VSAM data sets.

Recovery and CICS

The recovery of CICS resources is not affected by the presence of IBM MQ. CICS recognizes IBM MQ as a non-CICS resource (or external resource manager), and includes IBM MQ as a participant in any syncpoint coordination requests using the CICS resource manager interface (RMI). For more information about CICS recovery and the CICS resource manager interface, see the CICS product documentation.

Recovery and IMS

IMS recognizes IBM MQ as an external subsystem and as a participant in syncpoint coordination. IMS recovery for external subsystem resources is described in the IMS product documentation.

z/OS Preparing for recovery on an alternative site

If a total loss of an IBM MQ computing center, you can recover on another IBM MQ system at a recovery site.

To recover an IBM MQ system at a recovery site, you must regularly back up the page sets and the logs. As with all data recovery operations, the objectives of disaster recovery are to lose as little data, workload processing (updates), and time as possible.

At the recovery site:

- The recovery IBM MQ queue manager **must** have the same name as the lost queue manager.
- Ensure the system parameter module used on the recovery queue manager contains the same parameters as the lost queue manager.

See Administering IBM MQ for z/OS and Troubleshooting IBM MQ for z/OS problems for more information.

z/OS Example of queue manager backup activity

This topic shows as an example of queue manager backup activity.

When you plan your queue manager backup strategy, a key consideration is retention of the correct amount of log data. Managing the logs describes how to determine which log data sets are required, by reference to the system recovery RBA of the queue manager. IBM MQ determines the system recovery RBA using information about the following:

- Currently active units of work.
- Page set updates that have not yet been flushed from the buffer pools to disk.
- CF structure backups, and whether this queue manager's log contains information required in any recovery operation using them.

You must retain sufficient log data to be able to perform media recovery. While the system recovery RBA increases over time, the amount of log data that must be retained only decreases when subsequent backups are taken. CF structure backups are managed by IBM MQ, and so are taken into account when reporting the system recovery RBA. This means that in practice, the amount of log data that must be retained only reduces when page set backups are taken.

Figure 43 on page 202 shows an example of the backup activity on a queue manager that is a member of a queue sharing group, how the recovery RBA varies with each backup, and how that affects the amount of log data that must be retained. In the example the queue manager uses local and shared resources: page sets, and two CF structures, STRUCTURE1 and STRUCTURE2.

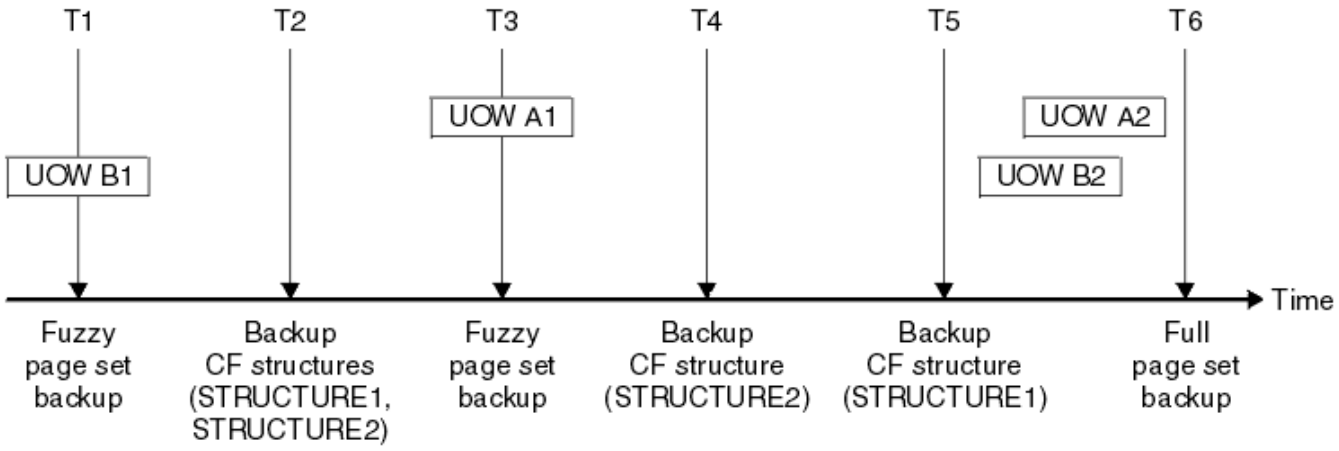


Figure 43. Example of queue manager backup activity

This is what happens at each point in time:

Point in time T1
A fuzzy backup is created of your page sets, as described in How to back up and recover page sets.

The system recovery RBA of the queue manager is the lowest of the following:

- The recovery RBAs of the page sets being backed up at this point.
- The lowest recovery RBA required to recover the CF application structures. This relates to the recovery of backups of STRUCTURE1 and STRUCTURE2 created earlier.
- The recovery RBA for the oldest currently active unit of work within the queue manager (UOWB1).

The system recovery RBA for this point in time is given by messages issued by the DISPLAY USAGE command, which is part of the fuzzy backup process.

Point in time T2
Backups of the CF structures are created. CF structure STRUCTURE1 is backed up first, followed by STRUCTURE2.

The amount of log data that must be retained is unchanged, because the same data as determined from the system recovery RBA at T1 is still required to recover using the page set backups taken at T1.

Point in time T3
Another fuzzy backup is created.

The system recovery RBA of the queue manager is the lowest of the following:

- The recovery RBAs of the page sets being backed up at this point.
- The lowest recovery RBA required to recover CF structure STRUCTURE1, because STRUCTURE1 was backed up before STRUCTURE2.
- The recovery RBA for the oldest currently active unit of work within the queue manager (UOWA1).

The system recovery RBA for this point in time is given by messages issued by the DISPLAY USAGE command, which is part of the fuzzy backup process.

You can now reduce the log data retained, as determined by this new system recovery RBA.

Point in time T4
A backup is taken of CF structure STRUCTURE2. The recovery RBA for the recovery of the oldest required CF structure backup relates to the backup of CF structure STRUCTURE1, which was backed up at time T2.

The creation of this CF structure backup has no effect on the amount of log data that must be retained.

Point in time T5

A backup is taken of CF structure STRUCTURE1. The recovery RBA for recovery of the oldest required CF structure backup now relates to recovery of CF structure STRUCTURE2, which was backed up at time T4.

The creation of this CF structure backup has no effect on amount of log data that must be retained.

Point in time T6

A full backup is taken of your page sets as described in How to back up and recover page sets.

The system recovery RBA of the queue manager is the lowest of the following:

- The recovery RBAs of the page sets being backed up at this point.
- The lowest recovery RBA required to recover the CF structures. This relates to recovery of CF structure STRUCTURE2.
- The recovery RBA for the oldest currently active unit of work within the queue manager. In this case, there are no current units of work.

The system recovery RBA for this point in time is given by messages issued by the DISPLAY USAGE command, which is part of the full backup process.

Again, the log data retained can be reduced, because the system recovery RBA associated with the full backup is more recent.

z/OS Planning your z/OS UNIX environment

Certain processes within the IBM MQ queue manager, channel initiator, and mqweb server use z/OS UNIX System Services (z/OS UNIX) for their normal processing.

The queue manager and channel initiator started task user IDs need an OMVS segment with a UID defined in order to be able to access z/OS UNIX. The user IDs require no special permissions in z/OS UNIX.

Note: Although the queue manager and channel initiator make use of z/OS UNIX facilities (for example, to interface with TCP/IP services), they do not need to access any of the content of the IBM MQ installation directory in the z/OS UNIX file system. As a result, the queue manager and channel initiator do not require any configuration to specify the path for the z/OS UNIX file system.

The mqweb server, which hosts the IBM MQ Console and REST API, makes use of files in the IBM MQ installation directory in the z/OS UNIX file system. It also needs access to another file system which is used to store data such as configuration and log files. The mqweb started task JCL needs to be customized to reference these z/OS UNIX file systems.

The content of the IBM MQ directory in the z/OS UNIX file system is also used by applications connecting to IBM MQ. For example, applications using the IBM MQ classes for Java or IBM MQ classes for JMS interfaces.

See the following topics for the relevant configuration instructions:

- Environment variables relevant to IBM MQ classes for Java
- IBM MQ classes for Java libraries
- Setting environment variables
- Configuring the Java Native Interface (JNI) libraries

z/OS Planning for Advanced Message Security

TLS (or SSL) can be used to encrypt and protect messages flowing on a network, but this does not protect messages when they are on a queue ("at rest"). Advanced Message Security (AMS) protects the messages from the time that they are first put to a queue, until they are got, so that only the intended recipients of the message can read that message. The messages are encrypted and signed during put processing, and unprotected during get processing.

AMS can be configured to protect messages in different ways:

1. A message can be signed. The message is in clear text, but there is a checksum, which is signed. This allows any changes in the message content to be detected. From the signed content, you can identify who signed the data.
2. A message can be encrypted. The contents are not visible to anyone without the decryption key. The decryption key is encrypted for each recipient.
3. A message can be encrypted and signed. They decryption key is encrypted for each recipient, and from the signing you can identify who sent the message.

The encryption and signing use digital certificates and key rings.

You can set up a client to use AMS, so the data is protected before the data is put on the client channel. Protected messages can be sent to a remote queue manager, and you need to configure the remote queue manager to process these messages.

Setting up AMS

An AMS address space is used for doing the AMS work. This has additional security set up, to give access to and protect the use of key rings and certificates.

You configure which queues are to be protected by using a utility program (CSQ0UTIL) to define the security policies for queues.

Once AMS is set up

You need to set up a digital certificate and a key ring for people who put messages, and the people who get messages.

If a user, Alice, on z/OS needs to send a message to Bob, AMS needs a copy of the public certificate for Bob.

If Bob wants to process a message from Alice, AMS needs the public certificate for Alice, or the same certificate authority certificate used by Alice.

Attention: You need to:

- Carefully plan who can put to, or get from, queues
- Identify the people and their certificate names.

It is easy to make mistakes, and problems can be hard to resolve.

Related concepts

“Planning for your queue manager” on page 148
When you are setting up a queue manager, your planning should allow for the queue manager to grow, so that the queue manager meets the needs of your enterprise.

z/OS Planning for Managed File Transfer

Use this section as guidance on how you need to set up your system to run Managed File Transfer (MFT) on z/OS.

z/OS Planning for Managed File Transfer - hardware and software requirements

Use this topic as guidance on how you need to set up hardware and software requirements on your system to run Managed File Transfer (MFT) on z/OS.

Software requirements

Managed File Transfer is written in Java, with some shell scripts and JCL to configure and operate the program.

Important: You must be familiar with z/OS UNIX System Services (z/OS UNIX) in order to configure Managed File Transfer. For example:

- The file directory structure, with names such as `/u/userID/myfile.txt`
- z/OS UNIX commands, for example:
  - `cd` (change directory)
  - `ls` (list)
  - `chmod` (change the file permissions)
  - `chown` (change file ownership or groups which can access the file or directory)

You require the following products in z/OS UNIX to be able to configure and run MFT:

1. Java, for example, in directory `/java/java80_bit64_GA/J8.0_64/`
2. IBM MQ 9.4.0, for example, in directory `/mqm/V9R3M0`
3. If you want to use Db2 for status and history, you need to install Db2 JDBC libraries, for example, in directory `/db2/db2v10/jdbc/libs`.

Product registration

At startup Managed File Transfer checks the registration in `sys1.parmlib`(IFAPRDxx) concatenation. The following code is an example of how you register MFT:

```
PRODUCT OWNER('IBM CORP')
NAME('WS MQ FILE TRANS')
ID(5655-MFT)
VERSION(*) RELEASE(*) MOD(*)
FEATURENAME('WS MQ FILE TRANS')
STATE(ENABLED)
```

Disk space

The IBM MQ for z/OS Program Directory states the DASD and zFS storage requirements for Managed File Transfer. For download links for the Program Directory for IBM MQ for z/OS, see IBM MQ 9.4 PDF files for product documentation and Program Directories.

z/OS Planning for Managed File Transfer - topologies

Use this topic as guidance on what topology you need on your system to run Managed File Transfer (MFT) on z/OS.

Managed File Transfer queue managers

IBM MQ Managed File Transfer topologies consist of:

Agents, and their associated queue managers
The agent uses system queues hosted on their agent queue manager to maintain state information and receive requests for work.

A command queue manager
This acts as a gateway into an MFT topology. It is connected to the agent queue managers through either sender and receiver channels, or clustering. When certain commands are run, they connect directly to the command queue manager, and send a message to the specified agent. This message is routed through the IBM MQ network to the agent queue manager, where it is picked up by the agent and processed.

A coordination queue manager
This is a central hub that has knowledge of the entire topology. The coordination queue manager is connected to all of the agent queue managers in a topology through either sender and receiver

channels, or using clustering. Agents regularly publish status information to the coordination queue manager, and store their transfer templates there.

It is possible for a single queue manager to perform multiple roles within a topology. For example, the same queue manager can be configured as both the coordination queue manager and the command queue manager for a topology.

If you are using multiple queue managers you need to set up channels between the queue managers. You can either do this by using clustering or by using point-to-point connections.

When using IBM MQ Managed File Transfer for z/OS, there are a number of things to consider when determining which queue managers to use for the different roles within a topology.

Agent queue managers

The agent queue manager for an IBM MQ Managed File Transfer for z/OS agent must be running on z/OS.

If:

- The agent is running Managed File Transfer for z/OS on IBM MQ 9.1 or later
- And, the agent queue manager is licensed for IBM MQ Advanced for z/OS Value Unit Edition (Advanced VUE)

the agent can connect to the queue manager using the CLIENT transport.

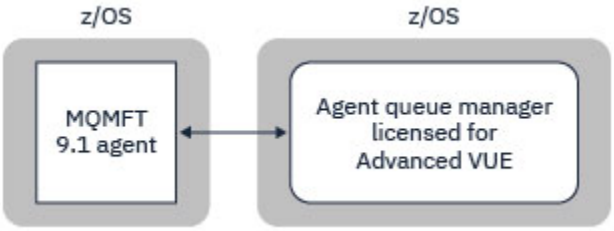


Figure 44. MFT 9.1 agents on z/OS can connect to a queue manager using the CLIENT transport, assuming the queue manager is licensed for Advanced VUE.

If:

- The agent is running Managed File Transfer for z/OS on IBM MQ 9.0 or earlier
- Or, the agent queue manager is running Managed File Transfer for z/OS on IBM MQ 9.0 or later, and the agent queue manager is licensed for either MFT, IBM MQ Advanced for z/OS, or Advanced VUE

the agent must connect to the queue manager using the BINDINGS transport.

Figure 45. MFT 9.0 agents on z/OS and 9.1 agents that have an agent queue manager licensed for either MFT or IBM MQ Advanced, must connect using the BINDINGS transport.

Command queue managers

The Which MFT commands and processes connect to which queue manager topic shows all of the commands that connect to the command queue manager for a Managed File Transfer topology.

Note: When running these commands on z/OS, the command queue manager must also be on z/OS.

If the command queue manager is licensed for Advanced VUE, the commands can connect to the queue manger using the CLIENT transport. Otherwise, the commands must connect to the command queue manager using the BINDINGS transport.

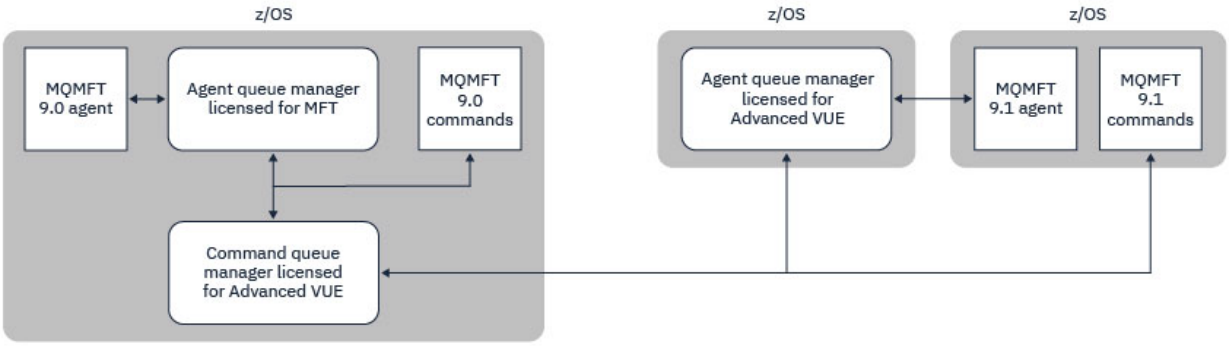


Figure 46. Commands connect to the command queue manager for an MFT topology. When running these commands on z/OS, the command queue manager must also be on z/OS

Coordination queue managers

IBM MQ Managed File Transfer for z/OS agents can be part of a topology where the coordination queue manager is either running on z/OS, or is running on a multiplatform.

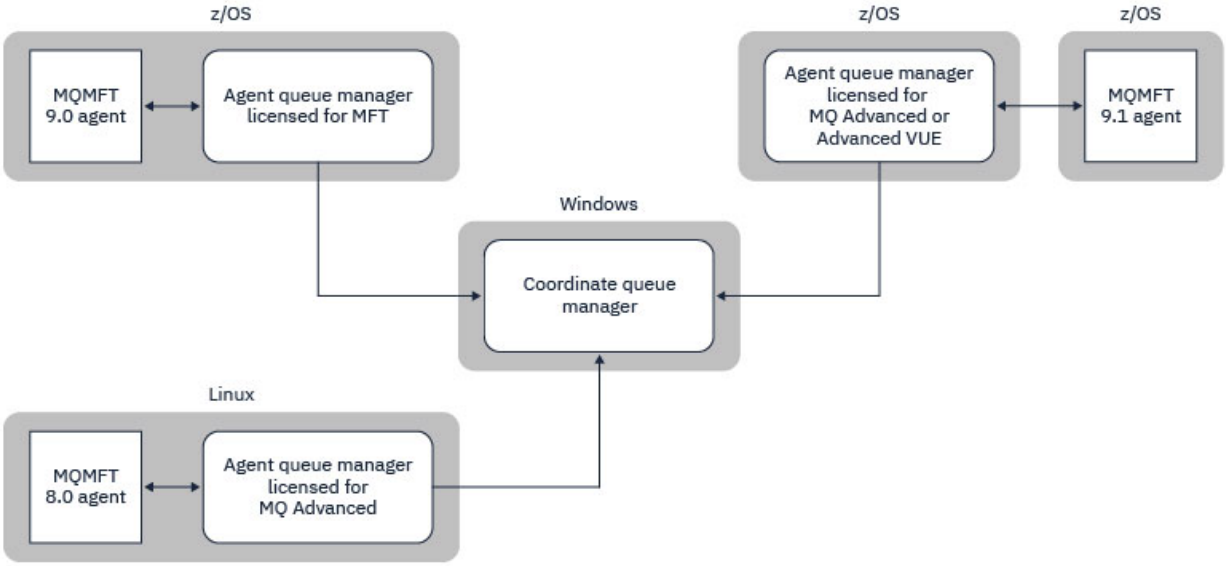


Figure 47. MFT agents running on z/OS can be part of an MFT topology where the coordination queue manager is running on an IBM MQ multiplatform.

The Which MFT commands and processes connect to which queue manager topic shows the commands that connect to the coordination queue manager for a Managed File Transfer topology. It is possible to run these commands on z/OS and have then connect to the coordination queue manager running on a different platform.

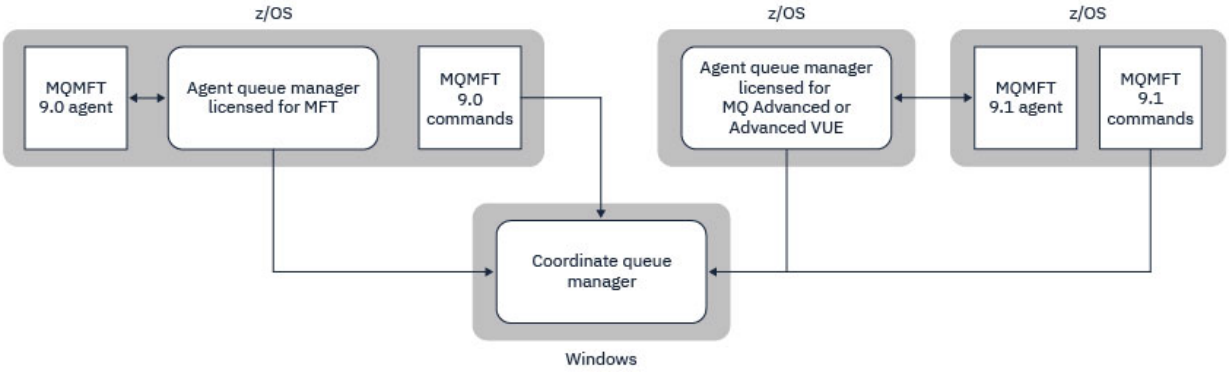


*Figure 48. Certain commands, such as **fteListAgents**, connect directly to the coordination queue manager for an MFT topology.*

How many agents do I need?

The agents do the work in transferring data, and when you make a request to transfer data you specify the name of an agent.

By default an agent can process 25 send and 25 receive requests concurrently. You can configure these processes. See Managed File Transfer configuration options on z/OS for more information.

If the agent is busy then work is queued. The time taken to process a request depends on multiple factors, for example, the amount of data to be sent, the network bandwidth, and the delay on the network.

You might want to have multiple agents to process work in parallel.

You can also control which resources an agent can access, so you might want some agents to work with a limited subset of data.

If you want to process requests with different priority you can use multiple agents and use workload manager to set the priority of the jobs.

Running the agents

Typically the agents are long running processes. The processes can be submitted as jobs that run in batch, or as started tasks.

z/OS Planning for Managed File Transfer - security considerations

Use this topic as guidance on what security considerations you need on your system to run Managed File Transfer (MFT) on z/OS.

Security

You need to identify which user IDs are going to be used for MFT configuration and for MFT operation.

You need to identify the files or queues you transfer, and which user IDs are going to be submitting transfer requests to MFT.

When you customize the agents and logger, you specify the group of users that is allowed to run MFT services, or do MFT administration.

You should set up this group before you start customizing MFT. As MFT uses IBM MQ queues, if you have security enabled in the queue manager, MFT requires access to the following resources:

Table 26. MQADMIN resource class

Name	Access required
QUEUE.SYSTEM.FTE.EVENT.agent_name	Update

Table 26. MQADMIN resource class (continued)	
Name	**Access required**
QUEUE.SYSTEM.FTE.COMMAND.agent_name	Update
CONTEXT.SYSTEM.FTE.COMMAND.agent_name	Update
QUEUE.SYSTEM.FTE.STATE.agent_name	Update
QUEUE.SYSTEM.FTE.DATA.agent_name	Update
QUEUE.SYSTEM.FTE.REPLY.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHAGT1.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHTRN1.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHOPS1.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHSCH1.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHMON1.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHADM1.agent_name	Update

Table 27. MQQUEUE resource class	
Name	**Access required**
SYSTEM.FTE.AUTHAGT1.agent_name	Update
SYSTEM.FTE.AUTHTRN1.agent_name	Update
SYSTEM.FTE.AUTHOPS1.agent_name	Update
SYSTEM.FTE.AUTHSCH1.agent_name	Update
SYSTEM.FTE.AUTHMON1.agent_name	Update

You can use user sandboxing to determine which parts of the file system the user who requests the transfer can access.

To enable user sandboxing, add the `userSandboxes=true` statement to the *agent.properties* file for the agent that you want to restrict, and add appropriate values to the `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name/UserSandboxes.xml` file.

See Working with user sandboxes for further information.

This user ID is configured in `UserSandboxes.xml` files.

This XML file has information like user ID, or user ID* and a list of resource that can be used (included), or cannot be used (excluded). You need to define specific user IDs that can access which resources: for example:

Table 28. Example user ID together with access to specific resources			
User ID	**Access**	**Include or Exclude**	**Resource**
Admin*	Read	Include	`/home/user/**`
Admin*	Read	Exclude	`/home/user/private/**`
Sysprog	Read	Include	`/home/user/**`
Admin*	Read	Include	Application.reply.queue

Notes:

1. If `type=queue` is specified, the resource is either a queue name, or `queue@qmgr`.
2. If the resource begins with `//`, the resource is a data set; otherwise the resource is a file in z/OS UNIX.
3. The user ID is the user ID from the MQMD structure, so this might not reflect the user ID that actually puts the message.
4. For requests on the local queue manager you can use MQADMIN CONTEXT.* to limit which users can set this value.
5. For requests coming in over a remote queue manager, you have to assume that the distributed queue managers have security enabled to prevent unauthorized setting of the user ID in the MQMD structure.
6. A user ID of SYSPROG1 on a Linux machine, is the same user ID SYSPROG1 for the security checking on z/OS.

z/OS Planning to use the IBM MQ Console and REST API on z/OS

The IBM MQ Console and REST API are applications that run in a WebSphere Liberty (Liberty) server known as mqweb. The mqweb server runs as a started task. The IBM MQ Console allows a web browser to be used to administer queue managers. The REST API provides a simple programmatic interface for applications to do queue manager administration, and to perform messaging.

Installation and configuration files

You need to install the IBM MQ for z/OS UNIX System Services Web Components feature, which will install the files needed to run the mqweb server in z/OS UNIX System Services (z/OS UNIX). You need to be familiar with z/OS UNIX to be able to configure and manage the mqweb server.

See IBM MQ for z/OS Program Directory PDF files for information on installing IBM MQ for z/OS UNIX System Services Components.

The IBM MQ files in z/OS UNIX are installed with various attributes set that are required for the correct operation of the mqweb server. If you need to copy the IBM MQ z/OS UNIX installation files, for example if you have installed IBM MQ on one system, and run IBM MQ on a different system, you should copy the IBM MQ ZFS created during the installation, and mount it read only at the destination. Copying the files in other ways might cause some file attributes to be lost.

You need to decide upon the location for, and create, a Liberty user directory when you create the mqweb server. This directory contains configuration and log files, and the location can be something similar to `/var/mqm/mqweb`.

Using the IBM MQ Console and REST API with queue managers at different levels

The REST API can directly interact only with queue managers that run at the same Version, Release, and Modification (VRM) as the mqweb server which runs the REST API. For example, the IBM MQ 9.4.0 REST API can directly interact only with local queue managers at IBM MQ 9.4.0, and the IBM MQ 9.3.5 REST API can directly interact only with local queue managers at IBM MQ 9.3.5.

You can use the REST API to administer a queue manager at a different version from the mqweb server by configuring a gateway queue manager. However, you need at least one queue manager at the same version as the mqweb server to act as the gateway queue manager. For more information, see Remote administration using the REST API.

The IBM MQ Console can be used to manage local queue managers that run at the same version as the IBM MQ Console. From IBM MQ 9.3.0, you can also use the IBM MQ Console to administer a queue manager running on a remote system, or at a different version to the IBM MQ Console. For more information, see IBM MQ Console: Adding a remote queue manager.

Migration

If you have only one queue manager, you can run the mqweb server as a single started task, and change the libraries it uses when you migrate your queue manager.

If you have more than one queue manager, during migration you can start mqweb servers at different versions by using started tasks with different names. These names can be any name you want. For example, you can start an IBM MQ 9.3.0 mqweb server using a started task named MQWB0930, and an IBM MQ 9.3.5 mqweb server using a started task named MQWB0935.

Then, when you migrate the queue managers from one version to a later version, the queue managers become available in the mqweb server for the later version, and are no longer available in the mqweb server for the earlier version.

After you have migrated all the queue managers to the later version, you can delete the mqweb server for the earlier version.

HTTP ports

The mqweb server uses up to two ports for HTTP:

- One for HTTPS, with a default value of 9443.
- One for HTTP. HTTP is not enabled by default, but if enabled, has a default value of 9080.

If the default port values are in use, you must allocate other ports. If you have more than one mqweb server running simultaneously for more than one version of IBM MQ, you must allocate separate ports for each version. For more information on setting the ports that the mqweb server uses, see Configuring the HTTP and HTTPS ports.

You can use the following TSO command to display information about a port:

```
NETSTAT TCP tcpip (PORT portNumber)
```

where *tcpip* is the name of the TCP/IP address space, and *portNumber* specifies the number of the port to display information about.

Security - starting the mqweb server

The mqweb server user ID needs certain authorities. For more information, see Authority required by the mqweb server started task user ID.

Security - using the IBM MQ Console and REST API

When you use the IBM MQ Console and REST API, you must authenticate as a user that is included in a configured registry. These users are assigned specific roles that determine the actions the users can perform. For example, to use the messaging REST API, a user must be assigned the `MQWebUser` role. For more information about the available roles for the IBM MQ Console and REST API, and the access that these roles grant, see Roles on the IBM MQ Console and REST API.

For more information about configuring security for the IBM MQ Console and REST API, see IBM MQ Console and REST API security.

Avisos

Esta información se ha desarrollado para productos y servicios ofrecidos en los Estados Unidos.

Es posible que IBM no ofrezca los productos, servicios o las características que se tratan en este documento en otros países. Consulte al representante local de IBM para obtener información sobre los productos y servicios que actualmente pueden adquirirse en su zona. Las referencias a programas, productos o servicios de IBM no pretenden establecer ni implicar que sólo puedan utilizarse dichos productos, programas o servicios de IBM. En su lugar podrá utilizarse cualquier producto, programa o servicio equivalente que no infrinja ninguno de los derechos de propiedad intelectual de IBM. No obstante, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio no IBM.

IBM puede tener patentes o solicitudes de patentes pendientes que cubran el tema principal descrito en este documento. El suministro de este documento no le otorga ninguna licencia sobre estas patentes. Puede enviar consultas sobre licencias, por escrito, a:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Para consultas sobre licencias relacionadas con información de doble byte (DBCS), póngase en contacto con el Departamento de propiedad intelectual de IBM de su país o envíe las consultas por escrito a:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokio 103-8510, Japón

El párrafo siguiente no se aplica al Reino Unido ni a ningún otro país donde estas disposiciones contradigan la legislación vigente: INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL" SIN NINGÚN TIPO DE GARANTÍA, YA SEA EXPLÍCITA O IMPLÍCITA, INCLUYENDO, PERO SIN LIMITARSE A, LAS GARANTÍAS IMPLÍCITAS DE NO INCUMPLIMIENTO, COMERCIABILIDAD O IDONEIDAD PARA UNA FINALIDAD DETERMINADA. Algunas legislaciones no contemplan la exclusión de garantías, ni implícitas ni explícitas, en determinadas transacciones, por lo que puede haber usuarios a los que no les afecte dicha norma.

Esta información puede contener imprecisiones técnicas o errores tipográficos. La información aquí contenida está sometida a cambios periódicos; tales cambios se irán incorporando en nuevas ediciones de la publicación. IBM puede realizar mejoras y/o cambios en los productos y/o programas descritos en esta publicación en cualquier momento sin previo aviso.

Las referencias hechas en esta publicación a sitios web que no son de IBM se proporcionan sólo para la comodidad del usuario y no constituyen de modo alguno un aval de esos sitios web. Los materiales de estos sitios web no forman parte de los materiales para este producto IBM, por lo que la utilización de dichos sitios web es a cuenta y riesgo del usuario.

IBM puede utilizar o distribuir cualquier información que el usuario le proporcione del modo que considere apropiado sin incurrir por ello en ninguna obligación con respeto al usuario.

Los titulares de licencias de este programa que deseen información del mismo con el fin de permitir: (i) el intercambio de información entre los programas creados de forma independiente y otros programas (incluido este) y (ii) el uso mutuo de la información intercambiada, deben ponerse en contacto con:

IBM Corporation
Software Interoperability Coordinator, Department 49XA

3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Dicha información puede estar disponible, sujeta a los términos y condiciones apropiados, incluyendo, en algunos casos, el pago de una cantidad.

El programa bajo licencia que se describe en esta información y todo el material bajo licencia disponible para el mismo lo proporciona IBM bajo los términos del Acuerdo de cliente de IBM, el Acuerdo de licencia de programas internacional de IBM o cualquier acuerdo equivalente entre las partes.

Los datos de rendimiento incluidos en este documento se han obtenido en un entorno controlado. Por consiguiente, los resultados obtenidos en otros entornos operativos pueden variar de manera significativa. Es posible que algunas mediciones se hayan realizado en sistemas en nivel de desarrollo y no existe ninguna garantía de que estas mediciones serán las mismas en sistemas disponibles generalmente. Además, es posible que algunas mediciones se hayan estimado mediante extrapolación. Los resultados reales pueden variar. Los usuarios de este documento deben verificar los datos aplicables a su entorno específico.

La información relativa a productos que no son de IBM se obtuvo de los proveedores de esos productos, sus anuncios publicados u otras fuentes de disponibilidad pública. IBM no ha comprobado estos productos y no puede confirmar la precisión de su rendimiento, compatibilidad o alguna reclamación relacionada con productos que no sean de IBM. Todas las preguntas sobre las prestaciones de productos que no son de IBM deben dirigirse a los proveedores de dichos productos.

Todas las declaraciones relacionadas con una futura intención o tendencia de IBM están sujetas a cambios o se pueden retirar sin previo aviso y sólo representan metas y objetivos.

Este documento contiene ejemplos de datos e informes que se utilizan diariamente en la actividad de la empresa. Para ilustrar los ejemplos de la forma más completa posible, éstos incluyen nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier similitud con los nombres y direcciones utilizados por una empresa real es puramente casual.

LICENCIA DE DERECHOS DE AUTOR:

Esta información contiene programas de aplicación de ejemplo en lenguaje fuente que ilustran técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir estos programas de ejemplo de cualquier forma sin pagar ninguna cuota a IBM para fines de desarrollo, uso, marketing o distribución de programas de aplicación que se ajusten a la interfaz de programación de aplicaciones para la plataforma operativa para la que se han escrito los programas de ejemplo. Los ejemplos no se han probado minuciosamente bajo todas las condiciones. IBM, por tanto, no puede garantizar la fiabilidad, servicio o funciones de estos programas.

Puede que si visualiza esta información en copia software, las fotografías e ilustraciones a color no aparezcan.

Información acerca de las interfaces de programación

La información de interfaz de programación, si se proporciona, está pensada para ayudarle a crear software de aplicación para su uso con este programa.

Este manual contiene información sobre las interfaces de programación previstas que permiten al cliente escribir programas para obtener los servicios de IBM MQ.

Sin embargo, esta información puede contener también información de diagnóstico, modificación y ajustes. La información de diagnóstico, modificación y ajustes se proporciona para ayudarle a depurar el software de aplicación.

Importante: No utilice esta información de diagnóstico, modificación y ajuste como interfaz de programación porque está sujeta a cambios.

Marcas registradas

IBM, el logotipo de IBM , ibm.com, son marcas registradas de IBM Corporation, registradas en muchas jurisdicciones de todo el mundo. Hay disponible una lista actual de marcas registradas de IBM en la web en "Copyright and trademark information"www.ibm.com/legal/copytrade.shtml. Otros nombres de productos y servicios pueden ser marcas registradas de IBM o de otras empresas.

Microsoft y Windows son marcas registradas de Microsoft Corporation en Estados Unidos y/o otros países.

UNIX es una marca registrada de Open Group en Estados Unidos y en otros países.

Linux es una marca registrada de Linus Torvalds en Estados Unidos y en otros países.

Este producto incluye software desarrollado por Eclipse Project (https://www.eclipse.org/).

Java y todas las marcas registradas y logotipos son marcas registradas de Oracle o sus afiliados.

Número Pieza:

(1P) P/N: