9.4

Planung für IBM MQ

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen” auf Seite 217 gelesen werden.

Diese Ausgabe bezieht sich auf Version 9 Release 4 von IBM® MQ und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuausgabe geändert wird.

Wenn Sie Informationen an IBMsenden, erteilen Sie IBM ein nicht ausschließliches Recht, die Informationen in beliebiger Weise zu verwenden oder zu verteilen, ohne dass eine Verpflichtung für Sie entsteht.

Inhaltsverzeichnis

IBM MQ-Architektur planen

Beachten Sie bei der Planung einer IBM MQ-Umgebung die Unterstützung, die IBM MQ für Architekturen mit einzelnen oder mehreren Warteschlangenmanagern sowie für Punkt-zu-Punkt- und Publish/Subscribe-Messaging bereitstellt. Planen Sie auch den Ressourcenbedarf und die Nutzung von Protokollierungs- und Sicherheitsfunktionen.

Informationen zu diesem Vorgang

Bevor Sie mit der Planung einer IBM MQ-Architektur beginnen, sollten Sie sich mit den grundlegenden IBM MQ-Konzepten vertraut machen. Siehe IBM MQ Technische Übersicht.

IBM MQ-Architekturen reichen von einfachen Architekturen mit einem einzigen Warteschlangenmanager bis hin zu komplexeren Netzen miteinander verbundener Warteschlangenmanager. Mehrere Warteschlangenmanager werden unter Verwendung von verteilten Warteschlangenverfahren miteinander verbunden. Weitere Informationen zum Planen eines einzelnen Warteschlangenmanagers und mehrerer WS-Manager-Architekturen finden Sie in den folgenden Themen:

- „Architekturen auf der Basis eines einzelnen Warteschlangenmanagers” auf Seite 19
- „Architekturen auf der Basis von mehreren Warteschlangenmanagern” auf Seite 20
  - „Verteilte Warteschlangen und Cluster planen” auf Seite 21
  - „Verteiltes Publish/Subscribe-Netz planen” auf Seite 77

z/OS Unter IBM MQ for z/OS können Sie mit gemeinsam genutzten Warteschlangen und Gruppen mit gemeinsamer Warteschlange die Implementierung des Lastausgleichs aktivieren und festlegen, dass Ihre IBM MQ-Anwendungen skalierbar und hoch verfügbar sind. Informationen zu gemeinsam genutzten Warteschlangen und Gruppen mit gemeinsamer Warteschlange finden Sie unter Gemeinsam genutzte Warteschlangen und Gruppen mit gemeinsamer Warteschlange.

IBM MQ stellt zwei verschiedene Release-Modelle bereit:

- Das Release Long Term Support (LTS) eignet sich am besten für Systeme, die eine langfristige Implementierung und maximale Stabilität erfordern.
- Das Release Continuous Delivery (CD) ist für Systeme gedacht, die schnell die neuesten funktionalen Erweiterungen für IBM MQnutzen müssen.

Beide Releasetypen werden auf die gleiche Weise installiert, aber es gibt Überlegungen in Bezug auf die Unterstützung und Migration, die Sie verstehen müssen. Weitere Informationen finden Sie unter IBM MQ -Releasetypen und -Versionssteuerung.

Weitere Informationen zur Planung für mehrere Installationen, Speicher-und Leistungsanforderungen sowie die Verwendung von Clients finden Sie in den anderen Unterabschnitten.

Zugehörige Konzepte

IBM MQ -Releasetypen und -Versionssteuerung

„Planning your IBM MQ environment on z/OS” auf Seite 153
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.

Verfügbarkeit, Wiederherstellung und Neustart

Zugehörige Tasks

Überprüfen der Anforderungen

Sicherstellen, dass Nachrichten nicht verloren gehen (Protokollierung)

IBM MQ -Releasetypen: Überlegungen zur Planung

Die beiden Hauptreleasetypen für IBM MQ sind Long Term Support (LTS) und ein Continuous Delivery (CD). Für jede unterstützte Plattform wirkt sich der von Ihnen ausgewählte Releasetyp auf Bestellung, Installation, Wartung und Migration aus.

Ausführliche Informationen zu den Releasetypen finden Sie unter IBM MQ -Releasetypen und -Versionssteuerung.

Hinweise zu IBM MQ for Multiplatforms

Bestellung

In Passport Advantage gibt es zwei separate eAssemblies für IBM MQ 9.4. Die eine enthält Installationsimages für IBM MQ 9.4.0 als Long Term Support-Release und die andere enthält Installationsimages für IBM MQ 9.4.x als Continuous Delivery-Release. Laden Sie die Installationsimages von der eAssembly gemäß Ihrer Wahl des Release herunter.

Alle IBM MQ-Versionen gehören zu derselben Produkt-ID; bei IBM MQ 9.4 gilt dies sowohl für die LTS-Releases als auch für die CD-Releases.

Die Nutzungsberechtigung für IBM MQ erstreckt sich auf das gesamte Produkt (PID), vorbehaltlich der Einschränkungen lizenzierter Komponenten und Preismetriken. Dies bedeutet, dass Sie frei zwischen LTS -Release und CD -Release-Installationsimages für IBM MQ 9.4wählen können.

Installation

Nachdem Sie ein Installationsimage von Passport Advantageheruntergeladen haben, sollten Sie nur die Komponenten für die Installation auswählen, für die Sie eine Berechtigung erworben haben. Weitere Informationen dazu, welche installierbaren Komponenten in den einzelnen gebührenpflichtigen Komponenten enthalten sind, finden Sie unter IBM MQ-Lizenzinformationen.

Sie können IBM MQ 9.4.0 LTS -Release und IBM MQ 9.4.x CD -Release in demselben Betriebssystemimage installieren. Wenn Sie dies tun, werden die Komponenten als unterschiedliche Installationen angezeigt, was durch die Unterstützung mehrerer Versionen von IBM MQ ermöglicht wird. Jede Version verfügt über unterschiedliche Gruppen von Warteschlangenmanagern, die dieser Version zugeordnet sind.

Jedes neue CD-Release wird als separates Installationsimage bereitgestellt. Das neue Release von CD kann zusammen mit einem vorhandenen Release installiert werden oder ein früheres CD -Release kann vom Installationsprogramm auf das neue Release aktualisiert werden.

CD -Releases enthalten funktionale Erweiterungen sowie die neuesten Fehlerkorrekturen und Sicherheitsupdates. Jedes CD -Release ist kumulativ und ersetzt vollständig alle vorherigen für diese Version von IBM MQ. Sie können also ein bestimmtes CD -Release überspringen, wenn es keine Funktion enthält, die für Ihr Unternehmen relevant ist.

Wartung

Das Release LTS wird von der Anwendung von Fixpacks, die Fehlerkorrekturen bereitstellen, und kumulativen Sicherheitsupdates (CSUs), die Sicherheitspatches bereitstellen, gewartet. Die Fixpacks und CSUs werden regelmäßig zur Verfügung gestellt und sind kumulativ.

Für CDwerden CSUs nur für das neueste CD -Release erstellt, das möglicherweise in einer nachfolgenden Version vorliegt.

Möglicherweise werden Sie gelegentlich vom IBM Support-Team angewiesen, einen vorläufigen Fix anzuwenden. Vorläufige Fixes werden auch als provisorische Fixes oder Testfixes bezeichnet und verwendet, um dringende Updates anzuwenden, die nicht auf die nächste Wartungsbereitstellung warten können.

Migration zwischen LTS-Release und CD-Release

Es gibt Einschränkungen und Grenzen, aber im Allgemeinen kann ein einzelner Warteschlangenmanager von der Verwendung des LTS-Release-Codes auf den CD-Release-Code oder von der Verwendung des

CD-Release-Codes auf den LTS-Release-Code migriert werden, sofern das Ziel-Release höher ist, als das vor der Migration verwendete.

Es sind zwei Ansätze möglich:

- Installieren Sie den neuen Releasecode anstelle einer vorhandenen Installation von IBM MQ, damit diese aktualisiert wird. Alle Warteschlangenmanager, die der Installation zugeordnet sind, verwenden das neue Release von Code, wenn sie gestartet werden.
- Installieren Sie das neue Release von Code als neue Installation, und verschieben Sie dann mit dem Befehl setmqm einzelne WS-Manager-Instanzen in die neue Installation.

Wenn ein Warteschlangenmanager mit der Ausführung eines CD -Release von Code startet, wird die Befehlsebene des Warteschlangenmanagers aktualisiert, um den neuen Release-Level anzugeben. Dies bedeutet, dass alle im Release bereitgestellten neuen Funktionen aktiviert sind und Sie den Warteschlangenmanager nicht mehr mit einem Code-Release mit einer niedrigeren VRM -Nummer erneut starten können.

Hinweise zu IBM MQ for z/OS

z/OS

Bestellung

Bei der Bestellung von IBM MQ for z/OS 9.4 werden in ShopZ zwei verschiedene Funktionen angeboten. Die beiden Funktionen entsprechen dem LTS-Release und dem CD-Release. Beide Funktionen sind auf dieselbe Produkt-ID (PID) anwendbar. Es handelt sich um die Produkt-ID, die lizenziert ist. Wenn eine Komponente lizenziert ist, besteht die Berechtigung, bei Bedarf die Alternativfunktion zu verwenden. Bei der Bestellung wählen Sie das Feature aus, das dem Release LTS oder dem Release CD entspricht.

Wenn Sie Produkte für die Aufnahme in ein ServerPacauswählen, können Sie das LTS -Release und das CD -Release nicht in derselben ServerPac -Reihenfolge auswählen, da die Produkte nicht mit SMP/E in derselben Zielzone installiert werden können.

Installation

LTS-Releases und CD-Releases werden in unterschiedlichen Gruppen von FMIDs bereitgestellt. Beachten Sie, dass diese FMIDs nicht in derselben SMP/E-Zielzone installiert werden können. Gehen Sie wie folgt vor, wenn Sie sowohl das Release LTS als auch das Release CD benötigen:

- Installieren Sie das Release LTS und das Release CD in separaten Zielzonen.
- Verwalten Sie separate Ziel-und Verteilungsbibliotheken für die beiden Releases.

Wenn sich Ihr Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange befindet und Sie ein Upgrade auf die neueste CD-Version durchführen, müssen Sie alle Warteschlangenmanager in der Gruppe aktualisieren.

Die Befehlsebene eines Warteschlangenmanagers ist die dreistellige VRM-Ebene. EinIBM MQ Programm kann aufrufen`MQINQ` , vorbei an der`MQIA_COMMAND_LEVEL` Selektor, um die Befehlsebene des Warteschlangenmanagers abzurufen, mit dem er verbunden ist.

Da die Releases unterschiedliche FMIDs verwenden, können Sie ein CD -Release nicht mit Wartung für ein LTS -Release oder umgekehrt aktualisieren. Ebenso gibt es keine Möglichkeit, eine Version des Produktcodes von einem LTS -Release zu einem CD -Release oder umgekehrt zu wechseln. Sie können einen Warteschlangenmanager jedoch zwischen den Releasemodellen wechseln. Siehe Migration zwischen LTS-Release und CD-Release.

Anmerkung:

Die Releases IBM MQ 9.0.x und IBM MQ 9.1.x CD hatten separate Versions-und Releaseabhängige FMIDs. Für die Umstellung von 9.0.x CD auf 9.1.x CD ist mindestens eine vollständige SMP/E-Installation erforderlich.

Ab IBM MQ for z/OS 9.2.0verwendet das CD -Release eine Gruppe von FMIDs, die für alle IBM MQ for z/OS -Releases mit der Versionsnummer 9gleich bleiben. Da jede neue Version von IBM MQ sowohl als CD -als auch als LTS -Release verfügbar ist, können Sie ein Upgrade für CD -Releases durchführen,

indem Sie PTFs auf eine einzelne SMP/E-Installation anwenden, auch wenn Sie eine Hauptversionsgrenze überschreiten. Sie können beispielsweise von IBM MQ for z/OS 9.2.0 CDzu IBM MQ for z/OS 9.2.2 CD, zu IBM MQ for z/OS 9.2.4 CDund zu IBM MQ for z/OS 9.3.0 CDwechseln, indem Sie nur PTFs anlegen.

Sie können ein LTS- von einem CD-Release mit demselben VRM-Stand unterscheiden, indem Sie die Nachricht CSQY000I im Jobprotokoll des Warteschlangenmanagers anzeigen.

Wartung

IBM MQ for z/OS verwendet PTFs für die Wartung.

LTS PTFs sind für eine bestimmte Gruppe von Bibliotheken vorgesehen, die einem bestimmten Release entsprechen. Für UNIX System Services-Features (d. h. JMS und Webbenutzerschnittstelle, Connector Pack und Managed File Transfer) sind die z/OS -PTFs direkt an den Multiplatforms-Fixpacks und kumulativen Sicherheitsupdates (CSUs) ausgerichtet. Diese Fixes sind kumulativ und werden gleichzeitig mit dem entsprechenden Multiplatforms-Fixpack oder CSU verfügbar gemacht.

CD CD CSUs werden normalerweise nicht zwischen CD-Releases zur Verfügung gestellt, sind aber im nächsten IBM MQ for z/OS CD -Release enthalten. Sie können sich auch an den Support wenden, um ein + + USERMOD-Modul anzufordern.

Andere Fixes in IBM MQ for z/OS sind unterschiedliche Fixes für bestimmte Teile. Diese Fixes lösen bestimmte Probleme, sind nicht kumulativ und werden verfügbar gemacht, wenn sie erstellt werden.

Migration zwischen LTS-Release und CD-Release

Es gibt Einschränkungen und Grenzen, aber in der Regel kann ein einzelner Warteschlangenmanager von der Verwendung des LTS-Release-Codes auf den CD-Release-Code oder von der Verwendung des CD-Release-Codes auf den LTS-Release-Code migriert werden, sofern das Ziel-Release höher ist, als vor der Migration.

Ab IBM MQ for z/OS 9.2.0können Sie zwischen CD -und LTS -Releases mit demselben VRM so oft wie nötig hin-und hermigrieren, ohne dass sich dies auf die Rückwärtsmigration auswirkt. Ein Warteschlangenmanager kann beispielsweise in IBM MQ for z/OS 9.3.0 LTSgestartet werden, dann in IBM MQ for z/OS 9.3.0 CDbeendet und gestartet werden und anschließend in IBM MQ for z/OS 9.3.0 LTSbeendet und gestartet werden.

IBM MQ for z/OS hat traditionell eine Rückgriffsfunktion (Rückwärtsmigration) bereitgestellt, sodass Sie nach einem Zeitraum nach einer Migration auf das vorherige Release zurückgreifen können. Diese Funktionalität wird für LTS -Releases und für CD -Releases mit dem Modifikator `0` wie 9.3.0 CDbeibehalten, ist jedoch nicht möglich, wenn die Quelle oder das Ziel einer Migration ein CD -Release mit einer Modifikatornummer ungleich null ist, z. B. 9.2.5 oder 9.3.1.

Im Folgenden finden Sie gültige Migrationsszenarios und veranschaulichen, wie dieses Prinzip funktioniert:

Quellenrelease	Zielrelease	Anmerkungen
9.1.0 LTS	9.4.0 LTS oder 9.4.0 CD	Rückwärtsmigration wird nicht unterstützt, da 9.1.0 LTS nicht die Standardunterstützung bietet.
9.2.0 LTS	9.4.0 LTS oder 9.4.0 CD	Die Rückwärtsmigration wird unterstützt.
9.3.0 LTS	9.4.0 LTS oder 9.4.0 CD	Die Rückwärtsmigration wird unterstützt.
CD 9.3.5	9.4.0 LTS oder 9.4.0 CD	Eine Rückwärtsmigration wird nicht unterstützt, da das Quellenrelease ein CD-Release mit einem Modifikator ungleich null ist.

Quellenrelease	Zielrelease	Anmerkungen
9.4.0 LTS oder 9.4.0 CD	CD 9.4.1	Eine Rückwärtsmigration wird nicht unterstützt, da das Zielrelease ein CD-Release mit einem Modifikator ungleich null ist. `Write to operator with reply` CSQY041D wird ausgegeben, um die Migration zu bestätigen.

Zugehörige Tasks

z/OS Wartung unter z/OS anwenden und entfernen

Zugehörige Informationen

Download vonIBM MQ 9.4

IBM MQ und IBM MQ Appliance lokal - Überlegungen zur DSGVO-Umsetzung

Für folgende Produkt-IDs:

Verteilt

- 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

Hinweis:

Dieses Dokument soll Ihnen bei den Vorbereitungen für die Umsetzung der EU-Datenschutz-Grundverordnung (DSGVO) helfen. Es enthält Informationen zu den Features von IBM MQ, die Sie konfigurieren können, sowie Aspekte der Verwendung des Produkts, die Sie in Betracht ziehen sollten, um die Umsetzung der DSGVO in Ihrer Organisation zu fördern. Diese Informationen sind keine erschöpfende Liste, da die Kunden viele Möglichkeiten haben, Funktionen auszuwählen und zu konfigurieren, und die große Vielfalt an Möglichkeiten, die das Produkt in sich selbst und mit Anwendungen und Systemen anderer Hersteller verwendet werden kann.

Es liegt allein in der Verantwortung der Kunden, die Einhaltung der verschiedenen Gesetze und Verordnungen sicherzustellen, z. B. der DSGVO. Es obliegt allein den Kunden, sich von kompetenter juristischer Stelle zu Inhalt und Auslegung aller relevanten Gesetze und Vorschriften beraten zu lassen, die ihre Geschäftstätigkeit und die von ihnen eventuell einzuleitenden Maßnahmen zur Einhaltung dieser Gesetze und Vorschriften betreffen.

Die hier beschriebenen Produkte, Services und anderen Funktionen sind nicht für alle Kundensituationen geeignet und können eine eingeschränkte Verfügbarkeit haben. IBM gibt keine Rechts-, Buchhaltungs- oder Wirtschaftsprüfungsberatung und übernimmt keine Verantwortung bzw. bietet keine

Gewährleistung, dass seine Services und Produkte die Einhaltung von Gesetzen oder Verordnungen durch den Kunden sicherstellen.

Inhaltsverzeichnis

DSGVO

Die Datenschutz-Grundverordnung (DSGVO) wurde von der Europäischen Union ("EU") verabschiedet und gilt seit dem 25. Mai 2018.

Warum ist die DSGVO wichtig?

Mit der DSGVO wird ein strengerer datenschutzrechtlicher Rahmen für die Verarbeitung personenbezogener Daten geschaffen. Folgende Änderungen sieht die DSGVO vor:

- Neue und erweiterte Rechte für Einzelpersonen
- Eine erweiterte Definition von personenbezogenen Daten
- Neue Verpflichtungen für "Auftragsverarbeiter"
- Potenzielle finanzielle Sanktionen bei Nichtkonformität
- Zwingende Meldepflicht bei Datenschutzverletzungen

Weitere Informationen zur DSGVO finden Sie hier:

- Informationsportal zur EU-Datenschutz-Grundverordnung
- ibm.com/GDPR-Website

Produktkonfiguration -Überlegungen zur DSGVO-Bereitschaft

In den folgenden Abschnitten finden Sie Hinweise zur Konfiguration von IBM MQ, um Ihre Organisation bei der Umsetzung der DSGVO zu unterstützen.

Datenlebenszyklus

IBM MQ ist ein auf dem transaktionsorientierten Nachrichtenaustausch basierendes Middlewareprodukt, das Anwendungen den asynchronen Austausch von Daten ermöglicht, die von Anwendungen bereitgestellt werden. IBM MQ unterstützt eine Reihe von Messaging-APIs, -Protokollen und -Bridges, um Verbindungen zwischen den Anwendungen zu ermöglichen. Daher kann IBM MQ zum Austausch vielfältiger Formen von Daten verwendet werden, die potenziell der DSGVO unterliegen können. Es gibt auch verschiedene Produkte von Drittanbietern, mit denen IBM MQ Daten austauschen kann. Einige dieser Produkte gehören IBM, doch viele andere Produkte werden von anderen Technologieunternehmen bereitgestellt. Auf der Website Software Product Compatibility Reports sind Listen der zugehörigen Software aufgeführt. Für Überlegungen zur DSGVO-Bereitschaft eines Fremdprodukts sollten Sie die Produktdokumentation zu

Rate ziehen. IBM MQ-Administratoren steuern über die Definition von Warteschlangen, Themen und Subskriptionen die Art und Weise, in der IBM MQ mit den Daten interagiert, die das Programm durchlaufen.

Welche Datentypen durchlaufen IBM MQ?

Da IBM MQ Messaging-Services für die asynchrone Übertragung von Anwendungsdaten bereitstellt, gibt es keine definitive Antwort auf diese Farge, weil die Anwendungsfälle von den jeweils bereitgestellten Anwendungen abhängt. Anwendungsnachrichtendaten werden in Warteschlangendateien (Seitengruppen oder Coupling-Facility unter z/OS), Protokollen und Archiven gespeichert, und die Nachricht selbst kann Daten enthalten, die von GDPR gesteuert werden. Die von den Anwendungen bereitgestellten Nachrichtendaten können auch in Dateien enthalten sein, die zu Fehlerbestimmungszwecken erfasst werden, wie z. B. Fehlerprotokolle, Tracedateien und FFSTs. Unter z/OS können die von den Anwendungen bereitgestellten Nachrichtendaten auch in Speicherauszügen des Adressraums oder der Coupling-Facility enthalten sein.

Es folgen einige typische Beispiele für personenbezogene Daten, die mithilfe von IBM MQ zwischen Anwendungen ausgetauscht werden können:

- Personenbezogene Daten der Mitarbeiter des IBM Kunden (z. B. wenn mit IBM MQ eine Verbindung zum Lohnbuchhaltungs- oder Personalabteilungssystem des IBM Kunden hergestellt wird)
- Personenbezogene Daten der Kunden des IBM Kunden (z. B. wenn ein IBM Kunde die Daten seiner eigenen Kunden mit IBM MQ zwischen Anwendungen austauscht, wie z. B. Anfragen von Kaufinteressenten und die im CRM-System gespeicherten Kundendaten)
- Sensible personenbezogene Daten der Kunden des IBM Kunden (z. B. wenn IBM MQ in branchenspezifischen Kontexten eingesetzt wird, in denen spezielle personenbezogene Daten ausgetauscht werden müssen, wie z. B. beim Austausch von Patientenakten nach dem HL7-Protokoll zwischen medizintechnischen Anwendungen.

Zusätzlich zu den von Anwendungen bereitgestellten Nachrichtendaten verarbeitet IBM MQ auch folgende Datentypen:

- Authentifizierungsnachweise (z. B. Benutzername und Kennwort, API-Schlüssel usw.)
- Technisch identifizierbare personenbezogen Daten (z. B. Einheiten-IDs, Nutzungskennungen, IP-Adresse usw. bei Verknüpfung mit einer Einzelperson)

Personenbezogene Daten, die für den Onlinekontakt mit IBM verwendet werden

IBM MQ Kunden können Kommentare/Feedback/Anfragen online einreichen, um IBM über IBM MQ Themen auf verschiedene Arten zu kontaktieren, in erster Linie:

- Öffentlicher Kommentarbereich auf Seiten im IBM MQ-Bereich auf IBM Developer
- Bereich der öffentlichen Kommentare auf Seiten von IBM MQ-Produktinformationen in IBM Documentation
- Öffentliche Kommentare in den IBM Unterstützungsforen
- Öffentliche Kommentare in IBM-Integrationsideen

In der Regel werden nur der Name und die E-Mail-Adresse des Kunden verwendet, um persönliche Antworten für den Betreff des Kontakts zu ermöglichen, und die Verwendung personenbezogener Daten entspricht der IBM Online-Datenschutzerklärung.

Datenerfassung

IBM MQ kann zum erfassen von personenbezogenen Daten verwendet werden. Wenn Sie Ihre Verwendung von IBM MQ sowie Ihren Maßnahmenbedarf zur Einhaltung der DSGVO-Bestimmungen überprüfen, sollten Sie die Arten personenbezogener Daten berücksichtigen, die in Ihrem Fall IBM MQ durchlaufen. Sie können z. B. Aspekte wie die folgenden berücksichtigen:

- Wie kommen die Daten zu Ihren Warteschlangenmanagern? (Über welche Protokolle? Sind die Daten verschlüsselt? Sind die Daten signiert?)
- Wie werden Daten von Ihren Warteschlangenmanagern gesendet? (Über welche Protokolle? Sind die Daten verschlüsselt? Sind die Daten signiert?)

- Wie werden Daten beim Durchlaufen eines Warteschlangenmanagers gespeichert? (Jede Messaging-Anwendung hat das Potenzial, Nachrichtendaten in statusabhängige Medien zu schreiben, selbst wenn eine Nachricht nicht persistent ist. Sind Sie sich bewusst, wie die Messaging-Funktionen Aspekte der Anwendungsnachrichtendaten, die das Produkt passieren, zugänglich machen könnten?)
- Wie werden die Berechtigungsnachweise erfasst und ggf. gespeichert, wenn sie von IBM MQ für den Zugriff auf Drittanbieteranwendungen benötigt werden?

IBM MQ muss möglicherweise mit anderen Systemen und Services kommunizieren, für die eine Authentifizierung erforderlich ist, z. B. LDAP. Bei Bedarf werden Authentifizierungsdaten (Benutzer-IDs, Kennwörter) konfiguriert und von IBM MQ für die Verwendung in dieser Kommunikation gespeichert. Wenn möglich, sollten Sie die Verwendung persönlicher Berechtigungsnachweise für die IBM MQ-Authentifizierung vermeiden. Berücksichtigen Sie den Schutz des Speichers, der für Authentifizierungsdaten verwendet wird. (Siehe Datenspeicherung weiter unten.)

Datenspeicherung

Wenn Nachrichtendaten über Warteschlangenmanager übertragen werden, wird IBM MQ die Daten (und möglicherweise mehrere Kopien davon) direkt dauerhaft auf statusabhängige Datenträger speichern. IBM MQ-Benutzer sollten die Nachrichtendaten lieber sichern, während sie sich in Ruhe befinden.

In den folgenden Punkten werden Bereiche hervorgehoben, in denen IBM MQ die von Anwendungen bereitgestellten Daten dauerhaft speichert und die deshalb bei Überlegungen zur Einhaltung der DSGVO-Bestimmungen von den Benutzern sorgfältig bedacht werden sollten.

- Anwendungsnachrichtenwarteschlangen:

  IBM MQ stellt Nachrichtenwarteschlangen bereit, um den asynchronen Datenaustausch zwischen Anwendungen zu ermöglichen. Nicht persistente und persistente Nachrichten, die in einer Warteschlange gespeichert sind, werden in statusabhängige Datenträger geschrieben.
- Dateiübertragungsagentenwarteschlangen:

  Um die zuverlässige Übertragung von Dateidaten zu koordinieren, verwendet IBM MQ Managed File Transfer Nachrichtenwarteschlangen, in denen Dateien, die personenbezogene Daten enthalten, und Datensätze von Übertragungen gespeichert sind.
- Übertragungswarteschlangen:

  Damit Nachrichten zuverlässig zwischen Warteschlangenmanagern übertragen werden können, werden Nachrichten temporär in Übertragungswarteschlangen gespeichert.
- Warteschlangen für nicht zustellbare Nachrichten:

  Es gibt Situationen, in denen Nachrichten nicht in eine Zielwarteschlange gestellt werden können und in einer Warteschlange für dead-Mail gespeichert werden, wenn eine solche Warteschlange auf dem Warteschlangenmanager konfiguriert ist.
- Rücksetzwarteschlangen:

  JMS-und XMS-Messaging-Schnittstellen bieten die Möglichkeit, falsch formatierte Nachrichten nach einer Reihe von Rücksetzungen in eine Rücksetzwarteschlange zu verschieben, damit andere gültige Nachrichten verarbeitet werden können.
- AMS-Fehlerwarteschlange:

  IBM MQ Advanced Message Security verschiebt Nachrichten, die einer Sicherheitsrichtlinie nicht entsprechen, in SYSTEM.PROTECTION.ERROR.QUEUE Fehlerwarteschlange auf ähnliche Weise wie die Warteschlange für nicht zustellbare Nachrichten.
- Ständige Veröffentlichungen:

  IBM MQ stellt eine Funktion für ständige Veröffentlichungen bereit, mit subskribierende Anwendungen vorherige Veröffentlichungen erneut abrufen können.
- Verzögerte Zustellung:

IBM MQ unterstützt die Zustellungsverzögerungsfunktion JMS 2.0 und Jakarta Messaging 3.0 , die die Zustellung von Nachrichten an ihr Ziel zu einem späteren Zeitpunkt ermöglicht. Nachrichten, die noch nicht zugestellt wurden, werden in der Warteschlange SYSTEM.DDELAY.LOCAL.QUEUE gespeichert.

Weitere Informationen finden Sie hier:

- Protokollierung: Stellen Sie sicher, dass die Nachrichten nicht verloren gehen.
- MFT Agent-Warteschlangeneinstellungen
- Verwenden der Warteschlange für dead-Mail
- Behandlung von Giftnachrichten in IBM MQ-Klassen für JMS
- AMS-Fehlerbehandlung
- Zurückgehaltene Veröffentlichungen
- JMS 2.0 -Zustellungsverzögerung

In den folgenden Punkten werden Bereiche hervorgehoben, in denen IBM MQ die von Anwendungen bereitgestellten Daten eventuell indirekt dauerhaft speichert und die deshalb bei Überlegungen zur Einhaltung der DSGVO-Bestimmungen von den Benutzern ebenfalls sorgfältig bedacht werden sollten.

- Trace-Route-Messaging:

  IBM MQ stellt Trace-Route-Funktionen zur Aufzeichnung der Route bereit, die eine Nachricht zwischen den Anwendungen durchläuft. Die generierten Ereignisnachrichten können technisch identifizierbare personenbezogene Daten, wie z. B. IP-Adressen, enthalten.

- Anwendungsaktivitätstrace:

  IBM MQstellt einen Anwendungsaktivitätstrace bereit, der die Messaging-API-Aktivitäten von Anwendungen und Kanälen aufzeichnet. Der Anwendungsaktivitätstrace kann den Inhalt der von Anwendungen bereitgestellten Nachrichtendaten in Ereignisnachrichten speichern.

- Service-Trace:

  IBM MQ stellt Service-Tracefunktionen bereit, mit denen die internen Codepfade aufgezeichnet werden, die die Nachrichtendatenflüsse durchlaufen. Im Rahmen dieser Funktionen kann IBM MQ den Inhalt der von Anwendungen bereitgestellten Nachrichtendaten in Tracedateien auf einem Datenträger speichern.

- Warteschlangenmanagerereignisse:

  IBM MQ kann Ereignisnachrichten generieren, die personenbezogene Daten enthalten können, z. B. Berechtigungs-, Befehls- und Konfigurationsereignisse.

Weitere Informationen finden Sie hier:

- Trace-Route-Messaging
- Trace verwenden
- Ereignisüberwachung
- WS-Manager-Ereignisse

Wenn Sie den Zugriff auf Kopien der von den Anwendungen bereitgestellten Nachrichtendaten schützen möchten, können Sie die folgende Aktionen in Betracht ziehen:

- Beschränken Sie den Zugriff privilegierter Benutzer auf IBM MQ-Daten im Dateisystem, z. B. indem Sie auf UNIX and Linux®-Plattformen die Benutzerzugehörigkeit zur Gruppe 'mqm' beschränken.
- Beschränken Sie den Anwendungszugriff auf IBM MQ-Daten mithilfe von dedizierten Warteschlangen und Zugriffssteuerung. Vermeiden Sie, wenn möglich, eine unnötige gemeinsame Nutzung von Ressourcen, wie z. B. Warteschlangen, durch mehrere Anwendungen und sorgen Sie für eine differenzierte Zugriffssteuerung für Warteschlangen- und Themenressourcen.
- Schränken Sie den Zugriff auf replizierte Kopien von IBM MQ-Daten in Hochverfügbarkeits-(HA-) oder Disaster-Recovery-(DR-)Konfigurationen ein und sichern Sie die für die Replikation verwendeten Verbindungen.
- Verwenden Sie IBM MQ Advanced Message Security für die Bereitstellung einer durchgängigen Signierung und/oder Verschlüsselung der Nachrichtendaten.

- Verwenden Sie die Verschlüsselung auf Datei-oder Datenträgerebene, um Verzeichnisse oder Dateisysteme zu schützen, die IBM MQ -Daten, -Traces oder -Protokolle enthalten könnten.
- Nachdem Sie einen Service-Trace an IBM hochgeladen haben, können Sie Ihre Service-Tracedateien und FFST-Daten löschen, wenn Sie besorgt sind, dass diese möglicherweise personenbezogene Daten enthalten.

Weitere Informationen finden Sie hier:

- Privilegierte Benutzer
- Unterstützung von Dateisystemen auf Multiplatforms planen
- Dateisystemverschlüsselung auf dem IBM MQ Appliance

Ein IBM MQ-Administrator kann einen Warteschlangenmanager mit Berechtigungsnachweisen (Benutzername und Kennwort, API-Schlüssel usw.) konfigurieren für Services anderer Anbieter wie LDAP. Diese Daten werden in der Regel im Datenverzeichnis des Warteschlangenmanagers gespeichert, das durch Dateisystemberechtigungen geschützt ist.

Beim Erstellen eines IBM MQ-Warteschlangenmanagers wird dessen Datenverzeichnis mit gruppenbasierter Zugriffssteuerung eingerichtet, damit IBM MQ die Konfigurationsdateien lesen und die Berechtigungsnachweise für Verbindungen zu diesen Systemen verwenden kann. Da IBM MQ-Administratoren als privilegierte Benutzer betrachtet werden und Mitglieder dieser Gruppe sind, haben sie Lesezugriff auf diese Dateien. Einige Dateien sind verschleiert, aber sie sind nicht verschlüsselt. Aus diesem Grund sollten Sie die folgenden Aktionen in Betracht ziehen, um den Zugriff auf Berechtigungsnachweise vollständig zu schützen:

- Beschränken Sie den Zugriff privilegierter Benutzer auf IBM MQ-Daten, z. B. indem Sie auf UNIX and Linux-Plattformen die Zugehörigkeit zur Gruppe 'mqm' beschränken.
- Verwenden Sie die Verschlüsselung auf Datei- oder Datenträgerebene, um den Inhalt des Datenverzeichnisses des Warteschlangenmanagers zu schützen.
- Verschlüsseln Sie die Backups des Produktionskonfigurationsverzeichnisses und speichern Sie sie mit entsprechenden Zugriffssteuerungen.
- Sie können Prüfprotokolle für Authentifizierungsfehler, Zugriffssteuerung und Konfigurationsänderungen mit Sicherheits-, Befehls-und Konfigurationsereignissen bereitstellen.

Weitere Informationen finden Sie hier:

- IBM MQ schützen

Datenzugriff

Die folgenden Produktschnittstellen können auf die Daten von IBM MQ-Warteschlangenmanagern zugreifen, wobei einige für den Zugriff über eine ferne Verbindung und andere für den Zugriff über eine lokale Verbindung konfiguriert sind.

- IBM MQ-Konsole [nur fern]
- IBM MQ-Administrative-REST-API [nur fern]
- IBM MQ-Messaging-REST-API [nur fern]
- MQI [Lokal und Fern]
- JMS [Lokal und Fern]
- XMS [Lokal und Fern]
- IBM MQ Telemetry (MQTT) [nur fern]
- IBM MQ Light (AMQP) [Nur Remote]
- IBM MQ IMS-Brücke [nur lokal]
- IBM MQ CICS-Bridge [nur lokal]
- IBM MQ MFT Protokollbridges [nur fern]
- IBM MQ Connect:Direct-Bridges [nur fern]

- IBM MQ MQAI [lokal und fern]
- IBM MQ-PCF-Befehle [lokal und fern]
- IBM MQ-MQSC-Befehle [lokal und fern]
- IBM MQ Explorer [lokal und fern]
- IBM MQ-Benutzerexits [nur lokal]
- IBM MQ Internet Pass-Thru [nur fern]
- Metriken für Red Hat® OpenShift® Monitoring (Prometheus) (die Metriken sind numerische Daten zu Statistiken der Warteschlangenmanager)
- IBM MQ Appliance - serielle Konsole [nur lokal]
- IBM MQ Appliance-SSH [nur fern]
- IBM MQ Appliance-REST-API [nur fern]
- IBM MQ Appliance-Webbenutzerschnittstelle [nur fern]
- V 9.4.0 IBM MQ Kafka Connectors (Kafka Connect) [lokal und fern]

Die Schnittstellen sind so konzipiert, dass Benutzer Änderungen an einem IBM MQ-Warteschlangenmanager und den darin gespeicherten Nachrichten vornehmen können. Die Verwaltungs- und Messaging-Operationen sind so gesichert, dass es drei Stufen gibt, wenn eine Anforderung gestellt wird.

- Authentifizierung
- Rollenzuordnung
- Autorisierung

Authentifizierung:

Wenn die Nachricht oder die Verwaltungsoperation von einer lokalen Verbindung angefordert wurde, ist die Quelle dieser Verbindung ein laufender Prozess auf demselben System. Der Benutzer, der den Prozess ausführt, muss alle vom Betriebssystem zur Verfügung gestellten Authentifizierungsschritte durchlaufen haben. Der Benutzername des Eigners des Prozesses, von dem aus die Verbindung hergestellt wurde, wird als Identität bestätigt. Dies könnte z. B. der Name des Benutzers sein, der die Shell ausführt, von der eine Anwendung gestartet wurde. Die möglichen Formen der Authentifizierung für lokale Verbindungen sind:

1. Bestätigter Benutzername (lokales Betriebssystem)
2. Optionaler Benutzername und Kennwort (OS, LDAP oder benutzerdefinierte Repositorys von Drittanbietern)
3. Sicherheitstoken (nur JWT) IBM MQ

Wenn die Verwaltungsaktion von einer fernen Verbindung angefordert wurde, wird die Kommunikation mit IBM MQ über eine Netzschnittstelle ausgeführt. Die folgenden Formen der Identität können für die Authentifizierung über Netzwerkverbindungen dargestellt werden:

1. Bestätigter Benutzername (vom fernen Betriebssystem)
2. Benutzername und Kennwort (Betriebssystem, LDAP oder benutzerdefinierte Repositorys von Drittanbietern)
3. Quellennetzadresse (z. B. IP-Adresse)
4. Digitales X.509-Zertifikat (gegenseitige SSL/TLS-Authentifizierung)
5. Sicherheitstoken (z. B. LTPA2 -Token oder JWT-Token)
6. Andere benutzerdefinierte Sicherheit (Funktionalität, die von Drittanbieterexits bereitgestellt wird)
7. SSH-Schlüssel

Die Integration von IBM MQmit IBM Cloud Pak for Integration fügt einen neuen Authentifizierungstyp für IBM MQ Consolehinzu: Single Sign-on mit Cloud Pak. (Nur CP4I)

Rollenzuordnung:

In der Rollenzuordnungsstufe können die Berechtigungsnachweise, die in der Authentifizierungsstufe bereitgestellt wurden, einer alternativen Benutzer-ID zugeordnet werden. Wenn der zugeordneten Benutzer-ID die Erlaubnis zum Fortfahren erteilt wird (z. B. können Benutzer mit Verwaltungsaufgaben durch Kanalauthentifizierungsregeln blockiert werden), wird die zugeordnete Benutzer-ID an die finale Stufe weitergeleitet, in der Aktivitäten für IBM MQ-Ressourcen autorisiert werden.

Autorisierung:

IBM MQ bietet die Möglichkeit, verschiedenen Benutzern für verschiedene Messaging-Ressourcen, z. B. Warteschlangen, Themen und andere Warteschlangenmanagerobjekte, unterschiedliche Berechtigungen zuzuweisen.

Protokollierungsaktivität:

Einige Benutzer von IBM MQ müssen möglicherweise einen Prüfsatz erstellen, um auf MQ-Ressourcen zugreifen zu können. Beispiele für wünschenswerte Prüfprotokolle können Konfigurationsänderungen enthalten, die Informationen über die Änderung enthalten, die zusätzlich zu den angeforderten Änderungen enthalten sind.

Für die Implementierung dieser Anforderung stehen die folgenden Informationsquellen zur Verfügung:

1. Ein IBM MQ-Warteschlangenmanager kann so konfiguriert werden, dass er ein Befehlsereignis generiert, wenn ein Verwaltungsbefehl erfolgreich ausgeführt wurde.
2. Ein IBM MQ-Warteschlangenmanager kann so konfiguriert werden, dass er Konfigurationsereignisse generiert, wenn eine Warteschlangenmanagerressource erstellt, geändert oder gelöscht wird.
3. Ein IBM MQ-Warteschlangenmanager kann so konfiguriert werden, dass er ein Berechtigungsereignis generiert, wenn eine Berechtigungsprüfung für eine Ressource fehlschlägt.
4. Fehlermeldungen, die auf fehlgeschlagene Berechtigungsprüfungen hinweisen, werden in die Fehlerprotokolle des Warteschlangenmanagers geschrieben.
5. Die IBM MQ-Konsole schreibt Prüfnachrichten in ihre Protokolle, wenn Authentifizierungs- bzw. Berechtigungsprüfungen fehlschlagen oder wenn Warteschlangenmanager erstellt, gestartet, gestoppt oder gelöscht werden.
6. Die IBM MQ Appliance schreibt Prüfnachrichten in ihre Protokolle, um Benutzeranmeldungen und Systemänderungen aufzuzeichnen.

Wenn diese Lösungsmöglichkeiten in Betracht gezogen werden, sollten die IBM MQ-Benutzer folgende Punkte berücksichtigen:

- Ereignisnachrichten sind nicht persistent, so dass beim erneuten Starten eines Warteschlangenmanagers die Informationen verloren gehen. Alle Ereignismonitore sollten so konfiguriert werden, dass sie ständig alle verfügbaren Nachrichten konsumieren und den Inhalt auf persistente Datenträger übertragen.
- Privilegierte IBM MQ-Benutzer haben ausreichende Berechtigungen, um Ereignisse zu deaktivieren, den Inhalt von Protokollen zu löschen oder Warteschlangenmanager zu löschen.

Weitere Informationen zur Sicherung des Zugriffs auf IBM MQ-Daten und Bereitstellung eines Prüfprotokolls finden Sie in folgenden Abschnitten:

- IBM MQ-Sicherheitsmechanismen
- Konfigurationsereignisse
- Befehlsereignisse
- Fehlerprotokolle verwenden

Datenverarbeitung

Verschlüsselung mit einer PKI-Infrastruktur (Public Key Infrastructure):

Sie können Netzverbindungen zu IBM MQ sichern, indem Sie angeben, dass die Verbindungen TLS verwenden, die auch die gegenseitige Authentifizierung der einleitenden Seite der Verbindung bereitstellen können.

Die Verwendung der PKI-Sicherheitseinrichtungen, die durch Transportmechanismen bereitgestellt werden, ist der erste Schritt, um die Datenverarbeitung mit IBM MQ zu sichern. Ohne weitere Sicherheitsfunktionen zu aktivieren, besteht das Verhalten einer konsumierenden Anwendung jedoch darin, alle Nachrichten zu verarbeiten, die an sie übermittelt wurden, ohne zu überprüfen, wo der Ursprung der Nachricht ist oder ob die Nachricht während der Übertragung geändert wurde.

IBM MQ-Benutzer, die für die Verwendung von AMS-Funktionen (Advanced Message Security) lizenziert sind, können durch die Definition und Konfiguration von Sicherheitsrichtlinien steuern, wie die in Nachrichten enthaltenen personenbezogenen Daten von Anwendungen verarbeitet werden. Sicherheitsrichtlinien ermöglichen es, dass digitale Signatur und/oder Verschlüsselung auf Nachrichtendaten zwischen Anwendungen angewendet werden können.

Es ist möglich, Sicherheitsrichtlinien zu verwenden, um eine digitale Signatur zu fordern und zu validieren, wenn Nachrichten konsumiert werden, um sicherzustellen, dass Nachrichten authentisch sind. Die AMS-Verschlüsselung stellt eine Methode zur Verfügung, mit der Nachrichtendaten von einem lesbaren Formular in eine verschlüsselte Version konvertiert werden, die nur von einer anderen Anwendung decodiert werden kann, wenn es sich um den beabsichtigten Empfänger oder die Nachricht handelt und Zugriff auf den richtigen Entschlüsselungsschlüssel hat.

Weitere Informationen zur Verwendung von SSL und Zertifikaten zum Sichern Ihrer Netzverbindungen finden Sie in den folgenden Abschnitten in der Produktdokumentation zu IBM MQ:

- TLS-Sicherheit für IBM MQ konfigurieren
- AMS-Übersicht

Datenlöschung

IBM MQ stellt Befehle und Benutzerschnittstellenaktionen zur Verfügung, mit denen Daten gelöscht werden, die dem Produkt bereitgestellt wurden. Das bedeutet, dass Benutzer von IBM MQ Daten löschen können, die sich auf bestimmte Personen beziehen, falls dies erforderlich sein sollte.

- Bereiche des IBM MQ-Verhaltens, die für die Einhaltung der DSGVO-Bestimmungen in Bezug auf das Löschen von Kundendaten bedacht werden sollten
  - Löschen von Nachrichtendaten, die in einer Anwendungswarteschlange gespeichert sind, durch:
    - Einzelne Nachrichten unter Verwendung der Messaging-API oder -Tools oder unter Verwendung von Nachrichtenverfallszeit entfernen
    - Angeben, dass Nachrichten nicht persistent sind, in einer Warteschlange gehalten werden, in der die nicht persistente Nachrichtenklasse normal ist und der Warteschlangenmanager erneut gestartet wird.
    - Die Warteschlange wird administrativ gelöscht.
    - Die Warteschlange wird gelöscht.
  - Gespeicherter Veröffentlichungsdaten, die in einem Thema gespeichert sind, löschen von:
    - Angeben, dass Nachrichten nicht persistent sind und den Warteschlangenmanager erneut starten.
    - Die aufbewahrten Daten durch neue Daten ersetzen oder die Nachrichtenablaufzeit verwenden.
    - Die Themenzeichenfolge wird administrativ gelöscht.
  - Löschen Sie auf einem Warteschlangenmanager gespeicherte Daten, indem Sie den gesamten Warteschlangenmanager und alle replizierten Kopien für die Hochverfügbarkeit oder Disaster-Recovery löschen.
  - Löschen Sie die Daten, die von den Service-Trace-Befehlen gespeichert werden, indem Sie die Dateien im Traceverzeichnis löschen.
  - Löschen Sie FFST-Daten, die gespeichert werden, indem Sie die Dateien im Fehlerverzeichnis löschen.
  - Löschen Sie die Speicherauszüge des Adressraums und der Coupling-Facility (unter z/OS).
  - Löschen Sie Archiv-, Backup-oder andere Kopien dieser Daten.

- Bereiche des IBM MQ-Verhaltens, die für die Einhaltung der DSGVO-Bestimmungen in Bezug auf das Löschen von Benutzeraccountdaten bedacht werden sollten
  - Sie können Benutzeraccountdaten und Vorgaben löschen, die von IBM MQ zum Herstellen von Verbindungen zu Warteschlangenmanagern und Drittanbieterservices gespeichert werden, indem Sie Folgendes löschen (einschließlich der Archiv- und Sicherungsdateien sowie anderweitig replizierter Kopien davon):
    - Authentifizierungsdaten des Warteschlangenmanagers, die Berechtigungsnachweise speichern.
    - WS-Manager-Berechtigungsdatensätze, die auf Benutzer-IDs verweisen.
    - WS-Manager-Kanalauthentifizierungsregeln, die bestimmte IP-Adressen, DNs oder Benutzer-IDs von Zertifikaten zuordnen oder blockieren.
    - Berechtigungsnachweisdateien, die von IBM MQ Managed File Transfer Agent, Logger und dem MFT-Plug-in für MQ Explorer für die Authentifizierung bei Warteschlangenmanager- und Dateiservern verwendet werden.
    - Digitale X.509-Zertifikate, die aus Keystores stammende Informationen zu einer Einzelperson darstellen oder enthalten, die von SSL/TLS-Verbindungen oder IBM MQ Advanced Message Security (AMS) verwendet werden.
    - Einzelne Benutzeraccounts aus IBM MQ Appliance, einschließlich des Verweises auf diese Accounts in Systemprotokolldateien.
    - Metadaten des Arbeitsbereichs von IBM MQ Explorer und Einstellungen für Eclipse
    - Kennwortspeicher von IBM MQ Explorer, wie im Abschnitt Password Preferences (Kennwortvorgaben) beschrieben.
    - Konfigurationsdateien für die IBM MQ-Konsole und den mqweb-Server.
    - IBM MQ Internet Pass-Thru-Konfigurationsdateien und -Keystores.

Weitere Informationen finden Sie hier:

- MFT-und IBM MQ-Verbindungsauthentifizierung
- Berechtigungsnachweise für einen Dateiserver mithilfe der Datei "ProtocolBridgeCredentials.xml" zuordnen
- IBM MQ Console -Benutzer und -Rollen konfigurieren

Datenüberwachung

IBM MQ stellt eine Reihe von Überwachungsfunktionen bereit, mit denen Benutzer die Leistung von Anwendungen und Warteschlangenmanagern besser überwachen können.

Außerdem stellt IBM MQ einige Funktionen zur Verwaltung von Fehlerprotokollen von Warteschlangenmanagern bereit.

Weitere Informationen finden Sie hier:

- IBM MQ-Netz überwachen
- Diagnosenachrichtenservices
- QMErrorLog-Service
- IBM MQ Appliance - Überwachung und Berichterstellung

Funktionalität für die Einschränkung der Verwendung von persönlichen Daten

Mithilfe der in diesem Dokument zusammengefassten Funktionen ermöglicht IBM MQ den Endbenutzern, die Verwendung ihrer personenbezogenen Daten zu beschränken.

IBM MQ-Nachrichtenwarteschlangen sollten nicht in derselben Weise wie eine Datenbank als permanenter Datenspeicher verwendet werden, was insbesondere bei der Verarbeitung von Anwendungsdaten zutrifft, die der DSGVO unterliegen.

Im Gegensatz zu einer Datenbank, in der Daten über eine Suchabfrage gefunden werden können, kann es schwierig sein, Nachrichtendaten zu finden, es sei denn, Sie kennen die Warteschlangen-, Nachrichten- und Korrelations-IDs einer Nachricht.

Wenn Nachrichten, die Daten einer bestimmten Einzelperson enthalten, leicht identifiziert und aufgefunden werden können, ist es mithilfe der standardmäßigen IBM MQ-Messaging-Funktionen möglich, auf die Nachrichtendaten zuzugreifen und sie zu bearbeiten.

Dateiverwaltung

1. IBM MQ Managed File Transfer führt keine Malware-Scans auf Dateien aus, die übertragen werden. Dateien werden übertragen, und es wird eine Integritätsprüfung durchgeführt, um sicherzustellen, dass die Dateidaten während der Übertragung nicht geändert werden. Die Quell- und Ziel-Prüfsummen werden als Teil der Übertragungsstatus-Publikation veröffentlicht. Es wird empfohlen, dass Endbenutzer geeignete Malware-Scans für ihre Umgebung implementieren, bevor MFT die Datei überträgt und nachdem MFT eine Datei an einen fernen Endpunkt übergibt.
2. IBM MQ Managed File Transfer führt keine Aktionen auf der Basis des MIME-Typs bzw. der Dateierweiterung durch. MFT liest die Datei und überträgt die Bytes genau wie aus der Eingabedatei gelesen.

Architekturen auf der Basis eines einzelnen Warteschlangenmanagers

Zu den einfachsten IBM MQ-Architekturen gehören solche, für die nur ein einzelner Warteschlangenmanager konfiguriert und verwendet wird.

Bevor Sie mit der Planung einer IBM MQ-Architektur beginnen, sollten Sie sich mit den grundlegenden IBM MQ-Konzepten vertraut machen. Siehe IBM MQ Technische Übersicht.

Eine Reihe möglicher Architekturen mit einem einzigen Warteschlangenmanager werden in den folgenden Abschnitten beschrieben:

- „Einzelwarteschlangenmanager mit lokalen Anwendungen, die auf einen Service zugreifen” auf Seite 19
- „Einzelnes Warteschlangenmanager mit fernen Anwendungen, die auf einen Service als Clients zugreifen” auf Seite 19
- „Einzelwarteschlangenmanager mit einer Publish/Subscribe-Konfiguration” auf Seite 20

Einzelwarteschlangenmanager mit lokalen Anwendungen, die auf einen Service zugreifen

Die erste Architektur auf der Basis eines einzigen Warteschlangenmanagers besteht darin, dass die Anwendungen, die auf einen Service zugreifen, auf demselben System ausgeführt werden wie die Anwendungen, die den Service bereitstellen. ein IBM MQ-Warteschlangenmanager stellt die asynchrone Kommunikation zwischen den Anwendungen, die den Service anfordern, und den Anwendungen, die den Service bereitstellen, bereit. Dies bedeutet, dass die Kommunikation zwischen den Anwendungen auch dann fortgesetzt werden kann, wenn eine der Anwendungen für einen längeren Zeitraum offline ist.

Einzelnes Warteschlangenmanager mit fernen Anwendungen, die auf einen Service als Clients zugreifen

Die zweite Architektur auf der Basis eines einzelnen Warteschlangenmanagers verfügt über die Anwendungen, die über Remotezugriff von den Anwendungen ausgeführt werden, die den Service bereitstellen. Die fernen Anwendungen werden auf verschiedenen Systemen für die Services ausgeführt. Die Anwendungen stellen eine Verbindung als Clients mit dem einzelnen Warteschlangenmanager her. Dies bedeutet, dass der Zugriff auf einen Service mehreren Systemen über einen einzigen Warteschlangenmanager zur Verfügung gestellt werden kann.

Eine Einschränkung dieser Architektur besteht darin, dass eine Netzverbindung verfügbar sein muss, damit eine Anwendung ausgeführt werden kann. Die Interaktion zwischen der Anwendung und dem WS-Manager über die Netzverbindung erfolgt synchron.

Einzelwarteschlangenmanager mit einer Publish/Subscribe-Konfiguration

Eine alternative Architektur, die einen einzelnen WS-Manager verwendet, ist die Verwendung einer Publish/Subscribe-Konfiguration. Beim Publish/Subscribe-Messaging können Sie den Anbieter von Informationen von den Konsumenten dieser Informationen entkoppeln. Dies unterscheidet sich vom Punkt-zu-Punkt-Messaging-Stile in den zuvor beschriebenen Architekturen, wo die Anwendungen Informationen über die Zielanwendung kennen müssen, z. B. den Namen der Warteschlange, in die Nachrichten gestellt werden sollen. Über IBM MQ Publish/Subscribe veröffentlicht die sendende Anwendung ein bestimmtes Thema, das auf dem Inhalt der Informationen basiert. IBM MQ sorgt für die Verteilung der Nachricht an Anwendungen, die mithilfe einer Subskription ihr Interesse an diesem Inhalt angemeldet haben. Die empfangenden Anwendungen müssen außerdem nichts über die Quelle der Nachrichten wissen, um sie zu empfangen. Weitere Informationen finden Sie unter Publish/Subscribe-Nachrichtenübermittlung und Beispiel für eine Publish/Subscribe-Konfiguration eines einzelnen Warteschlangenmanagers.

Zugehörige Konzepte

Einführung in IBM MQ

Zugehörige Tasks

„IBM MQ-Architektur planen" auf Seite 5
Beachten Sie bei der Planung einer IBM MQ-Umgebung die Unterstützung, die IBM MQ für Architekturen mit einzelnen oder mehreren Warteschlangenmanagern sowie für Punkt-zu-Punkt- und Publish/Subscribe-Messaging bereitstellt. Planen Sie auch den Ressourcenbedarf und die Nutzung von Protokollierungs- und Sicherheitsfunktionen.

Warteschlangenmanager auf Multiplatforms erstellen und verwalten

Architekturen auf der Basis von mehreren Warteschlangenmanagern

Sie können Methoden zur Steuerung der Warteschlangen für verteilte Nachrichten verwenden, um eine IBM MQ-Architektur zu erstellen, die die Konfiguration und Verwendung mehrerer Warteschlangenemanager umfasst.

Bevor Sie mit der Planung einer IBM MQ-Architektur beginnen, sollten Sie sich mit den grundlegenden IBM MQ-Konzepten vertraut machen. Siehe IBM MQ Technische Übersicht.

Eine IBM MQ-Architektur kann geändert werden, ohne dass Änderungen an Anwendungen, die Services bereitstellen, erforderlich sind, indem zusätzliche Warteschlangenmanager hinzugefügt werden.

Anwendungen können auf derselben Maschine wie ein Warteschlangenmanager gehostet werden und dann asynchrone Kommunikation mit einem Service erhalten, der auf einem anderen Warteschlangenmanager auf einem anderen System gehostet wird. Alternativ können Anwendungen, die auf einen Service zugreifen, als Clients eine Verbindung zu einem Warteschlangenmanager herstellen, der dann den asynchronen Zugriff auf den Service auf einem anderen Warteschlangenmanager bereitstellt.

Routes, die verschiedene Warteschlangenmanager und ihre Warteschlangen verbinden, werden mithilfe von verteilten Warteschlangenverfahren definiert. Die Warteschlangenmanager in der Architektur werden über Kanäle miteinander verbunden. Kanäle werden verwendet, um Nachrichten automatisch von einem Warteschlangenmanager in eine andere Richtung in eine andere Richtung zu versetzen, abhängig von der Konfiguration der Warteschlangenmanager.

Eine Übersicht über die Planung eines IBM MQ-Netzes finden Sie unter „Entwerfen verteilter WS-Manager-Netze" auf Seite 22.

Informationen zur Planung von Kanälen für die IBM MQ-Architektur finden Sie unter IBM MQ - Methode zur verteilten Warteschlangensteuerung.

Über die verteilte Warteschlangenverwaltung können Sie die Kommunikation zwischen Warteschlangenmanagern erstellen und überwachen. Weitere Informationen zur verteilten Warteschlangenverwaltung finden Sie im Abschnitt Einführung in die verteilte Warteschlangenverwaltung .

Zugehörige Tasks

„IBM MQ-Architektur planen” auf Seite 5
Beachten Sie bei der Planung einer IBM MQ-Umgebung die Unterstützung, die IBM MQ für Architekturen mit einzelnen oder mehreren Warteschlangenmanagern sowie für Punkt-zu-Punkt- und Publish/Subscribe-Messaging bereitstellt. Planen Sie auch den Ressourcenbedarf und die Nutzung von Protokollierungs- und Sicherheitsfunktionen.

Warteschlangenmanager auf Multiplatforms erstellen und verwalten

Verteilte Warteschlangen und Cluster planen

Sie können Warteschlangen, die sich auf verteilten Warteschlangenmanagern befinden, manuell verbinden, oder Sie können einen WS-Manager-Cluster erstellen und das Produkt die Warteschlangenmanager für Sie verbinden. Um eine geeignete Topologie für Ihr verteiltes Messaging-Netzwerk auszuwählen, müssen Sie Ihre Anforderungen für die manuelle Steuerung, die Netzgröße, die Häufigkeit von Änderungen, die Verfügbarkeit und die Skalierbarkeit in Betracht ziehen.

Vorbereitende Schritte

In dieser Task wird davon ausgegangen, dass Sie wissen, welche verteilten Nachrichtenübertragungsnetze und wie sie funktionieren. Eine technische Übersicht finden Sie unter Verteilte Steuerung von Warteschlangen und Clustern.

Informationen zu diesem Vorgang

Wenn Sie ein verteiltes Messaging-Netz erstellen möchten, können Sie Kanäle manuell konfigurieren, um Warteschlangen zu verbinden, die sich auf verschiedenen Warteschlangenmanagern befinden, oder Sie können einen Warteschlangenmanager-Cluster erstellen. Durch das Clustering können WS-Manager miteinander kommunizieren, ohne zusätzliche Kanaldefinitionen oder ferne Warteschlangendefinitionen einzurichten, wodurch ihre Konfiguration und Verwaltung vereinfacht wird.

Wenn Sie eine geeignete Topologie für Ihr verteiltes Publish/Subscribe-Netz auswählen möchten, müssen Sie die folgenden allgemeinen Fragen berücksichtigen:

- Wie viel manuelle Kontrolle benötigen Sie über die Verbindungen in Ihrem Netzwerk?
- Wie groß wird Ihr Netzwerk sein?
- Wie dynamisch wird es sein?
- Was sind Ihre Verfügbarkeits-und Skalierbarkeitsanforderungen?

Prozedur

- Überlegen Sie, wie viel manuelle Steuerung Sie über die Verbindungen in Ihrem Netzwerk benötigen.

  Wenn Sie nur ein paar Verbindungen benötigen oder wenn einzelne Verbindungen sehr genau definiert werden müssen, sollten Sie das Netzwerk wahrscheinlich manuell erstellen.

  Wenn Sie mehrere Warteschlangenmanager benötigen, die logisch miteinander verknüpft sind und die Daten und Anwendungen gemeinsam nutzen müssen, sollten Sie sie in Betracht ziehen, sie in einem Warteschlangenmanager-Cluster zusammenzufassen.
- Schätzen Sie, wie groß Ihr Netzwerk sein muss.
  a) Schätzen Sie, wie viele Warteschlangenmanager Sie benötigen. Denken Sie daran, dass Warteschlangen in mehr als einem Warteschlangenmanager gehostet werden können.
  b) Wenn Sie einen Cluster verwenden möchten, fügen Sie zwei zusätzliche Warteschlangenmanager hinzu, um als vollständige Repositorys zu agieren.

Bei größeren Netzen kann die manuelle Konfiguration und Verwaltung von Verbindungen sehr zeitaufwendig sein, und Sie sollten in Betracht ziehen, einen Cluster zu verwenden.

- Überlegen Sie, wie dynamisch die Netzaktivität sein wird.

  Planen Sie, dass ausgelastete Warteschlangen auf performanten WS-Managern gehostet werden.

  Wenn Sie erwarten, dass Warteschlangen häufig erstellt und gelöscht werden, sollten Sie einen Cluster verwenden.

- Berücksichtigen Sie Ihre Verfügbarkeits-und Skalierbarkeits
  a) Entscheiden Sie, ob Sie die hohe Verfügbarkeit von Warteschlangenmanagern gewährleisten müssen. Ist dies der Fall, schätzen Sie die Anzahl der Warteschlangenmanager, für die diese Anforderung gilt, ab.
  b) Überlegen Sie, ob einige Ihrer WS-Manager weniger fähig sind als andere.
  c) Überlegen Sie, ob die Kommunikationsverbindungen zu einigen Ihrer WS-Manager empfindlicher als andere sind.
  d) Ziehen Sie das Hosting von Warteschlangen auf mehreren Warteschlangenmanagern in Betracht

  Manuell konfigurierte Netze und Cluster können so konfiguriert werden, dass sie hoch verfügbar und skalierbar sind. Wenn Sie einen Cluster verwenden, müssen Sie zwei zusätzliche WS-Manager als vollständige Repositorys definieren. Wenn zwei vollständige Repositorys vorhanden sind, wird sichergestellt, dass der Cluster weiter betrieben wird, wenn eines der vollständigen Repositorys nicht mehr verfügbar ist. Stellen Sie sicher, dass die vollständigen WS-Manager-Repositorys robust, leistungsfähig und eine gute Netzkonnektivität sind. Es ist nicht geplant, die vollständigen WS-Manager-Repositorys für andere Arbeiten zu verwenden.

- Basierend auf diesen Berechnungen können Sie mithilfe der bereitgestellten Links entscheiden, ob Verbindungen zwischen Warteschlangenmanagern manuell konfiguriert werden sollen oder ob ein Cluster verwendet werden soll.

Nächste Schritte

Sie können jetzt Ihr verteiltes Messaging-Netz konfigurieren.

Zugehörige Tasks

Verteilte Warteschlangensteuerung konfigurieren

WS-Manager-Cluster konfigurieren

Entwerfen verteilter WS-Manager-Netze

IBM MQ sendet und empfängt Daten, die mithilfe von Warteschlangenmanagern und Kanälen über Netze zwischen Anwendungen ausgetauscht werden. Die Netzplanung umfasst die Definition von Anforderungen zum Erstellen eines Frameworks für die Verbindung dieser Systeme über ein Netz.

Kanäle können zwischen Ihrem System und jedem anderen System, mit dem Sie Kommunikation benötigen, erstellt werden. Multi-Hop-Kanäle können erstellt werden, um eine Verbindung zu Systemen herzustellen, auf denen Sie keine direkten Verbindungen haben. Die in den Szenarios beschriebenen Nachrichtenkanalverbindungen werden in Abbildung 1 auf Seite 23 als Netzdiagramm dargestellt.

IBM MQ Internet Pass-Thru vereinfacht die Konfiguration von Kanälen zwischen Systemen in verschiedenen physischen Netzen sowie die Einrichtung von Kanälen, die über eine Firewall kommunizieren. Weitere Informationen finden Sie unter IBM MQ Internet Pass-Thru.

Namen der Kanal-und Übertragungswarteschlangen

Der Übertragungswarteschlange kann ein beliebiger Name gegeben werden. Um jedoch Unklarheiten zu vermeiden, können Sie ihnen dieselben Namen wie die Namen des Zielwarteschlangenmanagers oder die Aliasnamen des Warteschlangenmanagers geben. Dadurch wird die Übertragungswarteschlange der Route zugeordnet, die sie verwenden, und gibt einen klaren Überblick über parallele Routen, die über temporäre (mehrere-hackte) Warteschlangenmanager erstellt werden.

Es ist nicht so klar, dass die Kanalnamen abgeschnitten sind. Die Kanalnamen in Abbildung 1 auf Seite 23 für QM2 müssen sich beispielsweise für eingehende und abgehende Kanäle unterscheiden. Alle Kanalnamen können noch ihre Namen für die Übertragungswarteschlange enthalten, aber sie müssen qualifiziert sein, um sie eindeutig zu machen.

In WSM2 gibt es beispielsweise einen WSM3-Kanal von WSM1 und ein WSM3-Kanal zu QM3. Um die Namen eindeutig zu machen, kann der erste Name QM3_from_QM1 heißen und der zweite Name mit dem Namen QM3_von_QM2 benannt werden. Auf diese Weise zeigen die Kanalnamen den Namen der Übertragungswarteschlange im ersten Teil des Namens an. Die Richtung und der benachbarte WS-Manager-Name werden im zweiten Teil des Namens angezeigt.

Eine Tabelle mit den vorgeschlagenen Kanalnamen für Abbildung 1 auf Seite 23 finden Sie in Tabelle 1 auf Seite 23.

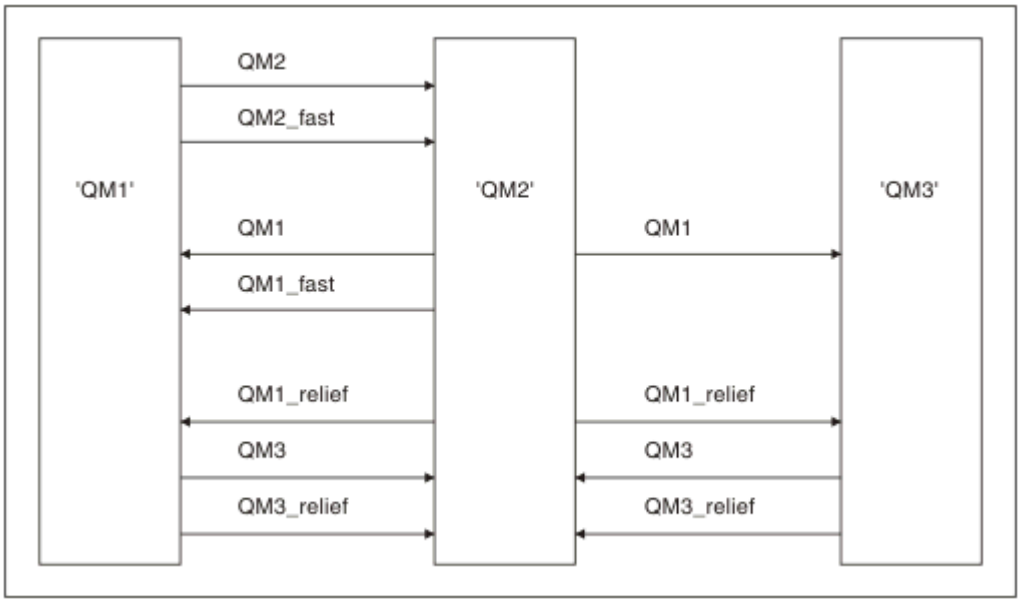


Abbildung 1. Netzdiagramm mit allen Kanälen

Tabelle 1. Beispiel für Kanalnamen

Leitwegname	**Kanal des Warteschlangenmanagers**	**Name der Übertragungswarteschlange**	**Empfohlene Kanalbezeichnung**
QM1	QM1 & QM2	WSM1 (bei WSM2)	QM1.from.QM2
QM1	QM2 & QM3	WSM1 (bei WSM3)	QM1.from.QM3
QM1_fast	QM1 & QM2	QM1_schnell (bei WSM2)	QM1_fast.from.QM2
QM1_relief	QM1 & QM2	QM1_relief (bei QM2)	QM1_relief.from.QM2
QM1_relief	QM2 & QM3	QM1_relief (bei QM3)	QM1_relief.from.QM3
QM2	QM1 & QM2	WSM2 (bei WSM1)	QM2.from.QM1
QM2_fast	QM1 & QM2	QM2_schnell (bei QM1)	QM2_fast.from.QM1
QM3	QM1 & QM2	WSM3 (bei WSM1)	QM3.from.QM1
QM3	QM2 & QM3	WSM3 (bei WSM2)	QM3.from.QM2
QM3_relief	QM1 & QM2	QM3_relief (bei QM1)	QM3_relief.from.QM1
QM3_relief	QM2 & QM3	QM3_relief (bei QM2)	QM3_relief.from.QM2

Anmerkung:

1. z/OS In IBM MQ for z/OS dürfen Warteschlangenmanagernamen nur vier Zeichen haben.
2. Nennen Sie alle Kanäle in Ihrem Netzwerk eindeutig. Wie in Tabelle 1 auf Seite 23 gezeigt, ist dies eine gute Möglichkeit, die Namen der Quellen-und Zielwarteschlangenmanager in den Kanalnamen zu verwenden.

Netzplaner

Bei der Erstellung eines Netzes wird davon ausgegangen, dass eine andere, übergeordnete Funktion von *network planner* vorhanden ist, deren Pläne von den anderen Mitgliedern des Teams implementiert werden.

Für weit verbreitete Anwendungen ist es ökonomischer, in Bezug auf lokale Zugriffsseiten für die Konzentration des Nachrichtenverkehrs zu denken. Verwenden Sie die Breitband-Verbindungen zwischen den lokalen Zugriffsseiten (siehe Abbildung 2 auf Seite 25).

In diesem Beispiel gibt es zwei Hauptsysteme und eine Reihe von Satellitensystemen. Die tatsächliche Konfiguration hängt von Geschäftsaspekten ab. Es gibt zwei Konzentratorwarteschlangenmanager, die sich in praktischen Centern befinden. Jeder QM-Konzentrator verfügt über Nachrichtenkanäle zu den lokalen WS-Managern:

- Der QM-Konzentrator 1 verfügt über Nachrichtenkanäle zu jedem der drei lokalen WS-Manager QM1, QM2 und QM3. Die Anwendungen, die diese WS-Manager verwenden, können über die QM-Konzentratoren miteinander kommunizieren.
- Der QM-Konzentrator 2 verfügt über Nachrichtenkanäle zu jedem der drei lokalen WS-Manager QM4, QM5 und QM6. Die Anwendungen, die diese WS-Manager verwenden, können über die QM-Konzentratoren miteinander kommunizieren.
- Die QM-Konzentratoren haben Nachrichtenkanäle untereinander, so dass jede Anwendung in einem WS-Manager Nachrichten mit jeder anderen Anwendung in einem anderen WS-Manager austauschen kann.

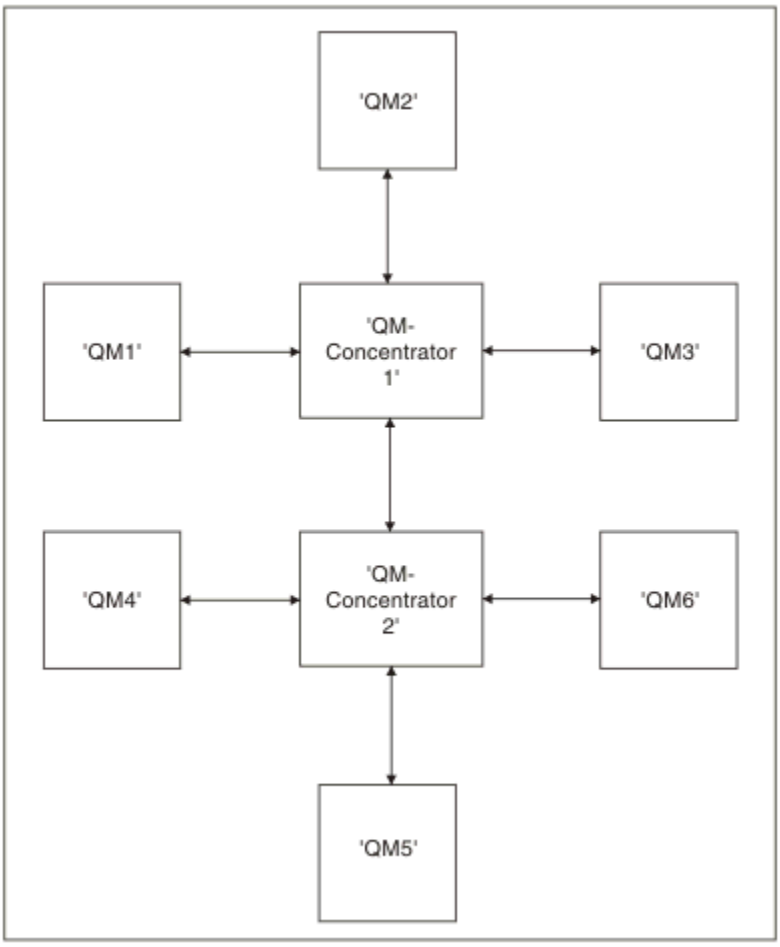


Abbildung 2. Netzdiagramm mit QM-Konzentratoren

Cluster entwerfen

Cluster bieten einen Mechanismus für die Verbindung von Warteschlangenmanagern in einer Weise, die sowohl die Erstkonfiguration als auch die laufende Verwaltung vereinfacht. Cluster müssen sorgfältig entworfen werden, um sicherzustellen, dass sie ordnungsgemäß funktionieren, und dass sie die erforderliche Verfügbarkeit und Reaktionsfähigkeit erreichen.

Vorbereitende Schritte

Eine Einführung in Clustering-Konzepte finden Sie in den folgenden Themen:

- Verteilte Steuerung von Warteschlangen und Clustern
- „Vergleich von Clustering und verteilter Steuerung von Warteschlangen” auf Seite 32
- Komponenten eines Clusters

Wenn Sie den WS-Manager-Cluster entwerfen, müssen Sie einige Entscheidungen treffen. Sie müssen zuerst entscheiden, welche WS-Manager im Cluster die vollständigen Repositorys der Clusterinformationen enthalten sollen. Jeder Warteschlangenmanager, den Sie erstellen, kann in einem Cluster arbeiten. Sie können eine beliebige Anzahl von Warteschlangenmanagern für diesen Zweck auswählen, aber die ideale Zahl ist zwei. Informationen zum Auswählen von Warteschlangenmanagern zum Speichern der vollständigen Repositorys finden Sie unter „Clusterwarteschlangenmanager für die Aufnahme von vollständigen Repositorys auswählen” auf Seite 34.

Weitere Informationen zum Entwerfen des Clusters finden Sie in den folgenden Abschnitten:

- „Beispielcluster” auf Seite 41

- „Cluster verwalten” auf Seite 36
- „Namenskonventionen für Cluster” auf Seite 36
- z/OS „Queue sharing groups and clusters” auf Seite 38
- „Überlappende Cluster” auf Seite 38

Nächste Schritte

Weitere Informationen zum Konfigurieren und Arbeiten mit Clustern finden Sie in den folgenden Abschnitten:

- Kommunikation in einem Cluster einrichten
- Warteschlangenmanager-Cluster konfigurieren
- Nachrichten an und von Clustern weiterleiten
- Cluster für das Workload-Management verwenden

Weitere Informationen zum Konfigurieren des Clusters finden Sie unter „Tipps zum Clustering” auf Seite 39.

Verwendung mehrerer Clusterübertragungswarteschlangen planen

Sie können Übertragungswarteschlangen explizit definieren oder das System die Übertragungswarteschlangen für Sie generieren lassen. Wenn Sie die Übertragungswarteschlangen selbst definieren, haben Sie mehr Kontrolle über die Warteschlangendefinitionen. z/OS Unter z/OS haben Sie auch mehr Kontrolle über die Seitengruppe, in der die Nachrichten gespeichert werden.

Übertragungswarteschlangen definieren

Es gibt zwei Methoden zum Definieren von Übertragungswarteschlangen:

- Automatisch unter Verwendung des Warteschlangenmanagerattributs DEFCLXQ wie folgt:

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

  DEFCLXQ (SCTQ) gibt an, dass die Standardübertragungswarteschlange für alle Clustersenderkanäle SYSTEM.CLUSTER.TRANSMIT.QUEUE ist. Dies ist der Standardwert.

  DEFCLXQ (CHANNEL) gibt an, dass jeder Clustersenderkanal standardmäßig eine eigene Übertragungswarteschlange mit dem Namen SYSTEM.CLUSTER.TRANSMIT. *channel name* verwendet. Jede Übertragungswarteschlange wird automatisch vom WS-Manager definiert. Weitere Informationen finden Sie unter „Automatisch definierte Clusterübertragungswarteschlangen” auf Seite 28.
- Manuell, indem eine Übertragungswarteschlange mit einem Wert definiert wird, der für das Attribut CLCHNAME angegeben wurde. Das Attribut CLCHNAME gibt an, welche Clustersenderkanäle die Übertragungswarteschlange verwenden sollen. z/OS Wenn Sie eine Übertragungswarteschlange unter z/OSmanuell definieren, finden Sie weitere Informationen in „Planung für manuell definierte Clusterübertragungswarteschlangen” auf Seite 29 .

Welche Sicherheit brauche ich?

Um einen Schalter einzuleiten, entweder automatisch oder manuell, benötigen Sie die Berechtigung zum Starten eines Kanals.

Um die Warteschlange definieren zu können, die als Übertragungswarteschlange verwendet werden soll, benötigen Sie die IBM MQ-Standardberechtigung.

Wann ist ein geeigneter Zeitpunkt für die Umsetzung der Änderung?

Wenn Sie die Übertragungswarteschlange ändern, die von Clustersenderkanälen verwendet wird, müssen Sie eine Zeit zuordnen, in der die Aktualisierung unter Berücksichtigung der folgenden Punkte gemacht werden soll:

- Die Zeit, die für einen Kanal benötigt wird, um die Übertragungswarteschlange zu wechseln, hängt von der Gesamtzahl der Nachrichten in der alten Übertragungswarteschlange, von der Anzahl der zu verschiebungsbedürftigen Nachrichten und von der Größe der Nachrichten ab.
- Anwendungen können Nachrichten weiterhin in die Übertragungswarteschlange stellen, während die Änderung stattfindet. Dies kann zu einer Erhöhung der Übergangszeit führen.
- Sie können den Parameter CLCHNAME einer beliebigen Übertragungswarteschlange oder DEFCLXQ zu einem beliebigen Zeitpunkt ändern, vorzugsweise wenn die Auslastung niedrig ist.

  Beachten Sie, dass nichts sofort passiert.
- Änderungen treten nur auf, wenn ein Kanal gestartet oder neu gestartet wird. Wenn ein Kanal gestartet wird, überprüft er die aktuelle Konfiguration und wechselt bei Bedarf in eine neue Übertragungswarteschlange.
- Es gibt mehrere Änderungen, die die Zuordnung eines Clustersenderkanals mit einer Übertragungswarteschlange ändern können:
  - Ändern Sie den Wert des CLCHNAME-Attributs einer Übertragungswarteschlange, wodurch CLCHNAME weniger spezifisch oder leer ist.
  - Ändern des Werts für das Attribut CLCHNAME einer Übertragungswarteschlange, wodurch CLCHNAME spezifischer wird.
  - Es wird eine Warteschlange mit dem angegebenen CLCHNAME gelöscht.
  - Ändern des Warteschlangenmanagerattributs DEFCLXQ.

Wie lange dauert der Wechsel?

Während des Übergangszeitraums werden alle Nachrichten für den Kanal von einer Übertragungswarteschlange in eine andere übertragen. Die Zeit, die für einen Kanal benötigt wird, um die Übertragungswarteschlange zu wechseln, hängt von der Gesamtzahl der Nachrichten in der alten Übertragungswarteschlange und von der Anzahl der zu verschiebungsbedürftigen Nachrichten ab.

Für Warteschlangen, die einige tausend Nachrichten enthalten, sollte es unter einer Sekunde dauern, bis die Nachrichten verschoben werden. Die tatsächliche Zeit hängt von der Anzahl und Größe der Nachrichten ab. Ihr Warteschlangenmanager sollte in der Lage sein, Nachrichten in vielen Megabyte pro Sekunde zu verschieben.

Anwendungen können Nachrichten weiterhin in die Übertragungswarteschlange stellen, während die Änderung stattfindet. Dies kann zu einer Erhöhung der Übergangszeit führen.

Jeder betroffene Clustersenderkanal muss erneut gestartet werden, damit die Änderung wirksam wird. Daher ist es am besten, die Konfiguration der Übertragungswarteschlange zu ändern, wenn der Warteschlangenmanager nicht ausgelastungslos ist, und es werden nur wenige Nachrichten in den Clusterübertragungswarteschlangen gespeichert.

Der **runswchl** Befehl z/OS oder der Befehl SWITCH CHANNEL (*) STATUS in CSQUTIL unter z/OS können verwendet werden, um den Status von Clustersenderkanälen und die anstehenden Änderungen abzufragen, die an ihrer Konfiguration der Übertragungswarteschlange ausstehen.

Vorgehensweise zum Implementieren der Änderung

Weitere Informationen dazu, wie Sie die Änderungen an mehreren Clusterübertragungswarteschlangen vornehmen, entweder automatisch oder manuell, finden Sie im Abschnitt System mit mehreren Clusterübertragungswarteschlangen implementieren .

Änderung rückgängig machen

z/OS

Weitere Informationen zum Zurücksetzen von Änderungen bei Problemen finden Sie im Abschnitt Änderungen an einer Übertragungswarteschlange unter z/OSrückgängig machen .

Automatisch definierte Clusterübertragungswarteschlangen
Sie können das System die Übertragungswarteschlangen für Sie generieren lassen.

Vorbereitende Schritte

z/OS So richten Sie die Cluster-Übertragungswarteschlangen manuell ein aufz/OS , sehen„Planung für manuell definierte Clusterübertragungswarteschlangen” auf Seite 29 .

Informationen zu diesem Vorgang

Wenn ein Kanal nicht über eine manuell definierte Clusterübertragungswarteschlange verfügt, die ihm zugeordnet ist, und Sie DEFCLXQ (CHANNEL) angeben, definiert der Kanal beim Starten des Kanals automatisch eine permanent-dynamische Warteschlange für den Clustersenderkanal. Die Modellwarteschlange SYSTEM.CLUSTER.TRANSMIT.MODEL.queue wird verwendet, um die permanente dynamische Clusterübertragungswarteschlange mit dem Namen SYSTEM.cluster.transmit `ChannelName` automatisch zu definieren.

Wichtig: z/OS InIBM MQ 8.0 hat der Warteschlangenmanager nicht dieSYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE . Sie können nicht direkt migrieren vonIBM MQ 8.0 zu dieser Version. Informationen zum Hinzufügen derSYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE in einen Warteschlangenmanager, der migriert wird vonIBM MQ 8.0 finden Sie in diesem Thema in der Dokumentation zur Zwischenversion, die Sie zum Migrieren des Warteschlangenmanagers verwendet haben.

Vorgehensweise

1. Verwenden Sie das WS-Manager-Attribut *DEFCLXQ* .
   Weitere Informationen zu diesem Attribut finden Sie in ALTER QMGR .

   Es gibt zwei Optionen:

   SCTQ
   Diese Option ist die Standardeinstellung und bedeutet, dass Sie die einzelne Warteschlange SYSTEM.CLUSTER.TRANSMIT.QUEUE verwenden.

   CHANNEL
   Bedeutet, dass Sie mehrere Clusterübertragungswarteschlangen verwenden.
2. Gehen Sie wie folgt vor, um zu der neuen Zuordnung
   - Stoppen Sie den Kanal und starten Sie ihn erneut.
   - Der Kanal verwendet die neue Definition der Übertragungswarteschlange.
   - Nachrichten werden von einem Übergangsschalterprozess aus der alten Warteschlange in die neue Übertragungswarteschlange übertragen.

   Beachten Sie, dass alle Anwendungsnachrichten in die alte Definition gestellt werden.

   Wenn die Anzahl der Nachrichten in der alten Warteschlange null erreicht, werden neue Nachrichten direkt in die neue Übertragungswarteschlange gestellt.
3. Gehen Sie wie folgt vor, um zu überwachen, wann der Switching
   a) Ein Switch der Übertragungswarteschlange, der von einem Kanal eingeleitet wird, wird im Hintergrund ausgeführt, und Ihr Administrator kann das Jobprotokoll des Warteschlangenmanagers überwachen, um festzustellen, wann es abgeschlossen ist.
   b) Überwachen Sie Nachrichten im Jobprotokoll, um den Fortschritt des Switch anzuzeigen.

c) Um sicherzustellen, dass nur die gewünschten Kanäle diese Übertragungswarteschlange verwenden, geben Sie den Befehl DIS CLUSQMGR(*) ein, wobei die Eigenschaft der Übertragungswarteschlange, die die Übertragungswarteschlange definiert, z. B. `APPQMGR.CLUSTER1.XMITQ`lautet.

d) z/OS

Verwenden Sie den Befehl SWITCH CHANNEL (*) STATUS unter CSQUTIL.

Diese Option gibt Auskunft darüber, welche anstehenden Änderungen ausstehen und wie viele Nachrichten zwischen den Übertragungswarteschlangen verschoben werden müssen.

Ergebnisse

Sie haben die Clusterübertragungswarteschlange oder die Warteschlangen für die Clusterübertragung konfiguriert.

Zugehörige Tasks

„Planung für manuell definierte Clusterübertragungswarteschlangen” auf Seite 29
AnIBM MQ for z/OS Wenn Sie die Übertragungswarteschlangen selbst definieren, haben Sie mehr Kontrolle über die Definitionen und den Seitensatz, auf dem die Nachrichten gespeichert werden.

Zugehörige Verweise

ALTER QMGR

DISPLAY CLUSQMGR

z/OS *Planung für manuell definierte Clusterübertragungswarteschlangen*

AnIBM MQ for z/OS Wenn Sie die Übertragungswarteschlangen selbst definieren, haben Sie mehr Kontrolle über die Definitionen und den Seitensatz, auf dem die Nachrichten gespeichert werden.

Vorbereitende Schritte

Informationen zum automatischen Einrichten von Cluster-Übertragungswarteschlangen finden Sie unter„Automatisch definierte Clusterübertragungswarteschlangen” auf Seite 28 .

Informationen zu diesem Vorgang

Ihr Administrator definiert manuell eine Übertragungswarteschlange und verwendet das Warteschlangenattribut CLCHNAME, um zu definieren, welcher Cluster-Senderkanal oder welche Cluster-Senderkanäle diese Warteschlange als Übertragungswarteschlange verwenden.

Beachten Sie, dass CLCHNAME am Anfang oder am Ende ein Platzhalterzeichen enthalten kann, damit eine einzige Warteschlange für mehrere Kanäle verwendet werden kann.

Vorgehensweise

1. Geben Sie z. B. Folgendes ein:

```
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)
```

Tipp: Sie müssen planen, welche Seitengruppe (und Pufferpool) Sie für Ihre Übertragungswarteschlangen verwenden. Sie können für unterschiedliche Warteschlangen unterschiedliche Seitensätze haben und diese voneinander isolieren, so dass das Auffüllen eines Seitensatzes keine Auswirkungen auf die Übertragungswarteschlangen in anderen Seitensätzen hat.

Informationen dazu, wie jeder Kanal die entsprechende Warteschlange auswählt, finden Sie im Abschnitt Mit Clusterübertragungswarteschlangen und Clustersenderkanälen arbeiten .

Wenn der Kanal startet, wechselt er seine Zuordnung zur neuen Übertragungswarteschlange. Um sicherzustellen, dass keine Nachricht verloren geht, überträgt der Warteschlangenmanager Nachrichten automatisch und der Reihe nach aus der alten Cluster-Übertragungswarteschlange in die neue Übertragungswarteschlange.

2. Verwenden Sie die Funktion CSQUTIL SWITCH, um in die neue Zuordnung zu wechseln.

   Weitere Informationen finden Sie im Abschnitt Die Übertragungswarteschlange, die Clustersenderkanälen (SWITCH) zugeordnet ist) umschalten.

   a) STOP den Kanal oder die Kanäle, deren Übertragungswarteschlange geändert werden soll, so dass sie sich im Status STOPPED befinden.

      For example:

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

   b) Ändern Sie das Attribut `CLCHNAME(XXXX)` in der Übertragungswarteschlange.

   c) Verwenden Sie die Funktion SWITCH, um die Nachrichten zu wechseln oder die Vorgänge zu überwachen.

      Verwenden Sie den Befehl:

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

      um die Nachrichten zu verschieben, ohne den Kanal zu starten.

   d) Starten Sie den Kanal oder die Kanäle, und überprüfen Sie, ob der Kanal die richtigen Warteschlangen verwendet.

      For example:

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

Tipp: Der folgende Prozess verwendet die CSQUTIL SWITCH-Funktion. Weitere Informationen finden Sie unter Wechseln der den Cluster-Sender-Kanälen zugeordneten Übertragungswarteschlange (SWITCH) .

Sie müssen diese Funktion nicht verwenden, aber mit dieser Funktion stehen weitere Optionen zur Auswahl:

- Mit SWITCH CHANNEL (*) STATUS können Sie den Schaltstatus von Clustersenderkanälen auf einfache Weise ermitteln. Es ermöglicht Ihrem Administrator, zu sehen, welche Kanäle derzeit geschaltet werden, und die Kanäle, die einen Switch anstehen, die wirksam werden, wenn diese Kanäle nächsten Start sind.

  Ohne diese Funktion muss der Administrator mehrere DISPLAY-Befehle verwenden und anschließend die resultierende Ausgabe verarbeiten, um diese Informationen zu ermitteln. Ihr Administrator kann auch bestätigen, dass eine Konfigurationsänderung das erforderliche Ergebnis hat.
- Wenn CSQUTIL zum Starten des Switch verwendet wird, überwacht CSQUTIL den Fortschritt dieser Operation weiter und wird nur beendet, wenn der Switch abgeschlossen ist.

  Dies kann die Ausführung dieser Operationen im Stapelbetrieb erheblich erleichtern. Wenn CSQUTIL zum Umschalten mehrerer Kanäle ausgeführt wird, führt CSQUTIL diese Aktionen nacheinander aus. Dies kann weniger Auswirkungen auf Ihr Unternehmen haben als mehrere Switches, die parallel ausgeführt werden.

Ergebnisse

Sie haben Ihre Cluster-Übertragungswarteschlange(n) eingerichtet aufz/OS .

Zugriffssteuerung und mehrere Clusterübertragungswarteschlangen

Wählen Sie zwischen drei Prüfmodi aus, wenn eine Anwendung Nachrichten in ferne Clusterwarteschlangen einreiht. Die Modi sind: Prüfung über Fernzugriff gegen die Clusterwarteschlange, lokale Prüfung gegen SYSTEM.CLUSTER.TRANSMIT.QUEUE oder Prüfung gegen lokale Profile für die Clusterwarteschlange oder den Clusterwarteschlangenmanager.

IBM MQ gibt Ihnen die Möglichkeit, entweder lokal oder lokal und remote zu überprüfen, ob ein Benutzer berechtigt ist, eine Nachricht in eine ferne Warteschlange einzureihen. Eine typische IBM MQ-Anwendung verwendet nur die lokale Überprüfung und ist darauf angewiesen, dass der ferne Warteschlangenmanager der Zugriffsprüfung vertraut, die auf dem lokalen Warteschlangenmanager durchgeführt wurde. Wenn die ferne Prüfung nicht verwendet wird, wird die Nachricht mit der Berechtigung des fernen Nachrichtenkanalprozesses in die Zielwarteschlange gestellt. Um die Fernprüfung verwenden zu können, müssen Sie die Berechtigung 'put' für den empfangenden Kanal auf die Kontextsicherheit setzen.

Die lokalen Prüfungen werden für die Warteschlange, die die Anwendung öffnet, durchgeführt. Bei der verteilten Steuerung von Warteschlangen öffnet die Anwendung in der Regel eine Definition einer fernen Warteschlange und die Zugriffsprüfungen werden auf die Definition der fernen Warteschlange gestellt. Wenn die Nachricht mit einem vollständigen Routing-Header verbunden wird, werden die Prüfungen für die Übertragungswarteschlange durchgeführt. Wenn eine Anwendung eine Clusterwarteschlange öffnet, die sich nicht im lokalen WS-Manager befindet, gibt es kein lokales Objekt, das überprüft werden kann. Die Zugriffssteuerungsprüfungen werden anhand der Clusterübertragungswarteschlange SYSTEM.CLUSTER.TRANSMIT.QUEUE durchgeführt. Selbst bei mehreren Clusterübertragungswarteschlangen werden lokale Zugriffssteuerungsprüfungen für ferne Clusterwarteschlangen anhand von SYSTEM.CLUSTER.TRANSMIT.QUEUEdurchgeführt.

Die Auswahl der lokalen oder fernen Prüfung ist eine Auswahl zwischen zwei Extremwerte. Die ferne Überprüfung ist in differenzierter Ausführung. Jeder Benutzer muss über ein Zugriffssteuerungsprofil auf jedem WS-Manager im Cluster verfügen, um in eine beliebige Clusterwarteschlange zu stellen. Die lokale Überprüfung ist grob-grainiert. Jeder Benutzer benötigt nur ein Zugriffssteuerungsprofil für die Clusterübertragungswarteschlange auf dem Warteschlangenmanager, mit dem sie verbunden sind. Mit diesem Profil können sie eine Nachricht in jede Clusterwarteschlange auf einem beliebigen WS-Manager in einem beliebigen Cluster einlegen.

Administratoren haben eine andere Möglichkeit, die Zugriffssteuerung für Clusterwarteschlangen einzurichten. Mit dem Befehl **setmqaut** können Sie ein Sicherheitsprofil für eine Clusterwarteschlange auf einem beliebigen Warteschlangenmanager im Cluster erstellen. Das Profil wirkt sich darauf aus, wenn Sie eine ferne Clusterwarteschlange lokal öffnen und dabei nur den Namen der Warteschlange angeben. Sie können auch ein Profil für einen fernen WS-Manager einrichten. Wenn Sie dies tun, kann der Warteschlangenmanager das Profil eines Benutzers überprüfen, der eine Clusterwarteschlange öffnet, indem er einen vollständig qualifizierten Namen bereitstellt.

Die neuen Profile funktionieren nur, wenn Sie die Zeilengruppe des Warteschlangenmanagers **ClusterQueueAccessControl** in RQMNameändern. Der Standardwert ist Xmitq . Sie müssen Profile für alle vorhandenen Clusterwarteschlangen erstellen, die Clusterwarteschlangen verwenden. Wenn Sie die Zeilengruppe in RQMName ändern, ohne Profile zu erstellen, werden die Anwendungen wahrscheinlich fehlschlagen.

Tipp: Die Zugriffsprüfung für Clusterwarteschlangen gilt nicht für die ferne Warteschlangensteuerung. Es werden weiterhin Zugriffsprüfungen für lokale Definitionen durchgeführt. Die Änderungen bedeuten, dass Sie denselben Ansatz verfolgen können, um die Zugriffsprüfung für Clusterwarteschlangen und Clusterthemen zu konfigurieren. z/OS Außerdem führen die Änderungen zu einer Annäherung zwischen der Zugriffsprüfungsmethode für Clusterwarteschlangen und z/OS. Unter z/OS werden zwar andere Befehle zum Konfigurieren der Zugriffsprüfung verwendet, doch in beiden Fällen wird die Zugriffsberechtigung anhand eines Profils und nicht anhand des Objekts selbst geprüft.

Zugehörige Konzepte

„Clustering: Anwendungsisolation mit mehreren Clusterübertragungswarteschlangen” auf Seite 51
Sie können die Nachrichtenflüsse zwischen Warteschlangenmanagern in einem Cluster isolieren. Sie können Nachrichten, die von verschiedenen Clustersenderkanälen transportiert werden, in verschiedene Clusterübertragungswarteschlangen stellen. Sie können den Ansatz in einem einzelnen Cluster oder mit

überlappenden Clustern verwenden. Das Thema enthält Beispiele und einige bewährte Verfahren, die Sie bei der Auswahl eines zu verwendenden Ansatzes führen.

Zugehörige Tasks

Einstellung `ClusterQueueAccessControl`

Vergleich von Clustering und verteilter Steuerung von Warteschlangen

Vergleichen Sie die Komponenten, die für die Verbindung von WS-Managern mit verteilter Steuerung von Warteschlangen und Clustering definiert werden müssen.

Wenn Sie keine Cluster verwenden, sind Ihre Warteschlangenmanager unabhängig und kommunizieren mit der verteilten Steuerung von Warteschlangen. Wenn ein Warteschlangenmanager Nachrichten an einen anderen senden muss, müssen Sie Folgendes definieren:

- eine Übertragungswarteschlange
- Ein Kanal zum fernen Warteschlangenmanager

Abbildung 3 auf Seite 32 zeigt die Komponenten, die für die verteilte Steuerung von Warteschlangen erforderlich sind.

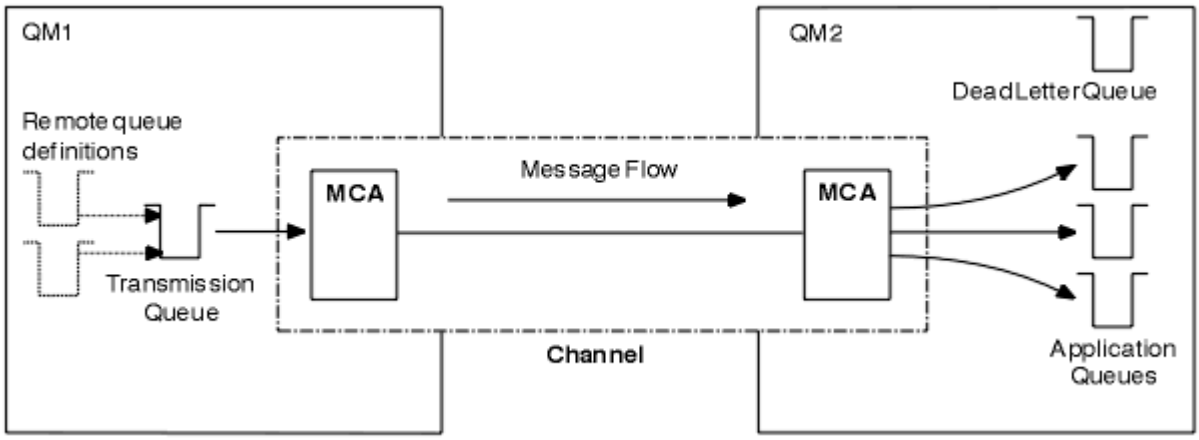


Abbildung 3. Verteilte Steuerung von Warteschlangen

Wenn Sie Warteschlangenmanager in einem Cluster gruppieren, stehen Warteschlangen in einem beliebigen WS-Manager allen anderen Warteschlangenmanagern im Cluster zur Verfügung. Jeder WS-Manager kann ohne explizite Definitionen eine Nachricht an jeden anderen Warteschlangenmanager in demselben Cluster senden. Sie stellen keine Kanaldefinitionen, Definitionen für ferne Warteschlangen oder Übertragungswarteschlangen für die einzelnen Ziele zur Verfügung. Jeder WS-Manager in einem Cluster verfügt über eine einzige Übertragungswarteschlange, von der er Nachrichten an jeden anderen WS-Manager im Cluster übertragen kann. Jeder WS-Manager in einem Cluster muss nur Folgendes definieren:

- Ein Clusterempfängerkanal, auf dem Nachrichten empfangen werden sollen.
- Ein Clustersenderkanal, mit dem er sich selbst einführt und die Informationen zum Cluster

Definitionen zum Festlegen eines Clusters im Vergleich zu verteilten Warteschlangensteuerung

Sehen Sie sich Abbildung 4 auf Seite 33 an, in dem jeweils vier Warteschlangenmanager mit jeweils zwei Warteschlangen angezeigt werden. Überlegen Sie, wie viele Definitionen benötigt werden, um diese WS-Manager mit Hilfe der verteilten Steuerung von Warteschlangen zu verbinden. Vergleichen Sie die Anzahl der Definitionen, die zum Festlegen des gleichen Netzes wie ein Cluster benötigt werden.

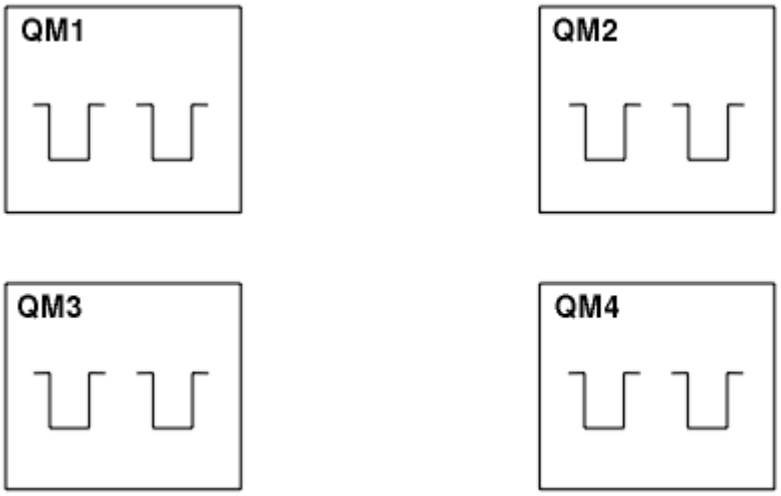


Abbildung 4. Ein Netz mit vier Warteschlangenmanagern

Definitionen zum Festlegen eines Netzes unter Verwendung der verteilten Steuerung von Warteschlangen

Um das in Abbildung 3 auf Seite 32 unter Verwendung der verteilten Warteschlangensteuerung gezeigte Netz einzurichten, können Sie die folgenden Definitionen verwenden:

Tabelle 2. Definitionen für verteilte Steuerung von Warteschlangen

Beschreibung	Anzahl pro WS-Manager	Gesamtanzahl
Eine Senderkanaldefinition für einen Kanal, auf dem Nachrichten an alle anderen Warteschlangenmanager gesendet werden sollen.	3	12
Eine Empfängerkanaldefinition für einen Kanal, auf dem Nachrichten von jedem anderen WS-Manager empfangen werden sollen.	3	12
Eine Übertragungswarteschlangendefinition für eine Übertragungswarteschlange zu jedem anderen Warteschlangenmanager	3	12
Eine lokale Warteschlangendefinition für jede lokale Warteschlange.	2	8
Eine Definition einer fernen Warteschlange für jede ferne Warteschlange, in die dieser WS-Manager Nachrichten einlegen möchte.	6	24

Sie können diese Anzahl von Definitionen unter Verwendung generischer Empfängerkanaldefinitionen reduzieren. Die maximale Anzahl von Definitionen kann bis zu 17 auf jedem Warteschlangenmanager betragen, was insgesamt 68 für dieses Netz ist.

Definitionen zum Festlegen eines Netzes mit Clustern

Um das in Abbildung 3 auf Seite 32 gezeigte Netz unter Verwendung von Clustern einzurichten, benötigen Sie die folgenden Definitionen:

Tabelle 3. Definitionen für Clustering

Beschreibung	Anzahl pro WS-Manager	Gesamtanzahl
Clustersenderkanaldefinition für einen Kanal, auf dem Nachrichten an einen Repository-WS-Manager gesendet werden sollen	1	4

Tabelle 3. Definitionen für Clustering (Forts.)		
Beschreibung	**Anzahl pro WS-Manager**	**Gesamtanzahl**
Eine Clusterempfängerkanaldefinition für einen Kanal, auf dem Nachrichten von anderen Warteschlangenmanagern im Cluster empfangen werden sollen.	1	4
Eine lokale Warteschlangendefinition für jede lokale Warteschlange.	2	8

Um diesen Cluster von Warteschlangenmanagern (mit zwei vollständigen Repositorys) zu konfigurieren, benötigen Sie auf jedem Warteschlangenmanager vier Definitionen, insgesamt sind insgesamt 16 Definitionen vorhanden. Außerdem müssen Sie die WS-Manager-Definitionen für zwei der Warteschlangenmanager ändern, damit sie vollständige WS-Manager-Repository-WS-Manager für den Cluster bilden.

Es ist nur eine CLUSSDR -und eine CLUSRCVR -Kanaldefinition erforderlich. Wenn der Cluster definiert ist, können Sie Warteschlangenmanager (außer den Repository-WS-Managern) hinzufügen oder entfernen, ohne dass die anderen WS-Manager gestört werden.

Wenn Sie einen Cluster verwenden, wird die Anzahl der Definitionen reduziert, die zum Festlegen eines Netzes mit vielen Warteschlangenmanagern erforderlich sind.

Wenn weniger Definitionen vorhanden sind, besteht die Gefahr eines Fehlers:

- Objektnamen stimmen immer überein, z. B. der Kanalname in einem Sender-Empfänger-Paar.
- Der in einer Kanaldefinition angegebene Übertragungswarteschlangenname stimmt immer mit der korrekten Übertragungswarteschlangendefinition oder dem Namen der Übertragungswarteschlange überein, die in einer Definition einer fernen Warteschlange angegeben ist.
- Eine QREMOTE -Definition verweist immer auf die richtige Warteschlange auf dem fernen Warteschlangenmanager.

Sobald ein Cluster konfiguriert ist, können Sie Clusterwarteschlangen von einem Warteschlangenmanager in einen anderen im Cluster verschieben, ohne dass Systemverwaltungsaufgaben für einen anderen Warteschlangenmanager ausgeführt werden müssen. Es besteht keine Möglichkeit, die Definitionen von Kanal-, Fernwarteschlangen-oder Übertragungswarteschlangen zu löschen oder zu ändern. Sie können neue Warteschlangenmanager zu einem Cluster hinzufügen, ohne dass das vorhandene Netz unterbrochen wird.

Clusterwarteschlangenmanager für die Aufnahme von vollständigen Repositorys auswählen

In jedem Cluster müssen Sie mindestens eine, vorzugsweise zwei WS-Manager auswählen, um vollständige Repositorys zu speichern. Zwei vollständige Repositorys sind für alle, aber die außergewöhnlichsten Umstände ausreichend. Wählen Sie, wenn möglich, Warteschlangenmanager aus, die auf stabilen und permanent verbundenen Plattformen gehostet sind, die keine übereinstimmenden Ausfälle haben und die sich geographisch in einer zentralen Position befinden. Beachten Sie außerdem, dass Systeme als vollständige Repository-Hosts dedizieren sind und diese Systeme nicht für andere Tasks verwenden.

Vollständige Repositorys sind WS-Manager, die ein vollständiges Bild des Status des Clusters erhalten. Um diese Informationen gemeinsam nutzen zu können, ist jedes vollständige Repository über CLUSSDR -Kanäle (und die zugehörigen CLUSRCVR -Definitionen) mit jedem anderen vollständigen Repository im Cluster verbunden. Sie müssen diese Kanäle manuell definieren.

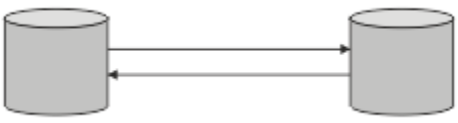


Abbildung 5. Zwei verbundene vollständige Repositorys.

Jeder andere WS-Manager im Cluster verwaltet ein Bild dessen, was er derzeit über den Status des Clusters in einem *Teilrepository* weiß. Diese WS-Manager veröffentlichen Informationen über sich selbst und fordern Informationen zu anderen Warteschlangenmanagern unter Verwendung von zwei verfügbaren vollständigen Repositorys an. Wenn ein ausgewähles vollständiges Repository nicht verfügbar ist, wird ein anderes verwendet. Wenn das ausgewählte vollständige Repository wieder verfügbar wird, erfasst es die neuesten neuen und geänderten Informationen von den anderen, so dass sie in Schritt halten. Wenn alle vollständigen Repositorys außer Betrieb sind, verwenden die anderen WS-Manager die Informationen, die sie in ihren Teilrepositorys haben. Sie beschränken sich jedoch auf die Verwendung der Informationen, die sie haben; neue Informationen und Anforderungen für Aktualisierungen können nicht verarbeitet werden. Wenn die vollständigen Repositorys wieder eine Verbindung zum Netz herstellen, werden Nachrichten ausgetauscht, um alle Repositorys (sowohl vollständige als auch partielle) auf dem neuesten Stand zu bringen.

Wenn Sie die Zuordnung von vollständigen Repositorys planen, müssen Sie die folgenden Aspekte berücksichtigen:

- Die Warteschlangenmanager, die zum Speichern der vollständigen Repositorys ausgewählt wurden, müssen zuverlässig und verwaltet sein. Wählen Sie die Warteschlangenmanager aus, die auf einer stabilen und permanent verbundenen Plattform gehostet werden.
- Berücksichtigen Sie die geplanten Ausfälle für die Systeme, die als Host für Ihre vollständigen Repositorys verwendet werden, und stellen Sie sicher, dass sie nicht übereinstimmende Ausfälle aufweisen.
- Berücksichtigen Sie die Netzleistung: Wählen Sie die Warteschlangenmanager aus, die sich geographisch in einer zentralen Position befinden oder die dasselbe System wie andere Warteschlangenmanager im Cluster gemeinsam nutzen.
- Überlegen Sie, ob ein Warteschlangenmanager Mitglied von mehreren Clustern ist. Es kann administrativ praktisch sein, denselben WS-Manager für die Verwendung der vollständigen Repositorys für mehrere Cluster zu verwenden, vorausgesetzt, dieser Vorteil ist ausgeglichen, wie ausgelastert die Auslastung des Warteschlangenmanagers ist.
- Sie sollten einige Systeme dedizieren, um nur vollständige Repositorys zu enthalten, und diese Systeme nicht für andere Tasks zu verwenden. Auf diese Weise wird sichergestellt, dass diese Systeme nur für die Konfiguration des Warteschlangenmanagers gewartet werden müssen und nicht aus dem Service für die Wartung anderer Geschäftsanwendungen entfernt werden. Außerdem stellt sie sicher, dass die Task zum Verwalten des Repositorys nicht mit Anwendungen für Systemressourcen konkursfähig ist. Dies kann besonders in großen Clustern (beispielsweise Cluster mit mehr als tausend Warteschlangenmanagern) von Vorteil sein, wenn die Auslastung des Clusterstatus durch die vollständigen Repositorys erheblich höher ist.

Es ist möglich, mehr als zwei vollständige Repositorys zu verwenden, wird aber selten empfohlen. Obwohl Objektdefinitionen (d. a. Warteschlangen, Themen und Kanäle) in alle verfügbaren vollständigen Repositorys fließen, werden nur Anforderungen von einem Teilrepository an maximal zwei vollständige Repositorys gestellt. Dies bedeutet, dass, wenn mehr als zwei vollständige Repositorys definiert sind und alle zwei vollständigen Repositorys nicht mehr verfügbar sind, einige Teilrepositorys möglicherweise keine Aktualisierungen empfangen, die sie erwarten würden. Weitere Informationen finden Sie unter MQ-Cluster: Warum nur zwei vollständige Repositorys?

Eine Situation, in der Sie möglicherweise mehr als zwei vollständige Repositorys definieren können, ist die Migration vorhandener vollständiger Repositorys auf neue Hardware oder neue Warteschlangenmanager. In diesem Fall sollten Sie die Ersatz-Vollrepositorys einführen und bestätigen, dass sie vollständig gefüllt wurden, bevor Sie die vorherigen vollständigen Repositorys entfernen. Wenn Sie ein vollständiges Repository hinzufügen, müssen Sie sich daran erinnern, dass Sie es mit CLUSSDR -Kanälen direkt mit jedem anderen vollständigen Repository verbinden müssen.

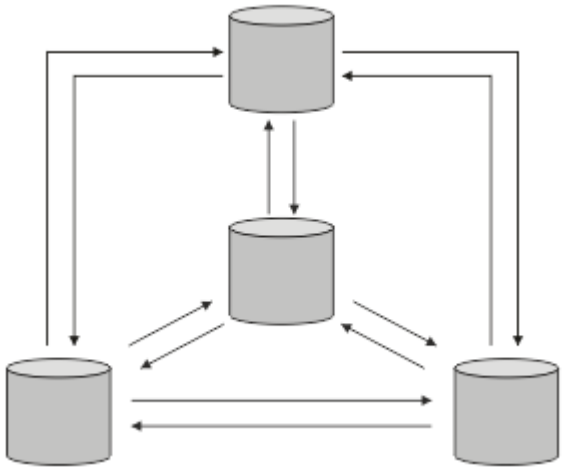


Abbildung 6. Mehr als zwei verbundene vollständige Repositorys

Zugehörige Informationen
MQ-Cluster: Warum nur zwei vollständige Repositorys?
Wie groß kann ein MQ-Cluster sein?

Cluster verwalten
Wählen Sie aus, welche WS-Manager mit welchem vollständigen Repository verknüpft werden sollen. Berücksichtigen Sie den Leistungseinwirkungseffekt, die Version des Warteschlangenmanagers und die Angabe, ob mehrere CLUSSDR -Kanäle wünschenswert sind.

Nachdem Sie die Warteschlangenmanager ausgewählt haben, um vollständige Repositorys zu speichern, müssen Sie festlegen, welche WS-Manager mit welchem vollständigen Repository verknüpft werden sollen. Die Kanaldefinition CLUSSDR verknüpft einen Warteschlangenmanager mit einem vollständigen Repository, aus dem es Informationen zu den anderen vollständigen Repositorys im Cluster herausfindet. Von da an sendet der Warteschlangenmanager Nachrichten an alle zwei vollständigen Repositorys. Es wird immer versucht, zuerst die Kanaldefinition in CLUSSDR zu verwenden. Sie können auswählen, ob ein Warteschlangenmanager mit einem vollständigen Repository verknüpft werden soll. Wählen Sie bei der Auswahl die Topologie Ihrer Konfiguration und die physische oder geografische Position der Warteschlangenmanager aus.

Da alle Clusterinformationen an zwei vollständige Repositorys gesendet werden, kann es zu Situationen kommen, in denen Sie eine zweite CLUSSDR -Kanaldefinition erstellen möchten. Sie können einen zweiten CLUSSDR -Kanal in einem Cluster definieren, der über viele vollständige Repositorys verfügt, die sich über einen weiten Bereich erstrecken. Sie können dann steuern, an welche zwei vollständigen Repositorys Ihre Daten gesendet werden.

Namenskonventionen für Cluster
Sie sollten die Benennung von Warteschlangenmanagern in demselben Cluster unter Verwendung einer Namenskonvention berücksichtigen, die den Cluster angibt, zu dem der Warteschlangenmanager gehört. Verwenden Sie eine ähnliche Namenskonvention für Kanalnamen und erweitern Sie diese, um die Kanalmerkmale zu beschreiben.

Bewährte Verfahren bei der Benennung von MQ -Clustern

Obwohl Clusternamen bis zu 48 Zeichen lang sein können, sind relativ kurze Clusternamen hilfreich, wenn Namenskonventionen auf andere Objekte angewendet werden. Weitere Informationen finden Sie unter „Bewährte Verfahren bei der Auswahl von Clusterkanalnamen” auf Seite 37.

Bei der Auswahl eines Clusternamens ist es normalerweise hilfreich, den 'Zweck 'des Clusters (der wahrscheinlich langlebig ist) und nicht den 'Inhalt' darzustellen. Beispiel: 'B2BPROD' oder 'ACTTEST' statt 'QM1_QM2_QM3_CLUS'.

Bewährte Verfahren bei der Auswahl von Clusterwarteschlangenmanagernamen

Wenn Sie einen neuen Cluster und seine Member völlig neu erstellen, ziehen Sie eine Namenskonvention für die Warteschlangenmanager in Betracht, die ihre Clusternutzung widerspiegelt. Jeder WS-Manager muss einen anderen Namen haben. Sie können jedoch Warteschlangenmanagern in einem Cluster eine Gruppe ähnlicher Namen geben, um logische Gruppierungen zu identifizieren und sich daran zu erinnern (z. B. 'ACTTQM1, ACTTQM2).

Relativ kurze Warteschlangenmanagernamen (z. B. weniger als 8 Zeichen) helfen Ihnen, wenn Sie die im nächsten Abschnitt beschriebene Konvention oder eine ähnliche Konvention für Kanalnamen verwenden.

Bewährte Verfahren bei der Auswahl von Clusterkanalnamen

Da Warteschlangenmanager und Cluster Namen mit bis zu 48 Zeichen haben können und ein Kanalname auf 20 Zeichen begrenzt ist, müssen Sie beim ersten Benennen von Objekten darauf achten, dass die Namenskonvention nicht in der Mitte eines Projekts geändert werden muss (siehe vorherigen Abschnitt).

Denken Sie beim Definieren von Kanälen daran, dass automatisch erstellte Clustersenderkanäle auf jedem Warteschlangenmanager im Cluster ihren Namen von dem entsprechenden Clusterempfängerkanal übernehmen, der auf dem empfangenden Warteschlangenmanager im Cluster konfiguriert ist. Diese müssen daher eindeutig sein und *auf fernen Warteschlangenmanagern im Cluster sinnvoll sein.*

Eine allgemeine Methode ist die Verwendung des Namens des Warteschlangenmanagers, dem der Clustername vorangestellt ist. Wenn der Clustername beispielsweise CLUSTER1 lautet und die Warteschlangenmanager QM1, QM2 sind, lauten die Clusterempfängerkanäle CLUSTER1.QM1, CLUSTER1.QM2.

Sie können diese Konvention erweitern, wenn Kanäle unterschiedliche Prioritäten haben oder unterschiedliche Protokolle verwenden. For example:

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

In diesem Beispiel könnte S1 der erste SNA-Kanal, N3 der NetBIOS -Kanal mit der Netzpriorität 3 und T4 die TCP/IP-Adresse unter Verwendung eines IPV4 -Netzes sein.

Gemeinsame Kanaldefinitionen benennen

Eine einzelne Kanaldefinition kann von mehreren Clustern gemeinsam genutzt werden. In diesem Fall müssen die hier vorgeschlagenen Namenskonventionen geändert werden. Wie in Kanaldefinitionen verwalten beschrieben ist es jedoch in der Regel vorzuziehen, für jeden Cluster diskrete Kanäle zu definieren.

Ältere Namenskonventionen für Kanäle

Außerhalb von Clusterumgebungen war es in der Vergangenheit üblich, eine 'FROMQM.TO.TARGETQM' -Namenskonvention zu verwenden, sodass Sie möglicherweise feststellen, dass vorhandene Cluster etwas Ähnliches verwendet haben (z. B. CLUSTER.TO.TARGET). Dies wird nicht als Teil eines neuen Clusterbenennungsschemas empfohlen, weil es die verfügbaren Zeichen weiter reduziert, um 'nützliche' Informationen in Ihrem Kanalnamen zu vermitteln.

z/OS Kanalnamen unter IBM MQ for z/OS

Sie können generische VTAM-Ressourcen oder generische DDNS-Namen (*Dynamic Domain Name Server*) definieren. Sie können Verbindungsnamen mit generischen Namen definieren. Wenn Sie jedoch eine Clusterempfängerdefinition erstellen, verwenden Sie keinen generischen Verbindungsnamen.

Das Problem bei der Verwendung generischer Verbindungsnamen für Clusterempfängerdefinitionen lautet wie folgt: Wenn Sie einen CLUSRCVR mit einem generischen CONNAME definieren, gibt es keine Garantie, dass Ihre CLUSSDR -Kanäle auf die Warteschlangenmanager verweisen, die Sie beabsichtigen. Der ursprüngliche CLUSSDR-Kanal kann auf jeden Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange verweisen, nicht notwendigerweise auf einen Warteschlangenmanager,

der ein vollständiges Repository enthält. Wenn ein Kanal erneut versucht, eine Verbindung herzustellen, stellt er möglicherweise eine Verbindung zu einem anderen Warteschlangenmanager mit demselben generischen Namen her, was den Nachrichtenfluss unterbricht.

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.

Sie können generische VTAM-Ressourcen oder generische DDNS-Namen (*Dynamic Domain Name Server*) definieren. Sie können Verbindungsnamen mit generischen Namen definieren. Wenn Sie jedoch eine Clusterempfängerdefinition erstellen, verwenden Sie keinen generischen Verbindungsnamen.

Das Problem bei der Verwendung generischer Verbindungsnamen für Clusterempfängerdefinitionen lautet wie folgt: Wenn Sie einen CLUSRCVR mit einem generischen CONNAME definieren, gibt es keine Garantie, dass Ihre CLUSSDR -Kanäle auf die Warteschlangenmanager verweisen, die Sie beabsichtigen. Der ursprüngliche CLUSSDR-Kanal kann auf jeden Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange verweisen, nicht notwendigerweise auf einen Warteschlangenmanager, der ein vollständiges Repository enthält. Wenn ein Kanal erneut versucht, eine Verbindung herzustellen, stellt er möglicherweise eine Verbindung zu einem anderen Warteschlangenmanager mit demselben generischen Namen her, was den Nachrichtenfluss unterbricht.

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.

Überlappende Cluster

Überlappende Cluster bieten zusätzliche Verwaltungsfunktionen. Verwenden Sie Namenslisten, um die Anzahl der Befehle zu reduzieren, die für die Verwaltung von überlappenden Clustern erforderlich sind.

Sie können Cluster erstellen, die sich überschneiden. Es gibt eine Reihe von Gründen für die Definition von überlappenden Clustern, z. B.:

- Damit andere Organisationen ihre eigene Verwaltung haben können.
- Damit unabhängige Anwendungen separat verwaltet werden können.
- Zum Erstellen von Serviceklassen.

In Abbildung 7 auf Seite 39 ist der Warteschlangenmanager STF2 ein Mitglied beider Cluster. Wenn ein Warteschlangenmanager Mitglied mehrerer Cluster ist, können Sie die Namenslisten nutzen, um die Anzahl der benötigten Definitionen zu reduzieren. Namelists enthalten eine Liste mit Namen, z. B. Clusternamen. Sie können eine Namensliste erstellen, die die Cluster benennt. Geben Sie die Namensliste im Befehl ALTER QMGR für STF2 an, um sie zu einem vollständigen Warteschlangenmanager-Repository für beide Cluster zu machen.

Wenn Sie mehr als einen Cluster in Ihrem Netzwerk haben, müssen Sie ihnen unterschiedliche Namen geben. Wenn zwei Cluster mit demselben Namen jemals zusammengeführt werden, ist es nicht möglich,

sie wieder zu trennen. Es ist auch eine gute Idee, den Clustern und Kanälen unterschiedliche Namen zu geben. Sie lassen sich einfacher unterscheiden, wenn Sie die Ausgabe der DISPLAY-Befehle anzeigen. Die Namen von Warteschlangenmanagern müssen innerhalb eines Clusters eindeutig sein, damit sie ordnungsgemäß funktionieren.

Serviceklassen definieren

Stellen Sie sich eine Universität vor, die über einen Warteschlangenmanager für jedes Mitglied des Personals und jeden Schüler verfügt. Nachrichten zwischen Mitarbeitern sind auf Kanälen mit hoher Priorität und hoher Bandbreite zu bereisen. Nachrichten zwischen Studenten werden auf billigeren, langsameren Kanälen zu reisen. Sie können dieses Netz mit Hilfe der traditionellen Methoden zur verteilten Steuerung von Warteschlangen konfigurieren. IBM MQ wählt die Kanäle, die verwendet werden sollen, anhand der Namen von Zielwarteschlange und Warteschlangenmanager aus.

Um die Mitarbeiter und Studenten eindeutig zu unterscheiden, können Sie ihre Warteschlangenmanager in zwei Clustern gruppieren (siehe Abbildung 7 auf Seite 39). IBM MQ verschiebt Nachrichten in die Sitzungswarteschlange im Mitarbeitercluster nur über Kanäle, die in diesem Cluster definiert sind. Die Nachrichten für die Warteschlange 'gossip' im Cluster 'students' gehen über Kanäle, die in diesem Cluster definiert sind, und empfängt die entsprechende Serviceklasse.

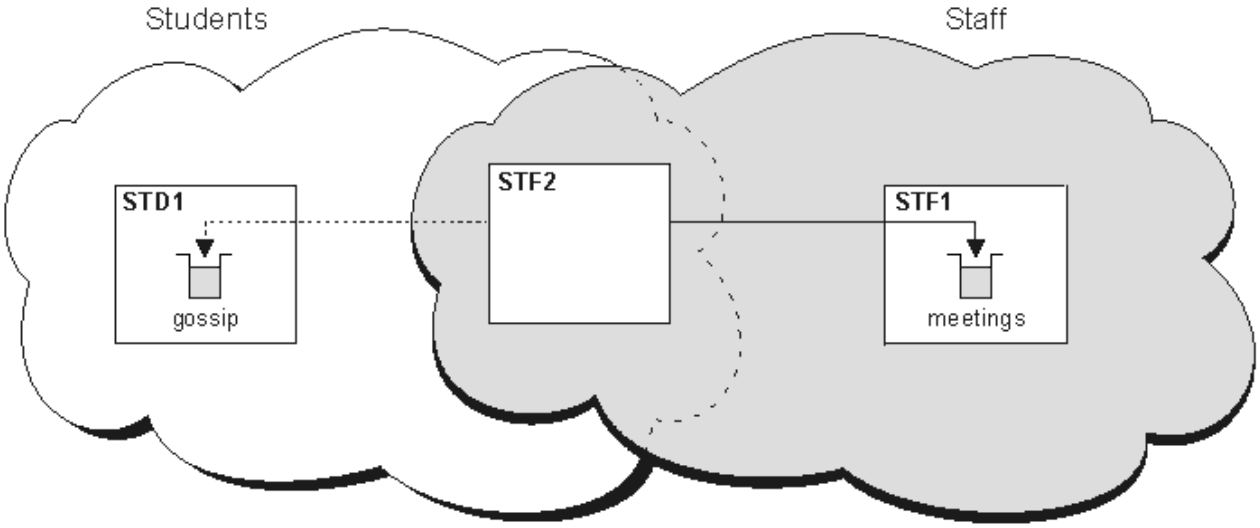


Abbildung 7. Serviceklassen

Tipps zum Clustering

Möglicherweise müssen Sie einige Änderungen an Ihren Systemen oder Anwendungen vornehmen, bevor Sie Clustering verwenden. Es gibt sowohl Ähnlichkeiten als auch Unterschiede zwischen dem Verhalten der verteilten Steuerung von Warteschlangen.

- Sie müssen manuelle Konfigurationsdefinitionen zu Warteschlangenmanagern außerhalb eines Clusters hinzufügen, damit sie auf Clusterwarteschlangen zugreifen können.
- Wenn Sie zwei Cluster mit demselben Namen zusammenführen, können Sie sie nicht erneut trennen. Daher ist es ratsam, allen Clustern einen eindeutigen Namen zu geben.
- Wenn eine Nachricht in einem Warteschlangenmanager ankommt, aber keine Warteschlange vorhanden ist, wird die Nachricht in die Warteschlange für dead-Mail gestellt. Wenn es keine Warteschlange für dead-letter gibt, schlägt der Kanal fehl und versucht erneut, die Warteschlange zu wiederholen. Die Verwendung der Warteschlange "dead-letter" ist mit der Verwendung der verteilten Steuerung von Warteschlangen identisch.
- Die Integrität persistenter Nachrichten wird beibehalten. Nachrichten werden aufgrund der Verwendung von Clustern nicht dupliziert oder gehen verloren.
- Die Verwendung von Clustern reduziert die Systemverwaltung. Cluster machen es einfach, größere Netzwerke mit vielen mehr WS-Managern zu verbinden, als Sie in der Lage wären, die verteilte Steue-

rung von Warteschlangen zu verwenden. Es besteht die Gefahr, dass Sie übermäßige Netzressourcen in Anspruch nehmen, wenn Sie versuchen, die Kommunikation zwischen jedem WS-Manager in einem Cluster zu aktivieren.

- Da IBM MQ Explorer die Warteschlangenmanager in einer Baumstruktur darstellt, kann die Ansicht sehr großer Cluster eventuell umständlich sein.
- Multi Der Zweck von Verteilerlisten besteht darin, einen einzelnen MQPUT -Befehl zu verwenden, um dieselbe Nachricht an mehrere Ziele zu senden. Verteilerliste werden von IBM MQ for Multiplatforms unterstützt. Sie können Verteilerlisten mit WS-Manager-Clustern verwenden. In einem Cluster werden alle Nachrichten zum Zeitpunkt MQPUT erweitert. Der Vorteil in Bezug auf den Datenaustausch im Netz ist nicht so hoch wie in einer Nicht-Clustering-Umgebung. Der Vorteil von Verteilerlisten besteht darin, dass die zahlreichen Kanäle und Übertragungswarteschlangen nicht manuell definiert werden müssen.
- Wenn Sie Cluster verwenden möchten, um die Auslastung Ihrer Workload zu überprüfen, untersuchen Sie Ihre Anwendungen. Sie können feststellen, ob Nachrichten von einem bestimmten Warteschlangenmanager oder in einer bestimmten Reihenfolge verarbeitet werden müssen. Solche Anwendungen sollen Nachrichtenaffinitäten haben. Möglicherweise müssen Sie Ihre Anwendungen ändern, bevor Sie sie in komplexen Clustern verwenden können.
- Sie können die Option MQOO_BIND_ON_OPEN in einem MQOPEN verwenden, um das Senden von Nachrichten an ein bestimmtes Ziel zu erzwingen. Wenn der Zielwarteschlangenmanager nicht verfügbar ist, werden die Nachrichten erst zugestellt, wenn der WS-Manager wieder verfügbar ist. Nachrichten werden aufgrund des Risikos der Duplizierung nicht an einen anderen WS-Manager weitergeleitet.
- Wenn ein WS-Manager ein Cluster-Repository hosten soll, müssen Sie seinen Hostnamen oder seine IP-Adresse kennen. Sie müssen diese Informationen im Parameter CONNAME angeben, wenn Sie die Definition CLUSSDR auf anderen Warteschlangenmanagern, die den Cluster verbinden, definieren. Wenn Sie DHCP verwenden, kann sich die IP-Adresse ändern, da DHCP bei jedem Neustart eines Systems eine neue IP-Adresse zuordnen kann. Daher dürfen Sie die IP-Adresse in den CLUSSDR -Definitionen nicht angeben. Selbst wenn alle CLUSSDR -Definitionen den Hostnamen und nicht die IP-Adresse angeben, sind die Definitionen immer noch nicht zuverlässig. DHCP aktualisiert nicht notwendigerweise den DNS-Verzeichniseintrag für den Host mit der neuen Adresse. Wenn Sie WS-Manager als vollständige Repositorys auf Systemen benennen müssen, die DHCP verwenden, installieren Sie die Software, die garantiert, dass Ihr DNS-Verzeichnis auf dem neuesten Stand ist.
- Verwenden Sie keine generischen Namen, z. B. generische VTAM-Ressourcen oder generische DDNS-Namen (Dynamic Domain Name Server) als Verbindungsnamen für Ihre Kanäle. Wenn dies der Fall ist, stellen die Kanäle möglicherweise eine Verbindung zu einem anderen Warteschlangenmanager her als erwartet.
- Sie können eine Nachricht nur aus einer lokalen Clusterwarteschlange abrufen, aber Sie können eine Nachricht in eine beliebige Warteschlange in einem Cluster stellen. Wenn Sie eine Warteschlange zur Verwendung des Befehls MQGET öffnen, öffnet der Warteschlangenmanager die lokale Warteschlange.
- Sie müssen keine Ihrer Anwendungen ändern, wenn Sie einen einfachen IBM MQ-Cluster einrichten. Die Anwendung kann die Zielwarteschlange im MQOPEN -Aufruf benennen und muss die Position des Warteschlangenmanagers nicht kennen. Wenn Sie einen Cluster für das Workload-Management einrichten, müssen Sie Ihre Anwendungen überprüfen und sie bei Bedarf ändern.
- Sie können die aktuellen Überwachungs-und Statusdaten für einen Kanal oder eine Warteschlange mit den Befehlen DISPLAY CHSTATUS und DISPLAY QSTATUS **runmqsc** anzeigen. Die Überwachungsdaten können verwendet werden, um die Leistung und den Status des Systems zu messen. Die Überwachung wird über WS-Manager-, Warteschlangen-und Kanalattribute gesteuert. Die Überwachung automatisch definierter Clustersenderkanäle ist mit dem WS-Manager-Attribut MONACLS möglich.

Zugehörige Konzepte

Cluster

„Vergleich von Clustering und verteilter Steuerung von Warteschlangen” auf Seite 32
Vergleichen Sie die Komponenten, die für die Verbindung von WS-Managern mit verteilter Steuerung von Warteschlangen und Clustering definiert werden müssen.

Komponenten eines Clusters

Zugehörige Tasks

WS-Manager-Cluster konfigurieren

Neuen Cluster einrichten

Wie lange werden die Informationen in den Warteschlangenmanager-Repositorys aufbewahrt?

WS-Manager-Repositorys behalten Informationen für 30 Tage bei. Ein automatischer Prozess aktualisiert die Informationen, die gerade verwendet werden, effizient.

Wenn ein Warteschlangenmanager einige Informationen über sich selbst sendet, speichern die vollständigen und partiellen Repository-WS-Manager die Informationen für 30 Tage. Informationen werden beispielsweise gesendet, wenn ein WS-Manager die Erstellung einer neuen Warteschlange bewirbt. Damit diese Informationen nicht auslaufen, werden die Warteschlangenmanager nach 27 Tagen automatisch alle Informationen über sich selbst erneut senden. Wenn ein Teilrepository eine neue Anforderung zum Teil über die 30-Tage-Lebensdauer sendet, bleibt die Ablaufzeit die ursprünglichen 30 Tage.

Wenn Informationen verfallen, wird sie nicht sofort aus dem Repository entfernt. Stattdessen wird sie für eine Karenzzeit von 60 Tagen gehalten. Wenn innerhalb der Karenzzeit keine Aktualisierung empfangen wird, werden die Informationen entfernt. Die Karenzzeit ermöglicht es, dass ein WS-Manager zum Ablaufdatum vorübergehend außer Betrieb war. Wenn ein WS-Manager länger als 90 Tage von einem Cluster getrennt wird, wird er nicht mehr Teil des Clusters. Wenn er jedoch wieder eine Verbindung zum Netz herstellt, wird er wieder Teil des Clusters. Vollständige Repositorys verwenden keine Informationen, die abgelaufen sind, um neue Anforderungen von anderen Warteschlangenmanagern zu erfüllen.

Wenn ein Warteschlangenmanager eine Anforderung zum Sichern/Enddatum aus einem vollständigen Repository sendet, dauert die Anforderung ebenfalls 30 Tage. Nach 27 Tagen IBM MQ prüft die Anforderung. Wenn sie in den 27 Tagen referenziert wurde, wird sie automatisch aktualisiert. Ist dies nicht der Fall, bleibt sie verfallen und wird vom WS-Manager aktualisiert, wenn sie erneut benötigt wird. Das Ablaufen von Anforderungen verhindert eine Ansammlung von Anforderungen für Informationen von ruhenden WS-Managern.

Anmerkung: Sie sollten das PTF für APAR PH43191herunterladen und installieren, das Systemfehler bei der Berechnung der Ablaufzeit eines Abonnements behebt. Diese Fehler können dazu führen, dass die Subskription frühzeitig abläuft (was dazu führt, dass die Nachricht CSQX456I ausgegeben wird) oder nach Ablauf des Objekts abläuft (was zu Fehlern des Typs MQRC 2085 (MQRC_UNKNOWN_OBJECT) führt).

Bei großen Clustern kann es unterbrechend sein, wenn viele Warteschlangenmanager automatisch alle Informationen zu sich selbst erneut senden. Nähere Informationen hierzu erhalten Sie im Abschnitt Die Aktualisierung in einem großen Cluster kann sich auf die Leistung und Verfügbarkeit auswirken.

Zugehörige Konzepte

„Clustering: Best Practices für REFRESH CLUSTER verwenden” auf Seite 74

Sie verwenden den Befehl **REFRESH CLUSTER** , um alle lokal gespeicherten Informationen zu einem Cluster zu löschen und diese Informationen aus den vollständigen Repositorys im Cluster erneut zu erstellen. Sie sollten diesen Befehl nicht verwenden, außer in außergewöhnlichen Umständen. Wenn Sie es verwenden müssen, gibt es besondere Hinweise darauf, wie Sie es verwenden. Diese Informationen sind ein Leitfaden, der auf Tests und Feedback von Kunden basiert.

Beispielcluster

Das erste Beispiel zeigt den kleinsten möglichen Cluster von zwei Warteschlangenmanagern. Im zweiten und dritten Beispiel werden zwei Versionen eines drei WS-Manager-Clusters angezeigt.

Der kleinste mögliche Cluster enthält nur zwei WS-Manager. In diesem Fall enthalten beide WS-Manager vollständige Repositorys. Sie benötigen nur wenige Definitionen, um den Cluster zu konfigurieren, und dennoch gibt es bei jedem WS-Manager einen hohen Grad an Autonomie.

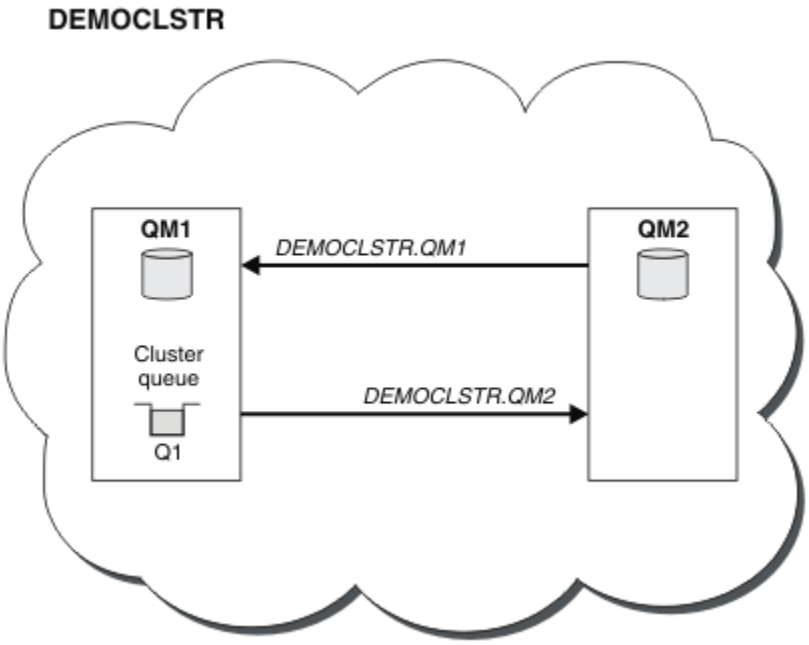


Abbildung 8. Kleiner Cluster mit zwei WS-Managern

- Die Warteschlangenmanager können lange Namen wie LONDON und NEWYORK aufweisen. z/OS In IBM MQ for z/OS dürfen Warteschlangenmanagernamen nur vier Zeichen haben.
- Jeder WS-Manager ist in der Regel auf einer separaten Maschine konfiguriert. Sie können jedoch mehrere WS-Manager auf derselben Maschine haben.

Anweisungen zum Konfigurieren eines ähnlichen Beispielclusters finden Sie im Abschnitt Neuen Cluster einrichten .

Abbildung 9 auf Seite 43 zeigt die Komponenten eines Clusters mit dem Namen CLSTR1.

- In diesem Cluster gibt es drei Warteschlangenmanager: QM1, QM2 und QM3.
- QM1 und QM2 Host-Repositorys mit Informationen zu allen Warteschlangenmanagern und clusterbezogenen Objekten im Cluster. Sie werden als *vollständige WS-Manager-Repository-Warteschlangenmanager* bezeichnet. Die Repositorys werden in dem Diagramm durch die schattierten Zylinder dargestellt.
- QM2 und QM3 enthalten einige Warteschlangen, auf die alle anderen Warteschlangenmanager im Cluster zugreifen können. Warteschlangen, die für alle anderen WS-Manager im Cluster zugänglich sind, werden als *Clusterwarteschlangen* bezeichnet. Die Clusterwarteschlangen werden in dem Diagramm durch die schraffierten Warteschlangen dargestellt. Auf Clusterwarteschlangen kann von einer beliebigen Position im Cluster aus zugegriffen werden. Mit dem IBM MQ-Clustering-Code wird sichergestellt, dass für Clusterwarteschlangen auf jedem Warteschlangenmanager, der auf sie verweist, entsprechende Definitionen für ferne Warteschlangen erstellt werden.

  Wie bei der verteilten Steuerung von Warteschlangen verwendet eine Anwendung den Aufruf MQPUT , um eine Nachricht in eine Clusterwarteschlange auf einem beliebigen Warteschlangenmanager im Cluster einzureihen. Eine Anwendung verwendet den Aufruf MQGET , um Nachrichten aus einer Clusterwarteschlange nur auf dem Warteschlangenmanager abzurufen, auf dem sich die Warteschlange befindet.
- Jeder Warteschlangenmanager verfügt über eine manuell erstellte Definition für das Empfangsende eines Kanals mit dem Namen *cluster_name. queue_manager_name*, auf dem Nachrichten empfangen werden können. Auf dem Empfangswarteschlangenmanager ist *cluster_name. queue_manager_name* ein Cluster-Empfängerkanal. Ein Clusterempfängerkanal ist wie ein Empfängerkanal, der in der verteilten Warteschlangensteuerung verwendet wird; er empfängt Nachrichten für den Warteschlangenmanager. Darüber hinaus erhält er auch Informationen über den Cluster.

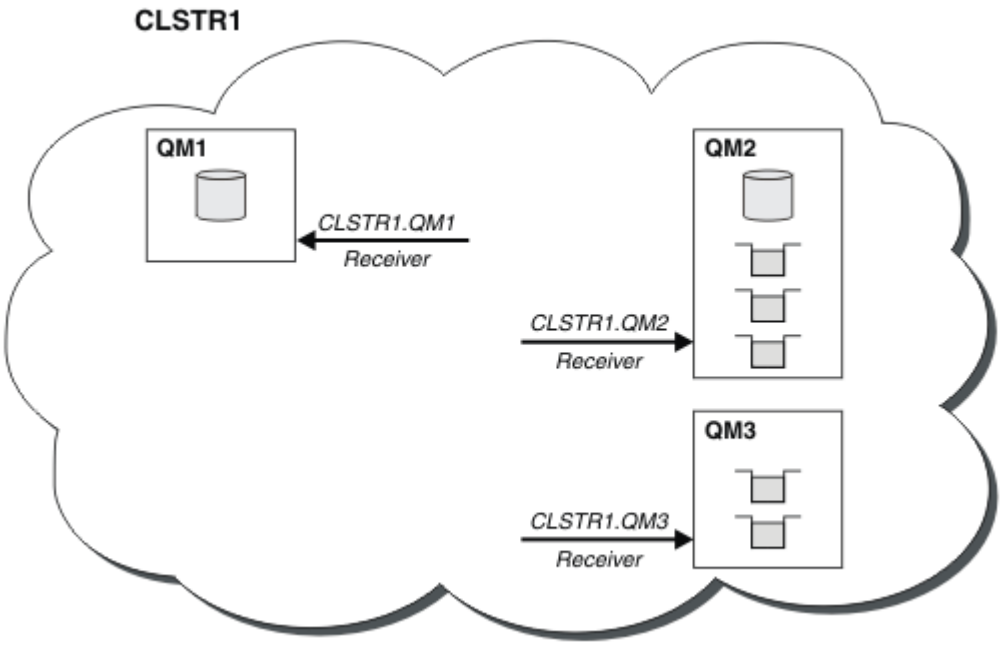
- 

Abbildung 9. Warteschlangenmanagercluster

- In Abbildung 10 auf Seite 44 verfügt jeder Warteschlangenmanager auch über eine Definition für die Sendeseite des Kanals. Es stellt eine Verbindung zum Clusterempfängerkanal eines der vollständigen WS-Manager-Repositorys her. Auf dem sendenden Warteschlangenmanager ist *cluster_name*.*queue_manager_name* ein Clustersenderkanal. QM1 und QM3 verfügen über Clustersenderkanäle, die eine Verbindung zu CLSTR1.QM2 herstellen, siehe gepunktete Linie "2".

  QM2 verfügt über einen Clustersenderkanal, der mit CLSTR1.QM1 verbunden ist (siehe gepunktete Linie "3"). Ein Clustersenderkanal ist vergleichbar mit einem Senderkanal, wie er bei der verteilten Steuerung von Warteschlangen verwendet wird. Über ihn werden Nachrichten an den empfangenden Warteschlangenmanager gesendet. Darüber hinaus sendet er auch Informationen zum Cluster.

  Wenn sowohl das Clusterempfängerende als auch das Clustersenderende eines Kanals definiert sind, wird der Kanal automatisch gestartet.

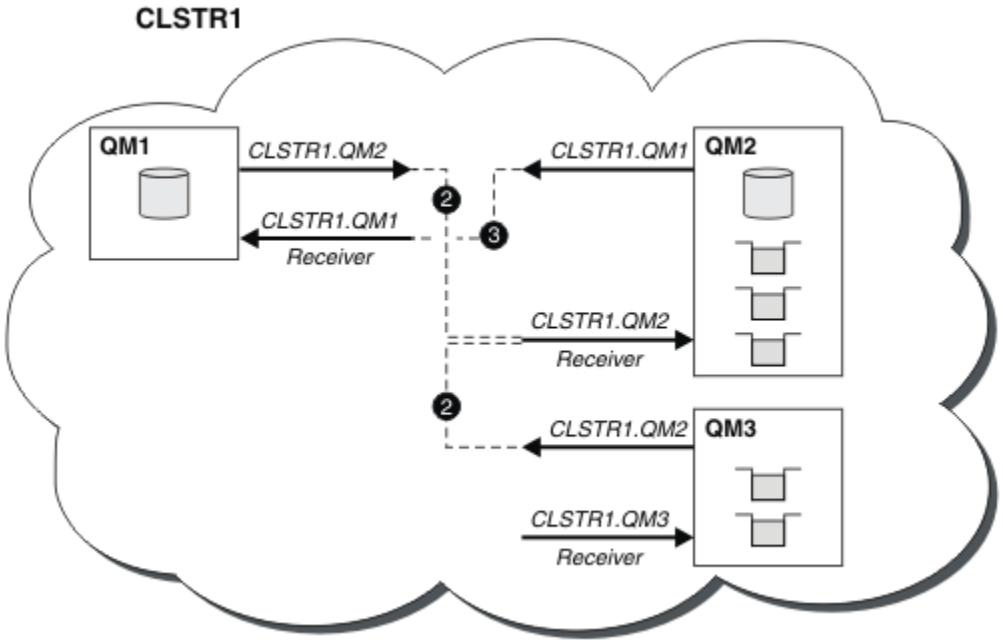


Abbildung 10. Ein Cluster von WS-Managern mit Senderkanälen

Wenn Sie einen Clustersenderkanal auf dem lokalen Warteschlangenmanager definieren, wird dieser Warteschlangenmanager zu einem der vollständigen WS-Manager-Repositorys eingeführt. Der vollständige Repository-WS-Manager aktualisiert die Informationen in seinem vollständigen Repository entsprechend. Anschließend erstellt es automatisch einen Clustersenderkanal zurück zum ursprünglichen Warteschlangenmanager und sendet diese WS-Manager-Informationen über den Cluster. Daher lernt ein WS-Manager über einen Cluster und ein Cluster lernt einen Warteschlangenmanager.

Sehen Sie sich Abbildung 9 auf Seite 43 noch einmal an. Angenommen, eine Anwendung, die mit dem Warteschlangenmanager QM3 verbunden ist, möchte einige Nachrichten an die Warteschlangen von QM2 senden. Wenn QM3 zum ersten Mal auf diese Warteschlangen zugreifen muss, erkennt es sie anhand eines vollständigen Repositorys. Das vollständige Repository ist in diesem Fall QM2, auf das über den Senderkanal CLSTR1.QM2 zugegriffen wird. Mit den Informationen aus dem Repository kann es automatisch ferne Definitionen für diese Warteschlangen erstellen. Wenn sich die Warteschlangen unter QM1 befinden, funktioniert dieser Mechanismus weiterhin, da QM2 ein vollständiges Repository ist. Ein vollständiges Repository verfügt über einen vollständigen Datensatz aller Objekte im Cluster. In diesem Fall würde QM3 auch automatisch einen Clustersenderkanal erstellen, der dem Clusterempfängerkanal unter QM1 entspricht, wodurch eine direkte Kommunikation zwischen den beiden Kanälen ermöglicht wird.

Abbildung 11 auf Seite 45 zeigt den gleichen Cluster mit den beiden automatisch erstellten Clustersenderkanälen. Die Clustersenderkanäle werden durch die beiden gestrichelten Linien dargestellt, die mit dem Clusterempfängerkanal CLSTR1.QM3verbunden sind. Außerdem wird die Clusterübertragungswarteschlange, SYSTEM.CLUSTER.TRANSMIT.QUEUE, angezeigt, die von QM1 zum Senden der Nachrichten verwendet wird. Alle WS-Manager im Cluster verfügen über eine Clusterübertragungswarteschlange, von der aus sie Nachrichten an einen beliebigen anderen Warteschlangenmanager in demselben Cluster senden können.

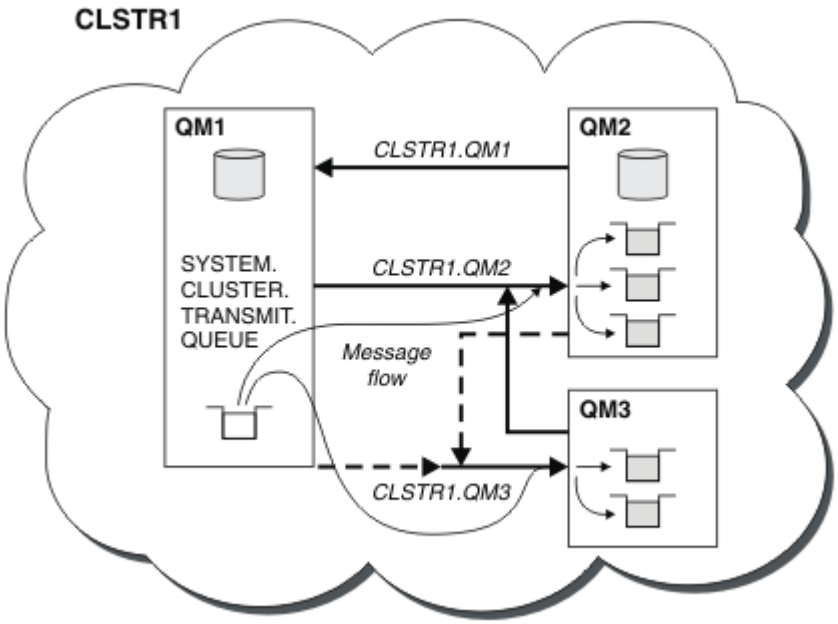


Abbildung 11. Ein Cluster von WS-Managern mit automatisch definierten Kanälen

Anmerkung: Andere Diagramme zeigen nur die Empfangsenden von Kanälen an, für die Sie manuelle Definitionen vornehmen. Die sendenden Enden werden weggelassen, da sie meist automatisch bei Bedarf definiert werden. Die automatische Definition der meisten Clustersenderkanäle ist entscheidend für die Funktion und die Effizienz von Clustern.

Zugehörige Konzepte

„Vergleich von Clustering und verteilter Steuerung von Warteschlangen” auf Seite 32
Vergleichen Sie die Komponenten, die für die Verbindung von WS-Managern mit verteilter Steuerung von Warteschlangen und Clustering definiert werden müssen.

Komponenten eines Clusters

Zugehörige Tasks

WS-Manager-Cluster konfigurieren

Neuen Cluster einrichten

Clustering: Bewährte Verfahren

Cluster stellen einen Mechanismus für die Verbindung von Warteschlangenmanagern bereit. Die in diesem Abschnitt beschriebenen bewährten Verfahren basieren auf Tests und Feedback von Kunden.

Für eine erfolgreiche Clusterkonfiguration sind eine gute Planung und umfassende Kenntnisse der Grundlagen von IBM MQ (z. B. ein gutes Anwendungsmanagement und ein durchdachter Netzentwurf) erforderlich. Stellen Sie sicher, dass Sie mit den Informationen in den zugehörigen Themen vertraut sind, bevor Sie fortfahren.

Zugehörige Konzepte

Verteilte Warteschlangen und Cluster

Cluster

Zugehörige Tasks

„Cluster entwerfen” auf Seite 25
Cluster bieten einen Mechanismus für die Verbindung von Warteschlangenmanagern in einer Weise, die sowohl die Erstkonfiguration als auch die laufende Verwaltung vereinfacht. Cluster müssen sorgfältig entworfen werden, um sicherzustellen, dass sie ordnungsgemäß funktionieren, und dass sie die erforderliche Verfügbarkeit und Reaktionsfähigkeit erreichen.

Cluster überwachen

Clustering: Besondere Hinweise zu überlappenden Clustern
Dieser Abschnitt enthält eine Anleitung zur Planung und Verwaltung von IBM MQ-Clustern. Diese Informationen sind ein Leitfaden, der auf Tests und Feedback von Kunden basiert.

Clustereigentumsrecht

Machen Sie sich mit überlappenden Clustern vertraut, bevor Sie die folgenden Informationen lesen. Informationen zu den erforderlichen Informationen finden Sie in „Überlappende Cluster” auf Seite 38 und Nachrichtenpfade zwischen Clustern konfigurieren .

Wenn Sie ein System konfigurieren und verwalten, das sich aus überlappenden Clustern zusammensetzt, ist es am besten, die folgenden Schritte zu befolgen:

- Obwohl IBM MQ-Cluster wie oben beschrieben nur 'lose verbunden' sind, ist es sinnvoll, einen Cluster als eine einzige Verwaltungseinheit zu betrachten. Dieses Konzept wird verwendet, da die Interaktion zwischen Definitionen auf einzelnen Warteschlangenmanagern für das reibungslose Funktionieren des Clusters von entscheidender Bedeutung ist. Beispiel: Bei der Verwendung von auslastungsabhängigen Clusterwarteschlangen ist es wichtig, dass ein einzelner Administrator oder ein Team die vollständige Gruppe möglicher Destinations für Nachrichten versteht, die von den im Cluster verteilten Definitionen abhängig sind. Weitere triviale, Cluster-Sender-/Empfänger-Kanal-Paare müssen in der gesamten Konfiguration kompatibel sein.
- In Anbetracht dieses früheren Konzeptes, bei dem mehrere Cluster (die von separaten Teams/Einzelpersonen verwaltet werden sollen), ist es wichtig, klare Richtlinien für die Verwaltung der Gateway-Warteschlangenmanager zu haben.
- Es ist sinnvoll, überlappende Cluster als einen einzigen Namensbereich zu behandeln: Kanalnamen und WS-Manager-Namen müssen in einem einzigen Cluster eindeutig sein. Die Verwaltung ist viel einfacher, wenn sie in der gesamten Topologie eindeutig ist. Es empfiehlt sich, eine geeignete Namenskonvention zu verwenden. Mögliche Konventionen werden in „Namenskonventionen für Cluster” auf Seite 36beschrieben.
- Manchmal ist die Zusammenarbeit zwischen Verwaltung und Systemmanagement von wesentlicher Bedeutung. Beispiel: Zusammenarbeit zwischen Organisationen, die unterschiedliche Cluster besitzen, die sich überschneiden müssen. Ein klares Verständnis dessen, wem welche und welche durchsetzbaren Regeln und Konventionen gehören, hilft beim reibungslosen Clustering, wenn sich Cluster überschneiden.

Überlappende Cluster: Gateways

Im Allgemeinen ist ein einzelner Cluster einfacher zu verwalten als mehrere Cluster. Daher ist die Erstellung einer großen Anzahl von kleinen Clustern (eine für jede Anwendung zum Beispiel) etwas zu vermeiden, die im Allgemeinen.

Sie können jedoch überlappende Cluster implementieren, um Serviceklassen zur Verfügung zu stellen. For example:

- Wenn Sie konzentrische Cluster haben, bei denen der kleinere für Publish/Subscribe vorgesehen ist. Weitere Informationen hierzu finden Sie im Abschnitt Vorgehensweise bei der Größe von Systemen .
- Wenn einige Warteschlangenmanager von verschiedenen Teams verwaltet werden sollen. Weitere Informationen finden Sie im vorherigen Abschnitt „Clustereigentumsrecht” auf Seite 46.
- Wenn es aus organisatorischer oder geographischer Sicht sinnvoll ist.
- If equivalent clusters work with name resolution, for example when implementing TLS in an existing cluster.

Es gibt keinen Sicherheitsvorteil durch überlappende Cluster. Cluster, die von zwei verschiedenen Teams verwaltet werden, können sich überschneiden, sodass sie effektiv den Teams und der Topologie beitreten:

- Alle Namen, die in einem solchen Cluster zugänglich gemacht werden, sind für den anderen Cluster zugänglich.

- Jeder Name, der in einem Cluster bekannt gemacht wird, kann im anderen Cluster bekannt gemacht werden, um infrage kommende Nachrichten abzuziehen.
- Jedes nicht zugänglich gemacht Objekt auf einem Warteschlangenmanager neben dem Gateway kann von allen Clustern aufgelöst werden, deren Mitglied das Gateway ist.

Der Namespace ist die Union der beiden Cluster und muss als einzelner Namensbereich behandelt werden. Daher wird das Eigentumsrecht an einem überlappenden Cluster von allen Administratoren beider Cluster gemeinsam genutzt.

Wenn ein System mehrere Cluster enthält, kann es erforderlich sein, Nachrichten von Warteschlangenmanagern in einem Cluster an Warteschlangen von Warteschlangenmanagern in einem anderen Cluster weiterzuleiten. In dieser Situation müssen die mehreren Cluster in irgendeiner Weise miteinander verbunden sein: Ein gutes Muster, das zu folgen ist, ist die Verwendung von Gateway-WS-Managern zwischen Clustern. Diese Anordnung verhindert die Erstellung eines schwer zu verwaltenden Netzes aus Punkt-zu-Punkt-Kanälen und stellt einen guten Platz bereit, um solche Probleme als Sicherheitsrichtlinien zu verwalten. Es gibt zwei verschiedene Möglichkeiten, diese Anordnung zu erreichen:

1. Platzieren Sie einen (oder mehrere) Warteschlangenmanager in beiden Clustern unter Verwendung einer zweiten Clusterempfängerdefinition. Diese Anordnung umfasst weniger Verwaltungsdefinitionen, bedeutet aber, wie bereits erwähnt, bedeutet, dass das Eigentumsrecht an einem überlappenden Cluster von allen Administratoren beider Cluster gemeinsam genutzt wird.
2. Verbinden Sie einen Warteschlangenmanager in Cluster eins mit einem Warteschlangenmanager in Cluster zwei mithilfe von konventionellen Punkt-zu-Punkt-Kanälen.

In einem dieser Fälle können verschiedene Tools verwendet werden, um den Datenverkehr entsprechend weiterzuleiten. Insbesondere können Aliasnamen von Warteschlangen oder Warteschlangenmanagern verwendet werden, um in den anderen Cluster zu steuern. Ein Alias eines Warteschlangenmanagers mit leerer **RQMNAME**-Eigenschaft sorgt für einen erneuten Lastausgleich, wo er gewünscht ist.

Zugehörige Konzepte

„Namenskonventionen für Cluster” auf Seite 36
Sie sollten die Benennung von Warteschlangenmanagern in demselben Cluster unter Verwendung einer Namenskonvention berücksichtigen, die den Cluster angibt, zu dem der Warteschlangenmanager gehört. Verwenden Sie eine ähnliche Namenskonvention für Kanalnamen und erweitern Sie diese, um die Kanalmerkmale zu beschreiben.

Clustering: Aspekte des Topologiedesigns

Dieser Abschnitt enthält eine Anleitung zur Planung und Verwaltung von IBM MQ-Clustern. Diese Informationen sind ein Leitfaden, der auf Tests und Feedback von Kunden basiert.

Wenn Sie darüber nachdenken, wo Benutzeranwendungen und interne Verwaltungsprozesse im Voraus lokalisiert werden sollen, können viele Probleme entweder vermieden oder zu einem späteren Zeitpunkt minimiert werden. Dieses Thema enthält Informationen zu Designentscheidungen, die die Leistung verbessern können, und vereinfachen die Verwaltungstasks als Clusterskalierungen.

- „Leistung der Clustering-Infrastruktur” auf Seite 47
- „Vollständige Repositorys” auf Seite 48
- „Sollen Anwendungen Warteschlangen in vollständigen Repositorys verwenden?” auf Seite 49
- „Kanaldefinitionen verwalten” auf Seite 50
- „Lastausgleich über mehrere Kanäle” auf Seite 50

Leistung der Clustering-Infrastruktur

Wenn eine Anwendung versucht, eine Warteschlange auf einem Warteschlangenmanager in einem Cluster zu öffnen, registriert der Warteschlangenmanager sein Interesse an den vollständigen Repositorys für diese Warteschlange, so dass er lernen kann, wo sich die Warteschlange im Cluster befindet. Alle Aktualisierungen an der Warteschlangenposition oder -konfiguration werden automatisch von den vollständigen Repositorys an den interessierten WS-Manager gesendet. Diese Registrierung von Interesse wird intern

als Subskription bezeichnet (diese Subskriptionen sind nicht mit den IBM MQ-Subskriptionen identisch, die für Publish/Subscribe-Messaging in IBM MQ verwendet werden).

Alle Informationen zu einem Cluster durchlaufen jedes vollständige Repository. Vollständige Repositorys werden daher immer in einem Cluster für den Verwaltungsnachrichtenverkehr verwendet. Die hohe Auslastung der Systemressourcen bei der Verwaltung dieser Subskriptionen und die Übertragung dieser Nachrichten und die daraus resultierenden Konfigurationsnachrichten können zu einer erheblichen Auslastung der Clustering-Infrastruktur führen. Es gibt eine Reihe von Faktoren, die zu berücksichtigen sind, wenn sichergestellt wird, dass diese Last wo immer möglich verstanden und minimiert wird:

- Je mehr einzelne WS-Manager eine Clusterwarteschlange verwenden, umso mehr Subskriptionen sind im System vorhanden, wodurch der Verwaltungsaufwand bei Änderungen größer ist und interessierte Subskribenten benachrichtigt werden müssen, insbesondere auf den vollständigen WS-Managern des Repositorys. Eine Möglichkeit, den unnötigen Datenverkehr und die Auslastung des gesamten Repositorys zu minimieren, besteht darin, ähnliche Anwendungen (d. a. die Anwendungen, die mit denselben Warteschlangen arbeiten) mit einer kleineren Anzahl von Warteschlangenmanagern zu verbinden.
- Neben der Anzahl der Subskriptionen im System, die sich auf die Leistung auswirken, kann sich die Änderungsrate in der Konfiguration von Clusterobjekten auf die Leistung auswirken, z. B. die häufige Änderung einer Clusterwarteschlangenkonfiguration.
- Wenn ein Warteschlangenmanager Mitglied mehrerer Cluster ist (d. B. er Teil eines überlappenden Clustersystems ist), wird jedes Interesse, das in einer Warteschlange erstellt wird, zu einer Subskription für jeden Cluster, zu dem er gehört, auch dann, wenn dieselben Warteschlangenmanager die vollständigen Repositorys für mehr als einen der Cluster sind. Diese Anordnung erhöht die Belastung des Systems und ist ein Grund, zu überlegen, ob mehrere überlappende Cluster erforderlich sind, und nicht nur ein einzelner Cluster.
- Anwendungsnachrichtenverkehr (d. h. Nachrichten, die von IBM MQ-Anwendungen an die Clusterwarteschlangen gesendet werden), gehen nicht über die vollständigen Repositorys, um die Zielwarteschlangenmanager zu erreichen. Dieser Nachrichtenverkehr wird direkt zwischen dem Warteschlangenmanager, in dem die Nachricht in den Cluster eintritt, und dem Warteschlangenmanager, in dem die Clusterwarteschlange vorhanden ist, gesendet. Daher ist es nicht erforderlich, hohe Aufwandmengen an Anwendungsnachrichtenverkehr in Bezug auf die vollständigen WS-Manager-Repositorys zu berücksichtigen, es sei denn, die vollständigen WS-Manager-WS-Manager sind einer der beiden genannten WS-Manager. Aus diesem Grund wird empfohlen, vollständige WS-Manager-Repositorys nicht für Anwendungsnachrichtenverkehr in Clustern zu verwenden, in denen die Clustering-Infrastrukturbelastung von Bedeutung ist.

Vollständige Repositorys

Ein Repository ist eine Zusammenstellung von Informationen über die Warteschlangenmanager, die zu einem Cluster gehören. Ein Warteschlangenmanager, der über einen vollständigen Satz von Informationen über jeden Warteschlangenmanager im Cluster verfügt, besitzt ein vollständiges Repository. Weitere Informationen zu vollständigen Repositorys und Teilrepositorys finden Sie unter Cluster-Repository.

Vollständige Repositorys müssen auf Servern eingesetzt werden, die zuverlässig und so hoch wie möglich verfügbar sind und Single Points of Failure vermieden werden müssen. Das Clusterdesign muss immer über zwei vollständige Repositorys verfügen. Wenn ein vollständiges Repository nicht vorhanden ist, kann der Cluster trotzdem ausgeführt werden.

Details zu Aktualisierungen für Clusterressourcen, die von einem Warteschlangenmanager in einem Cluster erstellt werden, z. B. Clusterwarteschlangen, werden von diesem Warteschlangenmanager an die meisten in diesem Cluster an zwei vollständige Repositorys gesendet (oder zu einem Cluster, wenn nur ein vollständiger Warteschlangenmanager im Cluster vorhanden ist). Diese vollständigen Repositorys enthalten die Informationen und geben sie an alle WS-Manager im Cluster weiter, die ein Interesse daran haben (d. a. sie subskribieren). Um sicherzustellen, dass jedes Member des Clusters über eine Up-to-Data-Ansicht der Clusterressourcen verfügt, muss jeder WS-Manager in der Lage sein, mit mindestens einem vollständigen WS-Manager-Repository gleichzeitig zu kommunizieren.

Wenn ein Warteschlangenmanager aus irgendeinem Grund nicht mit vollständigen Repositorys kommunizieren kann, kann er im Cluster abhängig von seiner bereits zwischengespeicherten Informationsstufe für

einen Zeitraum weiter funktionieren, aber es sind keine neuen Aktualisierungen oder der Zugriff auf zuvor nicht verwendete Clusterressourcen verfügbar.

Aus diesem Grund müssen Sie das Ziel haben, die beiden vollständigen Repositorys immer verfügbar zu halten. Diese Anordnung bedeutet jedoch nicht, dass extreme Maßnahmen ergriffen werden müssen, da der Cluster für eine kurze Zeit ohne vollständiges Repository ausreichend funktioniert.

Es gibt einen weiteren Grund, dass ein Cluster über zwei vollständige WS-Manager-Repository-WS-Manager verfügen muss, die nicht die Verfügbarkeit von Clusterinformationen sind: Dieser Grund besteht darin, sicherzustellen, dass die Clusterinformationen, die im vollständigen Repository-Cache gespeichert sind, an zwei Stellen zu Wiederherstellungszwecken vorhanden sind. Wenn nur ein vollständiges Repository vorhanden ist und seine Informationen zum Cluster verloren gehen, ist ein manueller Eingriff auf alle WS-Manager im Cluster erforderlich, damit der Cluster wieder funktionieren kann. Wenn jedoch zwei vollständige Repositorys vorhanden sind, weil die Informationen immer in zwei vollständigen Repositorys veröffentlicht und subskribiert werden, kann das Repository für fehlgeschlagene vollständige Repositorys mit dem Minimum an Aufwand wiederhergestellt werden.

- Es ist möglich, die Verwaltung von WS-Managern mit vollem Repository in einem zwei vollständigen Repository-Cluster-Design auszuführen, ohne die Benutzer des Clusters zu beeinträchtigen: Der Cluster funktioniert weiterhin mit nur einem Repository, so dass die Repositorys nach unten gebracht, die Wartung angewendet und ein weiteres Mal wieder gesichert werden kann. Selbst wenn ein Ausfall im zweiten vollständigen Repository vorhanden ist, wird die Ausführung von Anwendungen für mindestens drei Tage nicht beeinträchtigt.
- Es sei denn, es gibt einen guten Grund für die Verwendung eines dritten Repositorys, wie z. B. die Verwendung eines geographisch lokalen vollständigen Repositorys aus geographischen Gründen, die Verwendung der beiden Repository-Designs. Die Verwendung von drei vollständigen Repositorys bedeutet, dass Sie nie wissen, welche beiden derzeit verwendet werden, und es kann zu Verwaltungsproblemen kommen, die durch die Interaktionen zwischen mehreren Workload-Management-Parametern verursacht werden. Es wird nicht empfohlen, mehr als zwei vollständige Repositorys zu verwenden.
- Wenn Sie nach wie vor eine bessere Verfügbarkeit benötigen, sollten Sie die vollständigen WS-Manager-Repositorys als Multi-Instanz-Warteschlangenmanager oder plattformspezifische Hochverfügbarkeitsunterstützung verwenden, um die Verfügbarkeit zu verbessern.
- Sie müssen alle vollständigen WS-Manager-Repository-Warteschlangenmanager vollständig mit manuell definierten Clustersenderkanälen verbinden. Es ist besonders darauf zu achten, dass der Cluster aus einem vertretbaren Grund mehr als zwei vollständige Repositorys hat. In dieser Situation ist es oft möglich, einen oder mehrere Kanäle zu verpassen und dafür nicht sofort erkennbar zu sein. Wenn keine vollständige Verbindung auftritt, treten häufig Probleme bei der Diagnose von Problemen auf. Sie sind schwer zu diagnostizieren, da einige vollständige Repositorys nicht alle Repositorydaten enthalten und daher in Abhängigkeit von den vollständigen Repositorys, zu denen sie eine Verbindung herstellen, zu Warteschlangenmanagern im Cluster mit unterschiedlichen Sichten des Clusters führt.

Sollen Anwendungen Warteschlangen in vollständigen Repositorys verwenden?

Ein vollständiges Repository ist in der meisten Hinsicht genau wie jeder andere Warteschlangenmanager, und es ist daher möglich, Anwendungswarteschlangen im vollständigen Repository zu hosten und Anwendungen direkt mit diesen WS-Managern zu verbinden. Sollen Anwendungen Warteschlangen in vollständigen Repositorys verwenden?

Die allgemein akzeptierte Antwort lautet "Nein?". Obwohl diese Konfiguration möglich ist, bevorzugen viele Kunden, diese Warteschlangenmanager für die Verwaltung des gesamten Repository-Cluster-Caches zu halten. Punkte, die bei der Entscheidung für eine der beiden Optionen zu berücksichtigen sind, werden hier beschrieben, aber letztendlich muss die Clusterarchitektur den besonderen Anforderungen der Umgebung gerecht werden.

- Upgrades: Um neue Clusterfunktionen in neuen Releases von IBM MQ nutzen zu können, müssen normalerweise zuerst ein Upgrade der Warteschlangenmanager mit vollständigem Repository in diesem Cluster durchführen. Wenn eine Anwendung im Cluster neue Funktionen verwenden möchte, kann es nützlich sein, die vollständigen Repositorys (und eine Teilmenge von Teilrepositorys) zu aktualisieren, ohne eine Reihe von kolokationsfähigen Anwendungen testen zu müssen.

- Wartung: Auf ähnliche Weise, wenn Sie eine dringende Wartung auf die vollständigen Repositorys anwenden müssen, können sie mit dem Befehl **REFRESH** erneut gestartet oder aktualisiert werden, ohne Anwendungen zu berühren.
- Leistung: Wenn Cluster wachsen und die Anforderungen an die vollständige Repository-Cluster-Cache-Wartung größer werden, verringert sich die Gefahr, dass die Anwendungsleistung durch Konkurrenzsituationen getrennt wird, die die Anwendungsleistung beeinträchtigen, da die Systemressourcen in Konflikt stehen.
- Hardwarevoraussetzungen: Normalerweise müssen vollständige Repositorys nicht besonders leistungsfähig sein, z. B. ist ein einfacher UNIX-Server mit einer erwartungsgemäß guten Verfügbarkeit ausreichend. Alternativ dazu muss bei sehr großen oder sich ständig verändernden Clustern die Leistung des gesamten Repository-Computers berücksichtigt werden.
- Softwarevoraussetzungen: Anforderungen sind in der Regel der Hauptgrund für die Auswahl von Anwendungs-Warteschlangen in einem vollständigen Repository. In einem kleinen Cluster kann die Kollokation eine Voraussetzung für weniger WS-Manager/Server über alle bedeuten.

Kanaldefinitionen verwalten

Selbst in einem einzigen Cluster können mehrere Kanaldefinitionen vorhanden sein, die mehrere Routen zwischen zwei Warteschlangenmanagern enthalten.

Es gibt manchmal einen Vorteil, parallele Kanäle in einem einzigen Cluster zu haben, aber diese Designentscheidung muss gründlich durchdacht werden; abgesehen von der Komplexität der Komplexität kann dieses Design dazu führen, dass Kanäle untergenutzt werden, die die Leistung verringern. Diese Situation tritt auf, da beim Testen normalerweise viele Nachrichten mit einer konstanten Rate gesendet werden, so dass die parallelen Kanäle vollständig verwendet werden. Bei real-world-Bedingungen eines nicht konstanten Nachrichtenstroms bewirkt der Lastausgleichsalgorithmus jedoch, dass die Leistung sinkt, wenn der Nachrichtenfluss von Kanal zu Kanal umgeschaltet wird.

Wenn ein Warteschlangenmanager mehreren Clustern angehört, besteht die Möglichkeit, eine einzelne Kanaldefinition mit einer Clusternamensliste zu verwenden, anstatt einen separaten CLUSRCVR -Kanal für jeden Cluster zu definieren. Diese Konfiguration kann jedoch später zu Verwaltungsschwierigkeiten führen. Dies kann beispielsweise der Fall sein, wenn TLS auf einen Cluster, aber nicht auf einen zweiten Cluster angewendet werden soll. Es ist daher vorzuziehen, separate Definitionen zu erstellen, und die in „Namenskonventionen für Cluster” auf Seite 36 vorgeschlagene Namenskonvention unterstützt dies.

Lastausgleich über mehrere Kanäle

Diese Informationen sind als ein fortgeschrittenes Verständnis des Subjektes gedacht. Die grundlegende Erläuterung zu diesem Thema (die vor der Verwendung der Informationen hier zu verstehen ist) finden Sie im Abschnitt Cluster für das Workload-Management verwenden , Lastausgleich in Clustern und Algorithmus für Clusterauslastungsmanagement .

Der Algorithmus für die Clusterauslastungsverwaltung stellt eine große Gruppe von Tools zur Verfügung, aber sie dürfen nicht alle zusammen verwendet werden, ohne dass die Funktionsweise und die Interaktion vollständig verstanden werden müssen. Es ist möglicherweise nicht sofort ersichtlich, wie wichtig die Kanäle für den Lastausgleich sind: Der Algorithmus für die Verarbeitung des Workload-Managements verhält sich so, als ob mehrere Cluster-Kanäle zu einem Warteschlangenmanager, der Eigner einer Clusterwarteschlange ist, als mehrere Instanzen dieser Warteschlange behandelt werden. Dieser Prozess wird im folgenden Beispiel ausführlicher erläutert:

1. Es gibt zwei Warteschlangenmanager, die eine Warteschlange in einem Cluster hosten: QM1 und QM2.
2. Es gibt fünf Clusterempfängerkanäle für QM1.
3. Es gibt nur einen Clusterempfängerkanal für QM2.
4. Wenn **MQPUT** oder **MQOPEN** on QM3 eine Instanz auswählt, ist es fünf Mal wahrscheinlicher, dass der Algorithmus die Nachricht an QM1 sendet als an QM2.
5. Die Situation in Schritt 4 tritt auf, weil der Algorithmus sechs Optionen zur Auswahl vorsieht (5 + 1) und Round-Robin über alle fünf Kanäle auf QM1 und den einzelnen Kanal auf QM2.

Eine weitere kluge Funktion besteht darin, dass IBM MQ selbst beim Einreihen von Nachrichten in eine Clusterwarteschlange, für die auf dem lokalen Warteschlangenmanager zufällig eine Instanz konfiguriert ist, den Status des lokalen Clusterempfängerkanals verwendet, um zu entscheiden, ob Nachrichten in die lokale Instanz der Warteschlange oder in ferne Instanzen der Warteschlange gestellt werden sollen. In diesem Szenario gilt Folgendes:

1. Beim Einreihen von Nachrichten sieht der Algorithmus für die Auslastungsverwaltung nicht die einzelnen Clusterwarteschlangen an, sondern die Clusterkanäle, die diese Ziele erreichen können.
2. Um die lokalen Zieladressen zu erreichen, werden die lokalen Empfängerkanäle in diese Liste aufgenommen (obwohl sie nicht zum Senden der Nachricht verwendet werden).
3. Wenn ein lokaler Empfängerkanal gestoppt wird, bevorzugt der Workload-Management-Algorithmus eine alternative Instanz bevorzugt, wenn die CLUSRCVR nicht gestoppt ist. Wenn mehrere lokale CLUSRCVR-Instanzen für das Ziel vorhanden sind und mindestens eine Instanz nicht gestoppt ist, bleibt die lokale Instanz berechtigt.

Clustering: Anwendungsisolation mit mehreren Clusterübertragungswarteschlangen
Sie können die Nachrichtenflüsse zwischen Warteschlangenmanagern in einem Cluster isolieren. Sie können Nachrichten, die von verschiedenen Clustersenderkanälen transportiert werden, in verschiedene Clusterübertragungswarteschlangen stellen. Sie können den Ansatz in einem einzelnen Cluster oder mit überlappenden Clustern verwenden. Das Thema enthält Beispiele und einige bewährte Verfahren, die Sie bei der Auswahl eines zu verwendenden Ansatzes führen.

Bei der Bereitstellung einer Anwendung können Sie entscheiden, welche IBM MQ-Ressourcen die Anwendung mit anderen gemeinsam nutzen soll und welche nicht. Es gibt eine Reihe von Typen von Ressourcen, die gemeinsam genutzt werden können, wobei die Haupttypen der Server selbst, der Warteschlangenmanager, die Kanäle und die Warteschlangen sind. Sie können Anwendungen mit weniger gemeinsam genutzten Ressourcen konfigurieren. Sie können separate Warteschlangen, Kanäle, Warteschlangenmanager oder sogar Server für einzelne Anwendungen zuordnen. Wenn Sie dies tun, wird die Gesamtsystemkonfiguration größer und komplexer. Die Verwendung von IBM MQ-Clustern reduziert die Komplexität der Verwaltung von mehr Servern, Warteschlangenmanagern, Warteschlangen und Kanälen, führt aber eine weitere gemeinsam genutzte Ressource ein, die Cluster-Übertragungswarteschlange SYSTEM.CLUSTER.TRANSMIT.QUEUE.

Abbildung 12 auf Seite 52 ist eine Schicht aus einer großen IBM MQ-Implementierung, die die Bedeutung der gemeinsamen Nutzung von SYSTEM.CLUSTER.TRANSMIT.QUEUEveranschaulicht. Im Diagramm ist die Anwendung `Client App` mit dem Warteschlangenmanager QM2 im Cluster CL1 verbunden. Eine Nachricht von `Client App` wird von der Anwendung `Server App` verarbeitet. Die Nachricht wird von `Server App` aus der Clusterwarteschlange Q1 auf dem Warteschlangenmanager QM3 in CLUSTER2abgerufen. Da sich die Client- und Serveranwendungen nicht in demselben Cluster befinden, wird die Nachricht vom Gateway-Warteschlangenmanager QM1 übertragen.

Der normale Weg zur Konfiguration eines Cluster-Gateways besteht darin, den Gateway-Warteschlangenmanager zu einem Mitglied aller Cluster zu machen. Auf dem Gateway-WS-Manager werden Clusteraliasnamen für Clusterwarteschlangen in allen Clustern definiert. Die Aliasnamen für Clusterwarteschlangen sind in allen Clustern verfügbar. Nachrichten, die an die Clusterwarteschlangenaliasnamen gestellt werden, werden über den Gateway-Warteschlangenmanager an ihr korrektes Ziel weitergeleitet. Der Gateway-Warteschlangenmanager reiht Nachrichten, die an die Aliase der Clusterwarteschlangen gesendet werden, in die allgemeine SYSTEM.CLUSTER.TRANSMIT.QUEUE unter QM1 ein.

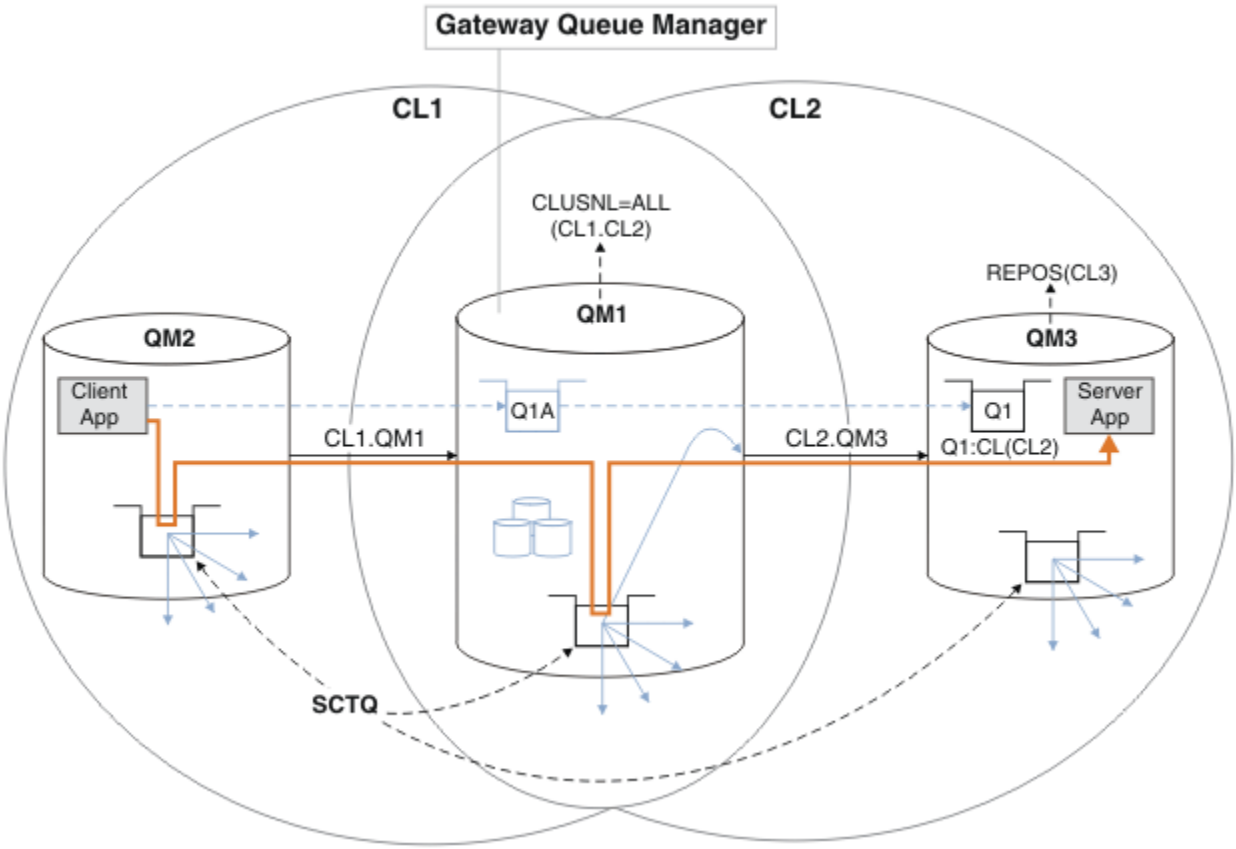
Die Hub-und Spoke-Architektur erfordert alle Nachrichten zwischen Clustern, die über den Gateway-Warteschlangenmanager übergeben werden. Das Ergebnis ist, dass alle Nachrichten durch die Warteschlange des einzelnen Clusters auf QM1, SYSTEM.CLUSTER.TRANSMIT.QUEUE fließen.

Aus einer Leistungsperspektive ist eine einzelne Warteschlange kein Problem. Eine allgemeine Übertragungswarteschlange stellt in der Regel keinen Leistungsengpass dar. Der Nachrichtendurchsatz auf dem Gateway wird weitgehend durch die Leistung der Kanäle bestimmt, die eine Verbindung zu ihm herstellen. Der Durchsatz wird in der Regel nicht von der Anzahl der Warteschlangen oder der Anzahl der Nachrichten in den Warteschlangen, die die Kanäle verwenden, beeinflusst.

Aus einigen anderen Perspektiven hat die Verwendung einer einzelnen Übertragungswarteschlange für mehrere Anwendungen Nachteile:

- Sie können den Fluss von Nachrichten nicht in ein Ziel vom Nachrichtenfluss zu einem anderen Ziel eingrenzen. Sie können die Speicherung von Nachrichten nicht trennen, bevor sie weitergeleitet werden, selbst wenn sich die Ziele in verschiedenen Clustern auf verschiedenen Warteschlangenmanagern befinden.

  Wenn eine Cluster-Destination nicht mehr verfügbar ist, werden Nachrichten für diese Zieladresse in der einzelnen Übertragungswarteschlange, und schließlich füllen sie die Nachrichten aus. Sobald die Übertragungswarteschlange voll ist, stoppt sie Nachrichten, die in die Übertragungswarteschlange für ein beliebtes Clusterziel gestellt werden.
- Es ist nicht einfach, die Übertragung von Nachrichten an verschiedene Cluster-Destinations zu überwachen. Alle Nachrichten befinden sich in der einzelnen Übertragungswarteschlange. Wenn Sie die Länge der Übertragungswarteschlange anzeigen, wird wenig darüber informiert, ob Nachrichten an alle Zieladressen übertragen werden.



Anmerkung: Die Pfeile in Abbildung 12 auf Seite 52 und die folgenden Abbildungen sind von unterschiedlichen Typen. Feste Pfeile stellen Nachrichtenflüsse dar. Bei den Beschriftungen auf festen Pfeilen handelt es sich um Nachrichtenkanalnamen. Die grauen ausgefüllten Pfeile sind potenzielle Nachrichtenflüsse von SYSTEM.CLUSTER.TRANSMIT.QUEUE auf Clustersenderkanälen. Schwarze gestrichelte Linien verbinden Beschriftungen mit ihren Zielen. Graue gestrichelte Pfeile sind Referenzen, z. B. von einem MQOPEN Aufruf von `Client` App an die Clusteraliaswarteschlangendefinition Q1A.

Abbildung 12. In Hub- und Spoke-Architektur mit IBM MQ-Clustern implementierte Client/Server-Anwendung

In Abbildung 12 auf Seite 52 öffnen Clients von `Server` App die Warteschlange Q1A. Nachrichten werden unter QM2 in SYSTEM.CLUSTER.TRANSMIT.QUEUE eingereiht, unter QM1 in SYSTEM.CLUSTER.TRANSMIT.QUEUE übertragen und anschließend unter QM3 in Q1 übertragen, wo sie von der Anwendung `Server` App empfangen werden.

Die Nachricht von `Client` App wird über die Clusterübertragungswarteschlangen des Systems durch QM2 und QM1übergeben. In Abbildung 12 auf Seite 52 besteht das Ziel darin, den Nachrichtenfluss auf

dem Gateway-Warteschlangenmanager von der Clientanwendung zu isolieren, damit seine Nachrichten nicht in SYSTEM.CLUSTER.TRANSMIT.QUEUE gespeichert werden. Sie können Flüsse auf einem der anderen Clusterwarteschlangenmanager isolieren. Sie können auch Flüsse in die andere Richtung isolieren, zurück an den Client. Um die Beschreibungen der Lösungen kurz zu halten, betrachten die Beschreibungen nur einen einzigen Nachrichtenfluss von der Clientanwendung.

Lösungen für die Isolierung des Clusternachrichtenverkehrs auf einem Cluster-Gateway-Warteschlangenmanager

Eine Möglichkeit, das Problem zu lösen, besteht darin, WS-Manager-Aliasnamen oder ferne Warteschlangendefinitionen zu verwenden, um eine Brücke zwischen Clustern zu schlagen. Erstellen Sie eine in Gruppen zusammengefasste ferne Warteschlangendefinition, eine Übertragungswarteschlange und einen Kanal, um die einzelnen Nachrichtenflüsse auf dem Gateway-Warteschlangenmanager zu trennen. Weitere Informationen finden Sie im Abschnitt Definition einer fernen Warteschlange zum Isolieren von Nachrichten, die von einem Gateway-Warteschlangenmanager gesendet werden hinzufügen.

Ab IBM WebSphere MQ 7.5 sind Clusterwarteschlangenmanager nicht auf die Verwendung einer einzigen Clusterübertragungswarteschlange beschränkt. Sie haben zwei Möglichkeiten:

1. Definieren Sie zusätzliche Clusterübertragungswarteschlangen manuell, und definieren Sie, welche Clustersenderkanäle Nachrichten aus jeder Übertragungswarteschlange übertragen. Weitere Informationen finden Sie im Abschnitt Clustersendewarteschlange zum Isolieren des von einem Gateway-Warteschlangenmanager gesendeten Clusternachrichtenverkehrs hinzufügen .
2. Ermöglichen Sie dem WS-Manager, zusätzliche Clusterübertragungswarteschlangen automatisch zu erstellen und zu verwalten. Sie definiert eine andere Clusterübertragungswarteschlange für jeden Clustersenderkanal. Weitere Informationen finden Sie im Abschnitt Standardwert in separate Clusterübertragungswarteschlangen ändern, um den Nachrichtendatenverkehr zu isolieren .

Sie können manuell definierte Clusterübertragungswarteschlangen für einige Clustersenderkanäle kombinieren, wobei der Warteschlangenmanager den Rest verwaltet. Bei der Kombination von Übertragungswarteschlangen handelt es sich um den Ansatz, der im Abschnitt Clustersendungswarteschlange zum Isolieren des von einem Gateway-Warteschlangenmanager gesendeten Clusternachrichtenverkehrs hinzugefügt wird . In dieser Lösung verwenden die meisten Nachrichten zwischen Clustern SYSTEM.CLUSTER.TRANSMIT.QUEUE allgemein. Eine Anwendung ist kritisch, und alle ihre Nachrichtenflüsse werden von anderen Nachrichtenflüssen isoliert, indem eine manuell definierte Clusterübertragungswarteschlange verwendet wird.

Die Konfiguration im Abschnitt Clustersendungswarteschlange zum Isolieren von Clusternachrichtenverkehr, die von einem Gateway-Warteschlangenmanager gesendet werden, wird begrenzt hinzugefügt ist begrenzt. Der Nachrichtendatenverkehr, der in eine Clusterwarteschlange auf demselben WS-Manager in demselben Cluster wie eine andere Clusterwarteschlange läuft, wird nicht getrennt. Sie können den Nachrichtenverkehr mit Hilfe der Definitionen der fernen Warteschlange, die Teil der verteilten Steuerung von Warteschlangen sind, in einzelne Warteschlangen aufteilen. Bei Clustern, die mehrere Clusterübertragungswarteschlangen verwenden, können Sie den Nachrichtenverkehr trennen, der zu verschiedenen Clustersenderkanälen führt. Mehrere Clusterwarteschlangen im selben Cluster teilen sich auf demselben Warteschlangenmanager einen Clustersenderkanal gemeinsam. Nachrichten für diese Warteschlangen werden in derselben Übertragungswarteschlange gespeichert, bevor sie vom Gateway-WS-Manager weitergeleitet werden. In der Konfiguration unter Cluster-und Clustersendewarteschlange zum Isolieren des von einem Gateway-Warteschlangenmanager gesendeten Clusternachrichtenverkehrs hinzufügen wird die Einschränkung durch die Hinzufügung eines anderen Clusters und durch das Erstellen des WS-Managers und der Clusterwarteschlange zu einem Member des neuen Clusters abgestuft. Der neue Warteschlangenmanager kann der einzige WS-Manager im Cluster sein. Sie können dem Cluster weitere Warteschlangenmanager hinzufügen und denselben Cluster verwenden, um Clusterwarteschlangen auf diesen Warteschlangenmanagern zu isolieren.

Zugehörige Konzepte

„Zugriffssteuerung und mehrere Clusterübertragungswarteschlangen” auf Seite 31
Wählen Sie zwischen drei Prüfmodi aus, wenn eine Anwendung Nachrichten in ferne Clusterwarteschlangen einreiht. Die Modi sind: Prüfung über Fernzugriff gegen die Clusterwarteschlange, lokale Prüfung ge-

gen SYSTEM.CLUSTER.TRANSMIT.QUEUE oder Prüfung gegen lokale Profile für die Clusterwarteschlange oder den Clusterwarteschlangenmanager.

Arbeiten mit Clusterübertragungswarteschlangen und Clustersenderkanälen

„Überlappende Cluster" auf Seite 38
Überlappende Cluster bieten zusätzliche Verwaltungsfunktionen. Verwenden Sie Namenslisten, um die Anzahl der Befehle zu reduzieren, die für die Verwaltung von überlappenden Clustern erforderlich sind.

Zugehörige Tasks

Berechtigung zum Einreihen von Nachrichten in ferne Clusterwarteschlangen berechtigen

Definition einer fernen Warteschlange hinzufügen, um Nachrichten zu isolieren, die von einem Gateway-Warteschlangenmanager gesendet wurden

Cluster-Übertragungswarteschlange zum Isolieren des Clusternachrichtenverkehrs hinzufügen, der von einem Gateway-Warteschlangenmanager gesendet wurde

Cluster und Cluster-Übertragungswarteschlange hinzufügen, um den Datenverkehr der Clusternachrichten zu isolieren, die von einem Gateway-Warteschlangenmanager gesendet werden

Ändern der Standardeinstellung in separate Clusterübertragungswarteschlangen, um den Nachrichtendatenverkehr zu isolieren

Erstellen von zwei überlappenden Clustern mit einem Gateway-Warteschlangenmanager

Nachrichtenpfade zwischen Clustern konfigurieren

Sicherung

Zugehörige Verweise

setmqaut

Clustering: Planung der Konfiguration von Clusterübertragungswarteschlangen
Sie werden durch die Auswahl von Clusterübertragungswarteschlangen geführt. Sie können eine gemeinsame Standardwarteschlange, separate Standardwarteschlangen oder manuell definierte Warteschlangen konfigurieren.

Vorbereitende Schritte

Lesen Sie den Abschnitt „Art der zu verwendenden Clusterübertragungswarteschlange auswählen" auf Seite 57.

Informationen zu diesem Vorgang

Wenn Sie planen, wie ein Warteschlangenmanager für die Auswahl einer Clusterübertragungswarteschlange konfiguriert werden soll, können Sie einige Optionen auswählen.

1. Was ist die Standard-Cluster-Übertragungswarteschlange für die Übertragung von Clusternachrichten?
   a. Eine allgemeine Clusterübertragungswarteschlange, SYSTEM.CLUSTER.TRANSMIT.QUEUE.
   b. Trennen Sie die Clusterübertragungswarteschlangen voneinander. Der WS-Manager verwaltet die separaten Clusterübertragungswarteschlangen. Sie werden als permanent dynamische Warteschlangen aus der Modellwarteschlange SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE erstellt. Er erstellt für jeden Clustersenderkanal, den er verwendet, eine Clusterübertragungswarteschlange.
2. Für die Clusterübertragungswarteschlangen, die Sie manuell erstellen möchten, haben Sie die folgenden beiden Möglichkeiten:
   a. Definieren Sie eine separate Übertragungswarteschlange für jeden Clustersenderkanal, den Sie manuell konfigurieren möchten. Setzen Sie in diesem Fall das Warteschlangenattribut **CLCHNAME** der Übertragungswarteschlange auf den Namen eines Clustersenderkanals. Wählen Sie den Clustersenderkanal aus, der Nachrichten aus dieser Übertragungswarteschlange übertragen soll.
   b. Kombinieren Sie den Nachrichtenverkehr für eine Gruppe von Clustersenderkanälen in derselben Clusterübertragungswarteschlange (siehe Abbildung 13 auf Seite 55). In diesem Fall setzen Sie das Warteschlangenattribut **CLCHNAME** jeder allgemeinen Übertragungswarteschlange auf den Namen eines generischen Clustersenderkanals. Ein generischer Clustersenderkanalname ist ein Filter zum Gruppieren von Namen von Clustersenderkanälen. Beispiel: SALES.* gruppiert alle Cluster-

senderkanäle, deren Namen mit SALES . beginnen. Sie können mehrere Platzhalterzeichen an einer beliebigen Position in der Filterzeichenfolge platzieren. Das Platzhalterzeichen ist ein Stern ("*"). Er steht für eine Zahl von null bis zu einer beliebigen Anzahl von Zeichen.

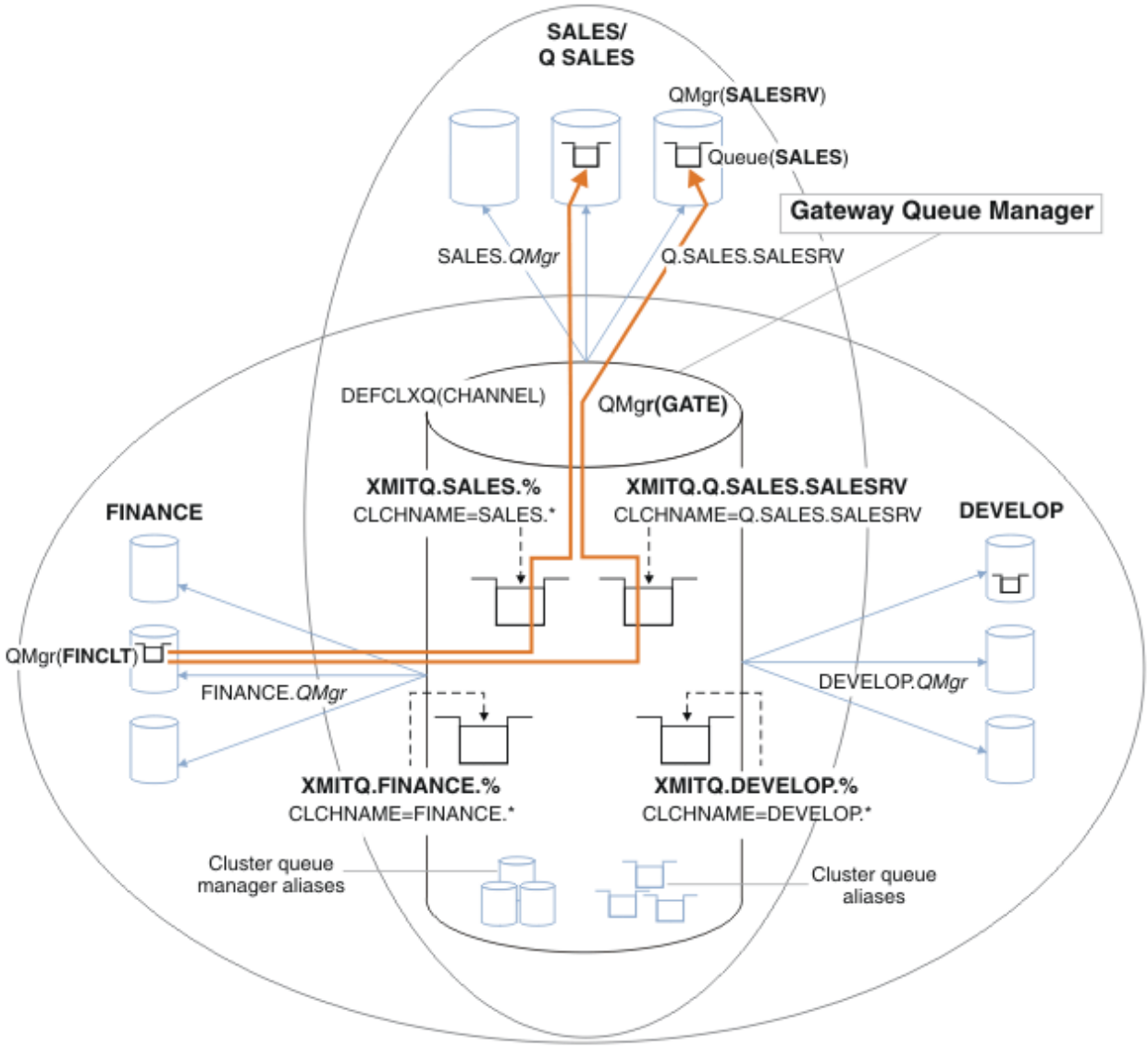


Abbildung 13. Beispiel für eigene Übertragungswarteschlangen für die verschiedenen IBM MQ-Abteilungscluster

Vorgehensweise

1. Wählen Sie den Typ der Standardübertragungswarteschlange für Cluster aus, die verwendet werden soll .
   - Wählen Sie eine einzelne Clusterübertragungswarteschlange oder separate Warteschlangen für jede Clusterverbindung aus.

   Übernehmen Sie die Standardeinstellung oder führen Sie den Befehl **MQSC** aus:

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

2. Isolieren Sie alle Nachrichtenflüsse, die keine Cluster-Übertragungswarteschlange mit anderen Flows gemeinsam nutzen dürfen .

- Siehe „Clustering: Beispielkonfiguration mehrerer Clusterübertragungswarteschlangen” auf Seite 59. Im Beispiel ist die SALES-Warteschlange, die isoliert werden muss, ein Mitglied des SALES-Clusters unter SALESRV. Erstellen Sie zum Isolieren der SALES-Warteschlange einen neuen `Q.SALES`-Cluster. Machen Sie den SALESRV-Warteschlangenmanager zu einem Mitglied und ändern die SALES-Warteschlange so, dass sie zu `Q.SALES` gehört.
- Warteschlangenmanager, die Nachrichten an SALES senden, müssen ebenfalls Mitglieder des neuen Clusters sein. Wenn Sie einen Clusterwarteschlangenaliasnamen und einen Gateway-Warteschlangenmanager verwenden, wie im Beispiel in vielen Fällen, können Sie die Änderungen begrenzen, um den Gateway-Warteschlangenmanager zu einem Mitglied des neuen Clusters zu machen.
- Durch die Trennung der Flüsse vom Gateway zum Ziel werden jedoch keine Flüsse in das Gateway vom Quellenwarteschlangenmanager getrennt. Aber es erweist sich manchmal als ausreichend, um die Abläufe vom Gateway zu trennen und nicht zum Gateway zu fließen. Wenn dies nicht ausreichend ist, fügen Sie den Quellenwarteschlangenmanager in den neuen Cluster ein. Wenn Nachrichten über das Gateway übertragen werden sollen, versetzen Sie den Clusteralias in den neuen Cluster, und senden Sie weiterhin Nachrichten an den Clusteralias auf dem Gateway und nicht direkt an den Ziel-WS-Manager.

Führen Sie die folgenden Schritte aus, um Nachrichtenflüsse zu isolieren:

a) Konfigurieren Sie die Ziele der Flows so, dass jede Zielwarteschlange die einzige Warteschlange in einem bestimmten Cluster ist, auf diesem Warteschlangenmanager .

b) Erstellen Sie die Cluster-Sender- und Clusterempfängerkanäle für alle neuen Cluster, die Sie nach einer systematischen Namenskonvention erstellt haben .

- Siehe „Clustering: Besondere Hinweise zu überlappenden Clustern” auf Seite 46.

c) Definieren Sie eine Clusterübertragungswarteschlange für jedes isolierte Ziel auf jedem Warteschlangenmanager, der Nachrichten an die Zielwarteschlange sendet.

- Eine Namenskonvention für Clusterübertragungswarteschlangen besteht darin, den Attributwert des Clusterkanalnamens `CLCHNAME` mit dem Präfix `XMITQ.` zu verwenden

3. Erstellen Sie Clusterübertragungswarteschlangen, um die Governance- oder Monitoring-Voraussetzungen zu erfüllen .

- Typische Governance-und Überwachungsanforderungen führen zu einer Übertragungswarteschlange pro Cluster oder einer Übertragungswarteschlange pro Warteschlangenmanager. Wenn Sie die Namenskonvention für Clusterkanäle (*`ClusterName.QueueManagerName`*) befolgen, ist es einfach, generische Kanalnamen zu erstellen, die einen Cluster von Warteschlangenmanagern oder alle Cluster auswählen, zu denen ein Warteschlangenmanager gehört. (Siehe „Clustering: Beispielkonfiguration mehrerer Clusterübertragungswarteschlangen” auf Seite 59.)
- Erweitern Sie die Namenskonvention für Clusterübertragungswarteschlangen, um generische Kanalnamen zu verwenden, indem Sie das Sternsymbol durch ein Prozentzeichen ersetzen. Beispiel:

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

Zugehörige Konzepte

Arbeiten mit Clusterübertragungswarteschlangen und Clustersenderkanälen

„Zugriffssteuerung und mehrere Clusterübertragungswarteschlangen” auf Seite 31
Wählen Sie zwischen drei Prüfmodi aus, wenn eine Anwendung Nachrichten in ferne Clusterwarteschlangen einreiht. Die Modi sind: Prüfung über Fernzugriff gegen die Clusterwarteschlange, lokale Prüfung gegen `SYSTEM.CLUSTER.TRANSMIT.QUEUE` oder Prüfung gegen lokale Profile für die Clusterwarteschlange oder den Clusterwarteschlangenmanager.

„Überlappende Cluster” auf Seite 38
Überlappende Cluster bieten zusätzliche Verwaltungsfunktionen. Verwenden Sie Namenslisten, um die Anzahl der Befehle zu reduzieren, die für die Verwaltung von überlappenden Clustern erforderlich sind.

Zugehörige Tasks

Definition einer fernen Warteschlange hinzufügen, um Nachrichten zu isolieren, die von einem Gateway-Warteschlangenmanager gesendet wurden

Cluster-Übertragungswarteschlange zum Isolieren des Clusternachrichtenverkehrs hinzufügen, der von einem Gateway-Warteschlangenmanager gesendet wurde

Cluster und Cluster-Übertragungswarteschlange hinzufügen, um den Datenverkehr der Clusternachrichten zu isolieren, die von einem Gateway-Warteschlangenmanager gesendet werden

Ändern der Standardeinstellung in separate Clusterübertragungswarteschlangen, um den Nachrichtendatenverkehr zu isolieren

Erstellen von zwei überlappenden Clustern mit einem Gateway-Warteschlangenmanager

Nachrichtenpfade zwischen Clustern konfigurieren

Art der zu verwendenden Clusterübertragungswarteschlange auswählen
Vorgehensweise zur Auswahl zwischen verschiedenen Konfigurationsoptionen für die Clusterübertragungswarteschlange.

Sie können auswählen, welche Clusterübertragungswarteschlange einem Clustersenderkanal zugeordnet ist.

1. Sie können alle Clustersenderkanäle der einzelnen Standardclusterübertragungswarteschlange SYSTEM.CLUSTER.TRANSMIT.QUEUEzuordnen. Diese Option ist die Standardeinstellung.
2. Sie können alle Clustersenderkanäle so festlegen, dass sie automatisch einer separaten Clusterübertragungswarteschlange zugeordnet werden. Die Warteschlangen werden vom Warteschlangenmanager aus der Modellwarteschlange SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE mit dem Namen SYSTEM.CLUSTER.TRANSMIT. *ChannelName* erstellt. Kanäle verwenden ihre eindeutig benannte Clusterübertragungswarteschlange, wenn das Warteschlangenmanagerattribut **DEFCLXQ** auf CHANNEL gesetzt ist.
3. Sie können bestimmte Clustersenderkanäle festlegen, die von einer einzelnen Clusterübertragungswarteschlange bedient werden sollen. Wählen Sie diese Option aus, indem Sie eine Übertragungswarteschlange erstellen und ihr Attribut **CLCHNAME** auf den Namen des Clustersenderkanals einstellen.
4. Sie können Gruppen von Clustersenderkanälen auswählen, die von einer einzelnen Clusterübertragungswarteschlange bedient werden sollen. Wählen Sie diese Option aus, indem Sie eine Übertragungswarteschlange erstellen und ihr Attribut **CLCHNAME** auf einen generischen Kanalnamen wie *ClusterName*.* einstellen. Wenn Sie Clusterkanäle nach den Namenskonventionen in „Clustering: Besondere Hinweise zu überlappenden Clustern” auf Seite 46benennen, wählt dieser Name alle Clusterkanäle aus, die mit Warteschlangenmanagern im Cluster *ClusterName*verbunden sind.

Sie können eine der Standardoptionen für die Clusterübertragungswarteschlange für einige Clustersenderkanäle mit einer beliebigen Anzahl spezifischer und generischer Cluster-Übertragungswarteschlangen-Konfigurationen kombinieren.

Bewährte Verfahren

In den meisten Fällen ist bei vorhandenen IBM MQ-Installationen die Standardkonfiguration die beste Wahl. Ein Clusterwarteschlangenmanager speichert Clusternachrichten in einer einzelnen Clusterübertragungswarteschlange, SYSTEM.CLUSTER.TRANSMIT.QUEUE. Sie haben die Möglichkeit, den Standardwert zum Speichern von Nachrichten für verschiedene Warteschlangenmanager und verschiedene Cluster in separaten Übertragungswarteschlangen oder für die Definition eigener Übertragungswarteschlangen zu ändern.

In den meisten Fällen ist bei neuen IBM MQ-Installationen die Standardkonfiguration auch die beste Wahl. Der Prozess der Umschaltung von der Standardkonfiguration auf die alternative Standardkonfiguration mit einer Übertragungswarteschlange für jeden Clustersenderkanal erfolgt automatisch. Die Umschaltung erfolgt ebenfalls automatisch. Die Auswahl der einen oder der anderen ist nicht kritisch, Sie können sie umkehren.

Der Grund für die Auswahl einer anderen Konfiguration ist eher mit Governance und Management zu tun, als mit Funktionalität oder Leistung. Bei einigen Ausnahmebedingungen wird das Verhalten des WS-Managers nicht von der Konfiguration mehrerer Clusterübertragungswarteschlangen unterstützt. Sie führt zu mehr Warteschlangen und erfordert, dass Sie die Überwachungs-und Managementprozeduren, die Sie bereits konfiguriert haben, ändern müssen, die sich auf die einzelne Übertragungswarteschlange

beziehen. Aus diesem Grund ist der Rest der Standardkonfiguration die beste Wahl, es sei denn, Sie verfügen über starke Governance-oder Verwaltungsgründe für eine andere Auswahl.

Die Ausnahmen beziehen sich beide darauf, was passiert, wenn die Anzahl der in SYSTEM.CLUSTER.TRANSMIT.QUEUE gespeicherten Nachrichten zunimmt. Wenn Sie alle Schritte zum Trennen der Nachrichten für ein Ziel von den Nachrichten für ein anderes Ziel verwenden, sollten Kanal-und Zustellungsprobleme mit einem Ziel die Zustellung an ein anderes Ziel nicht beeinträchtigen. Die Anzahl der in SYSTEM.CLUSTER.TRANSMIT.QUEUE gespeicherten Nachrichten kann sich jedoch erhöhen, da Nachrichten nicht schnell genug an ein Ziel zugestellt werden. Die Anzahl der Nachrichten in SYSTEM.CLUSTER.TRANSMIT.QUEUE für ein Ziel kann sich auf die Zustellung von Nachrichten an andere Ziele auswirken.

Um Probleme zu vermeiden, die sich aus der Befüllung einer einzelnen Übertragungswarteschlange ergeben, sollten Sie genügend Kapazität in Ihrer Konfiguration aufbauen. Wenn dann ein Ziel fehlschlägt und ein Nachrichtenrückstand zu erstellen beginnt, haben Sie die Zeit, das Problem zu beheben.

Wenn Nachrichten über einen Hub-Warteschlangenmanager, z. B. ein Cluster-Gateway, weitergeleitet werden, nutzen sie eine gemeinsame Übertragungswarteschlange, SYSTEM.CLUSTER.TRANSMIT.QUEUE. Wenn die Anzahl der in SYSTEM.CLUSTER.TRANSMIT.QUEUE auf dem Gateway-Warteschlangenmanager gespeicherten Nachrichten die maximale Länge erreicht, beginnt der Warteschlangenmanager, neue Nachrichten für die Übertragungswarteschlange zurückzuweisen, bis die Länge abnimmt. Die Überlastung wirkt sich auf Nachrichten für alle Ziele aus, die über das Gateway weitergeleitet werden. Nachrichten sichern die Übertragungswarteschlangen von anderen Warteschlangenmanagern, die Nachrichten an das Gateway senden. Das Problem manifestiert sich in Nachrichten, die in die Fehlerprotokolle des Warteschlangenmanagers geschrieben werden, den Nachrichtendurchsatz und die abgelaufenen Zeiten zwischen dem Senden einer Nachricht und dem Zeitpunkt, zu dem eine Nachricht an ihrem Zielort ankommt, länger verstrichen sind.

Die Auswirkungen von Überlastung auf eine einzelne Übertragungswarteschlange können sichtbar werden, auch bevor sie voll sind. Wenn Sie einen Mischnachrichtenverkehr mit einigen großen nicht persistenten Nachrichten und einigen kleinen Nachrichten haben, wird die Zeit, in der kleine Nachrichten gesendet werden, mit der Größe der Übertragungswarteschlange erhöht. Die Verzögerung ist darauf zurückzuführen, dass große nicht persistente Nachrichten auf Platte geschrieben werden, die normalerweise nicht auf Platte geschrieben werden. Wenn Sie zeitkritische Nachrichtenflüsse haben, die eine Cluster-Übertragungswarteschlange mit anderen gemischten Nachrichtenflüssen gemeinsam nutzen, könnte es sinnvoll sein, einen speziellen Nachrichtenpfad zu konfigurieren, um ihn von anderen Nachrichtenflüssen zu isolieren. Weitere Informationen finden Sie im Abschnitt Cluster-und Clustersendungswarteschlange zum Isolieren des Clusternachrichtenverkehrs, der von einem Gateway-Warteschlangenmanager gesendet wird , hinzufügen.

Die anderen Gründe für die Konfiguration getrennter Clusterübertragungswarteschlangen sind die Erfüllung der Governance-Anforderungen oder die Vereinfachung der Überwachung von Nachrichten, die an verschiedene Clusterziele gesendet werden. Möglicherweise müssen Sie beispielsweise nachweisen, dass Nachrichten für ein Ziel nie eine Übertragungswarteschlange mit Nachrichten für ein anderes Ziel gemeinsam nutzen.

Ändern Sie das Warteschlangenmanagerattribut **DEFCLXQ**, das die Standardclusterübertragungswarteschlange steuert, um für jeden Clustersenderkanal unterschiedliche Clusterübertragungswarteschlangen zu erstellen. Mehrere Ziele können einen Clustersenderkanal gemeinsam nutzen, sodass Sie Ihre Cluster so planen müssen, dass sie dieses Ziel vollständig erfüllen. Wenden Sie die Methode Cluster-und Clusterübertragungswarteschlange hinzufügen an, um den Clusternachrichtenverkehr, der von einem Gateway-Warteschlangenmanager gesendet wird, systematisch auf alle Ihre Clusterwarteschlangen zu isolieren. Das Ergebnis, das Sie anstreben, ist es, dass keine Cluster-Destination einen Clustersenderkanal mit einem anderen Clusterziel gemeinsam nutzen kann. Dies hat zur Folge, dass keine Nachricht für eine Cluster-Destination ihre Clusterübertragungswarteschlange mit einer Nachricht für ein anderes Ziel gemeinsam nutzt.

Wenn Sie eine separate Clusterübertragungswarteschlange für einen bestimmten Nachrichtenfluss erstellen, ist es einfach, den Nachrichtenfluss zu diesem Ziel zu überwachen. Wenn Sie eine neue Clusterübertragungswarteschlange verwenden möchten, definieren Sie die Warteschlange, ordnen Sie sie einem Clustersenderkanal zu, und stoppen Sie den Kanal und starten Sie ihn. Die Änderung muss nicht perma-

nent sein. Sie können einen Nachrichtenfluss für eine Weile isolieren, die Übertragungswarteschlange überwachen und anschließend die Standardübertragungswarteschlange erneut verwenden.

Zugehörige Tasks

Clustering: Beispielkonfiguration mehrerer Clusterübertragungswarteschlangen
In dieser Task wenden Sie die Schritte zum Planen mehrerer Clusterübertragungswarteschlangen auf drei sich überlappende Cluster an. Die Anforderungen bestehen darin, Nachrichten von allen anderen Nachrichtenflüssen in eine Clusterwarteschlange zu trennen und Nachrichten für verschiedene Cluster in verschiedenen Clusterübertragungswarteschlangen zu speichern.

Clustering: Clusterübertragungswarteschlangen wechseln
Planen Sie, wie die Änderungen an den Clusterübertragungswarteschlangen eines vorhandenen Produktionswarteschlangenmanagers in Kraft gebracht werden.

Clustering: Beispielkonfiguration mehrerer Clusterübertragungswarteschlangen
In dieser Task wenden Sie die Schritte zum Planen mehrerer Clusterübertragungswarteschlangen auf drei sich überlappende Cluster an. Die Anforderungen bestehen darin, Nachrichten von allen anderen Nachrichtenflüssen in eine Clusterwarteschlange zu trennen und Nachrichten für verschiedene Cluster in verschiedenen Clusterübertragungswarteschlangen zu speichern.

Informationen zu diesem Vorgang

Die Schritte in dieser Übung zeigen, wie die Prozedur in „Clustering: Planung der Konfiguration von Clusterübertragungswarteschlangen" auf Seite 54 angewendet wird und die in Abbildung 14 auf Seite 60 gezeigte Konfiguration erreicht wird. Es ist ein Beispiel für drei sich überlappende Cluster mit einem Gateway-WS-Manager, der mit separaten Clusterübertragungswarteschlangen konfiguriert ist. Die MQSC-Befehle zum Definieren der Cluster werden in „Beispielcluster erstellen" auf Seite 62 beschrieben.

Für das Beispiel gibt es zwei Anforderungen. Eine davon ist die Trennung des Nachrichtenflusses vom Gateway-Warteschlangenmanager zu der Verkaufsanwendung, die die Verkäufe protokolliert. Der zweite Punkt ist die Abfrage, wie viele Nachrichten zu einem beliebigen Zeitpunkt darauf warten, an verschiedene Abteilbereiche gesendet zu werden. Die Cluster `SALES`, `FINANCE` und `DEVELOP` sind bereits definiert. Clusternachrichten werden derzeit von `SYSTEM.CLUSTER.TRANSMIT.QUEUE` weitergeleitet.

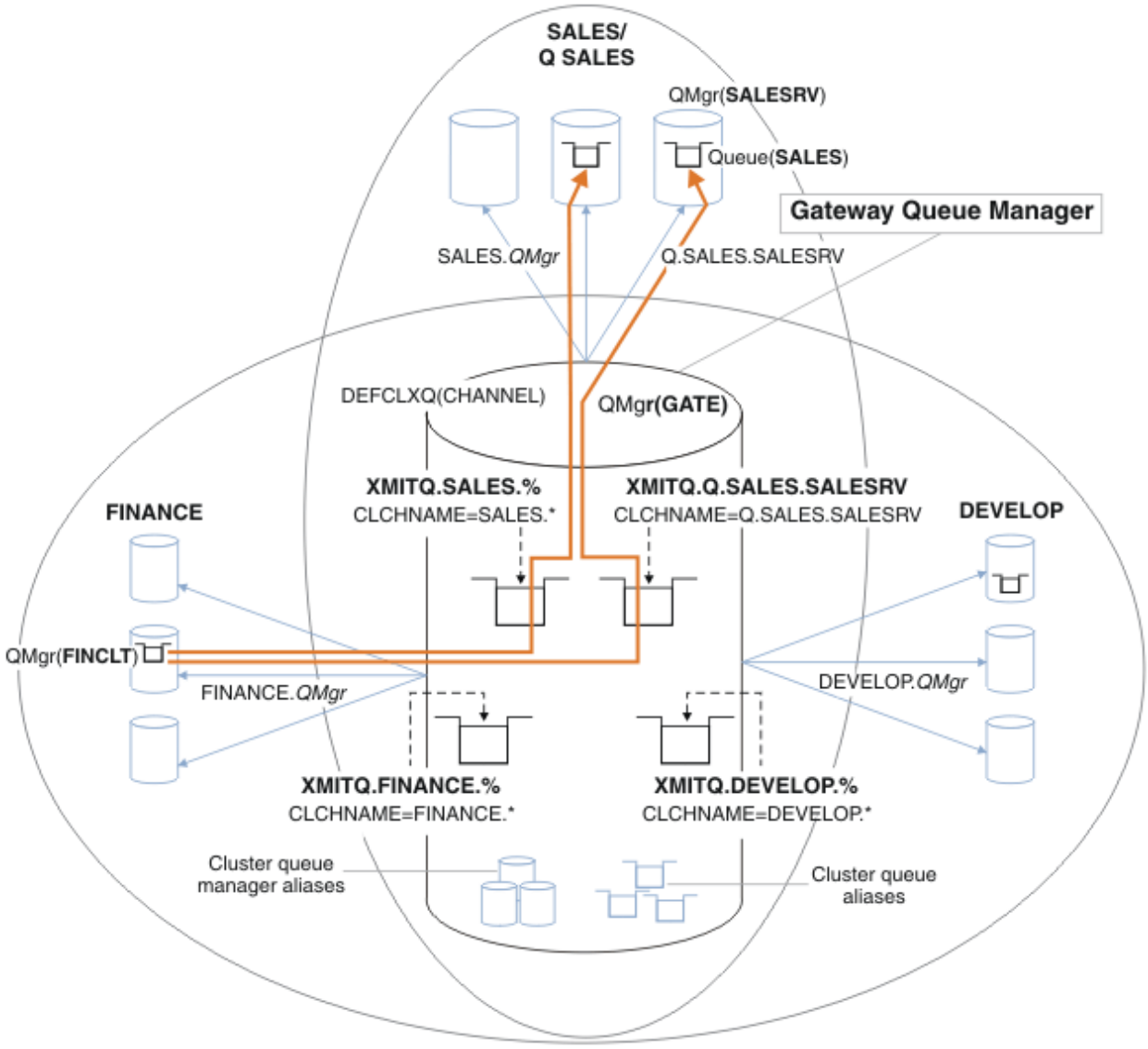


Abbildung 14. Beispiel für eigene Übertragungswarteschlangen für die verschiedenen IBM MQ-Abteilungscluster

Gehen Sie wie folgt vor, um die Cluster zu ändern. Die Definitionen finden Sie unter Änderungen zum Isolieren der Vertriebswarteschlange in einem neuen Cluster und Trennen der Gateway-Clusterübertragungswarteschlangen.

Vorgehensweise

1. Der erste Konfigurationsschritt ist in " Wählen Sie den Typ der Standardübertragungswarteschlange für Cluster aus, die verwendet werden soll ".

   Die Entscheidung besteht darin, separate Standardclusterübertragungswarteschlangen zu erstellen, indem der folgende **MQSC** -Befehl auf dem GATE -Warteschlangenmanager ausgeführt wird.

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

   Es gibt keinen starken Grund für die Auswahl dieser Standardeinstellung, da die Clusterübertragungswarteschlangen manuell definiert werden sollen. Die Auswahl weist einen schwachen Diagnosewert auf. Wenn eine manuelle Definition falsch ausgeführt wird und eine Nachricht eine Standard-Cluster-Übertragungswarteschlange abfließt, wird sie in der Erstellung einer permanentdynamischen Clusterübertragungswarteschlange angezeigt.

2. Der zweite Konfigurationsschritt befindet sich in " Isolieren Sie alle Nachrichtenflüsse, die keine Cluster-Übertragungswarteschlange mit anderen Flows gemeinsam nutzen dürfen ".

   In diesem Fall muss die Vertriebsanwendung, die Nachrichten aus der Warteschlange SALES unter SALESRV empfängt, isoliert werden. Es ist nur die Isolation von Nachrichten vom Gateway-WS-Manager erforderlich. Die drei Unterschritte sind:

   a) " Konfigurieren Sie die Ziele der Flows so, dass jede Zielwarteschlange die einzige Warteschlange in einem bestimmten Cluster ist, auf diesem Warteschlangenmanager ".

      In diesem Beispiel muss der Warteschlangenmanager SALESRV einem neuen Cluster innerhalb der Vertriebsabteilung hinzugefügt werden. Wenn Sie nur wenige Warteschlangen haben, die isoliert werden müssen, können Sie entscheiden, einen bestimmten Cluster für die SALES-Warteschlange zu erstellen. Eine mögliche Namenskonvention für den Clusternamen ist es, solche `Q.` *QueueName*-Cluster beispielsweise mit `Q.SALES` zu benennen. Ein alternativer Ansatz, der praktischer ist, wenn Sie eine große Anzahl von Warteschlangen zu isolieren haben, besteht darin, Cluster von isolierten Warteschlangen zu erstellen, in denen und wann dies erforderlich ist. Die Clusternamen können `QUEUES.` *n* lauten.

      In diesem Beispiel heißt der neue Cluster `Q.SALES`. Informationen zum Hinzufügen des neuen Clusters finden Sie in den Definitionen unter Änderungen zum Isolieren der Vertriebswarteschlange in einem neuen Cluster und Trennen der Gateway-Clusterübertragungswarteschlangen. Die Zusammenfassung der Definitionsänderungen lautet wie folgt:

      i) Fügen Sie `Q.SALES` zur Namensliste der Cluster auf den Repository-Warteschlangenmanagern hinzu. Auf die Namensliste wird im Parameter **REPOSNL** des Warteschlangenmanagers verwiesen.

      ii) Fügen Sie `Q.SALES` zur Namensliste der Cluster auf dem Gateway-Warteschlangenmanager hinzu. Die Namensliste wird in allen Aliasnamendefinitionen der Clusterwarteschlange und des Cluster-WS-Managers auf dem Gateway-WS-Manager bezeichnet.

      iii) Erstellen Sie eine Namensliste auf dem Warteschlangenmanager SALESRV für beide Cluster, zu denen er gehört, und ändern Sie die Clusterzugehörigkeit der SALES-Warteschlange:

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

      Die SALES-Warteschlange ist ein Mitglied beider Cluster, aber nur für den Übergang. Sobald die neue Konfiguration ausgeführt wird, entfernen Sie die SALES-Warteschlange aus dem SALES-Cluster (siehe Abbildung 15 auf Seite 65).

   b) " Erstellen Sie die Cluster-Sender- und Clusterempfängerkanäle für alle neuen Cluster, die Sie nach einer systematischen Namenskonvention erstellt haben ".

      i) Fügen Sie den Clusterempfängerkanal `Q.SALES.` *RepositoryQMgr* zu jedem der Repository-Warteschlangenmanager hinzu

      ii) Fügen Sie den Clustersenderkanal `Q.SALES.` *OtherRepositoryQMgr* zu jedem der Repository-Warteschlangenmanager hinzu, um eine Verbindung zum anderen Repository-Manager herzustellen. Starten Sie diese Kanäle.

      iii) Fügen Sie die Clusterempfängerkanäle `Q.SALES.SALESRV` und `Q.SALES.GATE` einem der aktiven Repository-Warteschlangenmanager hinzu.

      iv) Fügen Sie die Clustersenderkanäle `Q.SALES.SALESRV` und `Q.SALES.GATE` den Warteschlangenmanagern SALESRV und GATE hinzu. Verbinden Sie den Clustersenderkanal mit dem Repository-WS-Manager, auf dem Sie die Clusterempfängerkanäle erstellt haben.

   c) " Definieren Sie eine Clusterübertragungswarteschlange für jedes isolierte Ziel auf jedem Warteschlangenmanager, der Nachrichten an die Zielwarteschlange sendet ".

Definieren Sie auf dem Gateway-Warteschlangenmanager die Clusterübertragungswarteschlange XMITQ.Q.SALES.SALESRV für den Q.SALES.SALESRV-Clustersenderkanal:

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

3. Der dritte Konfigurationsschritt ist in " Erstellen Sie Clusterübertragungswarteschlangen, um die Governance- oder Monitoring-Voraussetzungen zu erfüllen " enthalten.

   Definieren Sie auf dem Gateway-WS-Manager die Clusterübertragungswarteschlangen:

```
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
```

Nächste Schritte

Wechseln Sie in die neue Konfiguration auf dem Gateway-Warteschlangenmanager.

Der Switch wird ausgelöst, indem die neuen Kanäle gestartet werden, und die Kanäle, die jetzt verschiedenen Übertragungswarteschlangen zugeordnet sind, erneut gestartet werden. Alternativ können Sie den Gateway-WS-Manager stoppen und starten.

1. Stoppen Sie die folgenden Kanäle auf dem Gateway-WS-Manager:

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

2. Starten Sie die folgenden Kanäle auf dem Gateway-WS-Manager:

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

Wenn der Wechsel abgeschlossen ist, entfernen Sie die SALES-Warteschlange aus dem SALES-Cluster (siehe Abbildung 15 auf Seite 65).

Zugehörige Konzepte

Art der zu verwendenden Clusterübertragungswarteschlange auswählen
Vorgehensweise zur Auswahl zwischen verschiedenen Konfigurationsoptionen für die Clusterübertragungswarteschlange.

Zugehörige Tasks

Clustering: Clusterübertragungswarteschlangen wechseln
Planen Sie, wie die Änderungen an den Clusterübertragungswarteschlangen eines vorhandenen Produktionswarteschlangenmanagers in Kraft gebracht werden.

Beispielcluster erstellen
Die Definitionen und Anweisungen zum Erstellen des Beispielclusters und zum Ändern des Clusters, um die SALES-Warteschlange und separate Nachrichten auf dem Gateway-Warteschlangenmanager zu isolieren.

Informationen zu diesem Vorgang

Die vollständigen **MQSC** -Befehle zum Erstellen der FINANCE-, SALES-und Q.SALES -Cluster werden in Definitionen für die Basiscluster, Änderungen zum Isolieren der Vertriebswarteschlange in einem neuen Cluster und zum Trennen der Gateway-Clusterübertragungswarteschlangenund Entfernen der Vertriebswarteschlange auf dem Warteschlangenmanager SALESRV aus dem Vertriebsclusterbereitgestellt. Der DEVELOP -Cluster wird in den Definitionen nicht angegeben, um die Definitionen kürzer zu halten.

Vorgehensweise

1. Erstellen Sie die Cluster SALES und FINANCE und den Gateway-Warteschlangenmanager.
   a) Erstellen Sie die Warteschlangenmanager.

      Führen Sie den Befehl `crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE` *QmgrName* für jeden Warteschlangenmanagernamen in Tabelle 4 auf Seite 63 aus.

Tabelle 4. Namen und Portnummern des Warteschlangenmanagers		
Beschreibung	**Name des Warteschlangenmanagers**	**Portnummer**
Finanzrepository	FINR1	1414
Finanzrepository	FINR2	1415
Finanzclient	FINCLT	1418
Verkaufsrepository	SALER1	1416
Verkaufsrepository	SALER2	1417
Verkaufsserver	SALESRV	1419
Gateway	GATE	1420

   b) Alle WS-Manager starten

      Führen Sie den Befehl `strmqm` *QmgrName* für jeden Warteschlangenmanagernamen in Tabelle 4 auf Seite 63 aus.

   c) Erstellen Sie die Definitionen für jeden der Warteschlangenmanager.

      Führen Sie den Befehl `runmqsc` *QmgrName* < *filename* aus, wobei die Dateien in Definitionen für die Basisclusteraufgelistet sind und der Dateiname mit dem Namen des Warteschlangenmanagers übereinstimmt.

      Definitionen für die Basiscluster
      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(FI
NANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1414)') CLUSTER(FI
NANCE) 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(FI
NANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINR2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1415)') CLUSTER(FI
NANCE) 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(FI
NANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINCLT) CHLTYPE(CLUSRCVR) CONNAME('localhost(1418)') CLUS
TER(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(SA□
LES) REPLACE
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1416)') CLUSTER(SA□
LES) 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(SA□
LES) REPLACE
DEFINE CHANNEL(SALES.SALER2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1417)') CLUSTER(SA□
LES) 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(SA□
LES) REPLACE
DEFINE CHANNEL(SALES.SALESRV) CHLTYPE(CLUSRCVR) CONNAME('localhost(1419)') CLUSTER(SA□
LES) 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(FI□
NANCE) REPLACE
DEFINE CHANNEL(FINANCE.GATE) CHLTYPE(CLUSRCVR) CONNAME('LOCALHOST(1420)') CLUSTER(FI□
NANCE) REPLACE
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('LOCALHOST(1416)') CLUSTER(SA□
LES) REPLACE
DEFINE CHANNEL(SALES.GATE)  CHLTYPE(CLUSRCVR) CONNAME('LOCALHOST(1420)') CLUSTER(SA□
LES) REPLACE
DEFINE QALIAS(A.SALES) CLUSNL(ALL) TARGET(SALES) TARGTYPE(QUEUE) DEFBIND(NOTFIXED) RE□
PLACE
DEFINE QREMOTE(FINCLT) RNAME(' ') RQMNAME(FINCLT) CLUSNL(ALL) REPLACE
DEFINE QREMOTE(SALESRV) RNAME(' ') RQMNAME(SALESRV) CLUSNL(ALL) REPLACE
```

2. Testen Sie die Konfiguration, indem Sie das Beispielanforderungsprogramm ausführen.
   a) Starten Sie das Auslösemonitorprogramm auf dem SALESRV-Warteschlangenmanager

      Öffnen Sie unter Windows ein Befehlsfenster und führen Sie den Befehl `runmqtrm -m SALESRV` aus

   b) Führen Sie das Beispielanforderungsprogramm aus, und senden Sie eine Anforderung.

      Öffnen Sie unter Windows ein Befehlsfenster und führen Sie den Befehl `amqsreq A.SALES FINCLT` aus

      Die Anforderungsnachricht wird zurückgemeldet, und nach 15 Sekunden wird das Beispielprogramm beendet.
3. Erstellen Sie die Definitionen, um die SALES-Warteschlange im `Q.SALES`-Cluster und separate Clusternachrichten für den SALES- und `FINANCE`-Cluster im Gateway-Warteschlangenmanager zu isolieren.

   Führen Sie den Befehl `runmqsc` *QmgrName* < *filename* aus, wobei die Dateien in der folgenden Liste aufgeführt sind und der Dateiname fast mit dem Namen des Warteschlangenmanagers übereinstimmt.

Änderungen zum Isolieren der Verkaufswarteschlange in einem neuen Cluster und Trennen der Gateway-Cluster-Übertragungswarteschlangen

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.SA
LES) REPLACE
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1416)') CLUSTER(Q.SA
LES) 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.SA
LES) REPLACE
DEFINE CHANNEL(Q.SALES.SALER2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1417)') CLUSTER(Q.SA
LES) REPLACE
```

chgsalesrv.txt

```
DEFINE NAMELIST (CLUSTERS) NAMES(SALES, Q.SALES)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)') CLUSTER(Q.SA
LES) REPLACE
DEFINE CHANNEL(Q.SALES.SAVESRV) CHLTYPE(CLUSRCVR) CONNAME('localhost(1419)') CLUS
TER(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.SA
LES) REPLACE
DEFINE CHANNEL(Q.SALES.GATE)  CHLTYPE(CLUSRCVR) CONNAME('localhost(1420)') CLUSTER(Q.SA
LES) 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. Entfernen Sie die Warteschlange SALES aus dem SALES-Cluster.

   Führen Sie den Befehl **MQSC** in Abbildung 15 auf Seite 65 aus:

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

   Abbildung 15. Entfernen Sie die Verkaufswarteschlange auf dem WS-Manager SALESRV aus dem Vertriebscluster.

5. Schalten Sie die Kanäle in die neuen Übertragungswarteschlangen ein.

   Voraussetzung ist, dass alle Kanäle, die vom GATE-Warteschlangenmanager verwendet werden, gestoppt und gestartet werden. Stoppen und starten Sie den WS-Manager mit der geringsten Anzahl an Befehlen.

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

Nächste Schritte

1. Das Beispielanforderungsprogramm erneut ausführen, um die neue Konfiguration zu überprüfen. Siehe Schritt „2” auf Seite 64

2. Überwachen Sie die Nachrichten, die durch alle Clusterübertragungswarteschlangen auf dem GATE-Warteschlangenmanager fließen:
   a. Ändern Sie die Definition der einzelnen Clusterübertragungswarteschlangen, um die Warteschlangenüberwachung zu aktivieren.

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

   b. Überprüfen Sie, ob die Statistiküberwachung des Warteschlangenmanagers OFF ist, um die Ausgabe zu minimieren. Setzen Sie das Überwachungsintervall auf einen niedrigeren Wert, um mehrere Tests bequem auszuführen.

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

   c. Starten Sie den Warteschlangenmanager GATE erneut.
   d. Führen Sie das Beispielanforderungsprogramm einige Male aus, um sicherzustellen, dass eine gleiche Anzahl von Nachrichten durch SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV und SYSTEM.CLUSTER.TRANSMIT.QUEUE fließt. Anforderungen durchlaufen SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV und antworten über SYSTEM.CLUSTER.TRANSMIT.QUEUE.

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

   e. Die Ergebnisse in einigen Intervallen lauten wie folgt:

```
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.
```

Eine Anforderungs-und Antwortnachricht wurde im ersten Intervall und zwei in der zweiten Nachricht gesendet. Sie können daraus schließen, dass die Anforderungsnachrichten auf SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV und die Antwortnachrichten auf SYSTEM.CLUSTER.TRANSMIT.QUEUE platziert wurden.

Clustering: Clusterübertragungswarteschlangen wechseln
Planen Sie, wie die Änderungen an den Clusterübertragungswarteschlangen eines vorhandenen Produktionswarteschlangenmanagers in Kraft gebracht werden.

Vorbereitende Schritte

Wenn Sie die Anzahl der Nachrichten reduzieren, die der Switching-Prozess in die neue Übertragungswarteschlange übertragen muss, wird der Wechsel schneller abgeschlossen. Lesen Sie den Abschnitt Wie der Prozess zum Umschalten des Clustersenderkanals in eine andere Übertragungswarteschlange funktioniert aus den Gründen, die versucht haben, die Übertragungswarteschlange zu leeren, bevor Sie fortfahren.

Informationen zu diesem Vorgang

Sie haben die Wahl zwischen zwei Möglichkeiten, die Änderungen an Clusterübertragungswarteschlangen wirksam zu machen.

1. Lassen Sie den WS-Manager die Änderungen automatisch vornehmen. Dies ist die Standardeinstellung. Der Warteschlangenmanager wechselt die Clustersenderkanäle mit anstehender Übertragungswarteschlange, wenn ein Clustersenderkanal als Nächstes gestartet wird.
2. Nehmen Sie die Änderungen manuell vor. Sie können die Änderungen an einem Clustersenderkanal vornehmen, wenn er gestoppt wird. Sie können sie von einer Clusterübertragungswarteschlange in eine andere übertragen, bevor der Clustersenderkanal gestartet wird.

Welche Faktoren berücksichtigen Sie bei der Entscheidung, welche der beiden Optionen Sie auswählen können, und wie verwalten Sie den Switch?

Prozedur

- Option 1: Lassen Sie den Warteschlangenmanager die Änderungen automatisch vornehmen (siehe „Aktive Clustersenderkanäle zu einer anderen Gruppe von Clusterübertragungswarteschlangen wechseln" auf Seite 69).

  Wählen Sie diese Option aus, wenn der WS-Manager den Switch für Sie herstellen soll.

  Eine alternative Möglichkeit, diese Option zu beschreiben, besteht darin, dass der Warteschlangenmanager einen Clustersenderkanal umschaltet, ohne dass der Kanal gestoppt werden muss. Sie haben die Möglichkeit, den Kanal zu zwingen, den Kanal zu stoppen und dann den Kanal zu starten, damit der Schalter früher passiert. Die Umschaltung wird gestartet, wenn der Kanal gestartet wird und er wird ausgeführt, während der Kanal aktiv ist, was sich von Option 2 unterscheidet. In Option 2 erfolgt die Umschaltung, wenn der Kanal gestoppt wird.

  Wenn Sie diese Option auswählen, indem Sie den Switch automatisch passieren lassen, wird der Umschaltvorgang gestartet, wenn ein Clustersenderkanal gestartet wird. Wenn der Kanal nicht gestoppt wird, wird er gestartet, wenn er inaktiv wird, wenn eine Nachricht zum Verarbeiten vorhanden ist. Wenn der Kanal gestoppt ist, starten Sie ihn mit dem Befehl `START CHANNEL`.
  Der Switchprozess wird beendet, sobald keine Nachrichten mehr für den Clustersenderkanal in der Übertragungswarteschlange übrig sind, die der Kanal bedient hat. Sobald dies der Fall ist, werden neu eingetrohtes Nachrichten für den Clustersenderkanal direkt in der neuen Übertragungswarteschlange gespeichert. Bis dahin werden Nachrichten in der alten Übertragungswarteschlange gespeichert, und der Switching-Prozess überträgt Nachrichten aus der alten Übertragungswarteschlange in die neue Übertragungswarteschlange. Der Clustersenderkanal leitet Nachrichten aus der neuen Clusterübertragungswarteschlange während des gesamten Switching-Prozesses weiter.
  Wenn der Switchprozess abgeschlossen ist, hängt vom Status des Systems ab. Wenn Sie die Änderungen in einem Wartungsfenster vornehmen, müssen Sie vorher prüfen, ob der Switching-Prozess in der Zeit abgeschlossen ist. Ob die Zeit vollständig abgeschlossen wird, hängt davon ab, ob die Anzahl der Nachrichten, die auf die Übertragung aus der alten Übertragungswarteschlange warten, null erreicht.

  Der Vorteil der ersten Methode ist, dass sie automatisch ist. Ein Nachteil besteht darin, dass Sie sicher sein müssen, dass Sie das System steuern können, um den Switchprozess im Wartungsfenster zu beenden, wenn die Konfigurationsänderungen auf ein Verwaltungsfenster beschränkt sind. Wenn Sie sich nicht sicher sein können, ist Option 2 möglicherweise eine bessere Wahl.
- Option 2: Nehmen Sie die Änderungen manuell vor („Gestoppten Clustersenderkanal in eine andere Clusterübertragungswarteschlange wechseln" auf Seite 70).

  Wählen Sie diese Option aus, wenn Sie den gesamten Switching-Prozess manuell steuern möchten oder ob Sie einen gestoppten oder inaktiven Kanal umschalten wollen. Es ist eine gute Wahl, wenn Sie einige Clustersenderkanäle umschalten und während eines Wartungsfensters den Switch ausführen möchten.

  Eine alternative Beschreibung dieser Option ist die Angabe, dass Sie den Clustersenderkanal umschalten, während der Clustersenderkanal gestoppt ist.

Wenn Sie diese Option auswählen, haben Sie die vollständige Kontrolle über den Zeitpunkt, an dem der Switch ausgeführt wird.

Sie können sicher sein, dass Sie den Switching-Prozess in einem festen Zeitraum in einem Wartungsfenster abschließen können. Der Zeitpunkt, zu dem der Switch ausgeführt wird, hängt davon ab, wie viele Nachrichten von einer Übertragungswarteschlange an die andere übertragen werden müssen. Wenn Nachrichten weiterhin ankommen, kann es zu einer Zeit dauern, bis der Prozess alle Nachrichten übertragen hat.

Sie haben die Möglichkeit, den Kanal zu wechseln, ohne Nachrichten aus der alten Übertragungswarteschlange zu übertragen. Der Switch ist " instant ".

Wenn Sie den Clustersenderkanal erneut starten, beginnt er mit der Verarbeitung von Nachrichten in der Übertragungswarteschlange, die Sie neu zugeordnet haben.

Der Vorteil der zweiten Methode besteht darin, dass Sie die Kontrolle über den Schaltvorgang haben. Der Nachteil besteht darin, dass Sie die zu vermittelnden Clustersenderkanäle identifizieren müssen, die erforderlichen Befehle ausführen und alle unbestäuben Kanäle auflösen müssen, die möglicherweise verhindern, dass der Clustersenderkanal gestoppt wird.

Zugehörige Konzepte

Art der zu verwendenden Clusterübertragungswarteschlange auswählen
Vorgehensweise zur Auswahl zwischen verschiedenen Konfigurationsoptionen für die Clusterübertragungswarteschlange.

Funktionsweise des Prozesses zum Wechseln des Clustersenderkanals in eine andere Übertragungswarteschlange

Zugehörige Tasks

Clustering: Beispielkonfiguration mehrerer Clusterübertragungswarteschlangen
In dieser Task wenden Sie die Schritte zum Planen mehrerer Clusterübertragungswarteschlangen auf drei sich überlappende Cluster an. Die Anforderungen bestehen darin, Nachrichten von allen anderen Nachrichtenflüssen in eine Clusterwarteschlange zu trennen und Nachrichten für verschiedene Cluster in verschiedenen Clusterübertragungswarteschlangen zu speichern.

Aktive Clustersenderkanäle zu einer anderen Gruppe von Clusterübertragungswarteschlangen wechseln
Mit dieser Task erhalten Sie drei Optionen zum Wechseln der aktiven Clustersenderkanäle. Eine Möglichkeit besteht darin, dass der WS-Manager den Switch automatisch macht, was die Ausführung von Anwendungen nicht beeinträchtigt. Die anderen Optionen sind zum manuellen Stoppen und Starten von Kanälen oder zum erneuten Starten des Warteschlangenmanagers.

Vorbereitende Schritte

Ändern Sie die Konfiguration der Clusterübertragungswarteschlange. Sie können das WS-Manager-Attribut **DEFCLXQ** ändern oder das Attribut **CLCHNAME** von Übertragungswarteschlangen hinzufügen oder ändern.

Wenn Sie die Anzahl der Nachrichten reduzieren, die der Switching-Prozess in die neue Übertragungswarteschlange übertragen muss, wird der Wechsel schneller abgeschlossen. Lesen Sie den Abschnitt Wie der Prozess zum Umschalten des Clustersenderkanals in eine andere Übertragungswarteschlange funktioniert aus den Gründen, die versucht haben, die Übertragungswarteschlange zu leeren, bevor Sie fortfahren.

Informationen zu diesem Vorgang

Verwenden Sie die Schritte in der Task als Basis für die Bearbeitung eines eigenen Plans für die Änderung der Konfiguration der Clusterübertragungswarteschlange.

Vorgehensweise

1. Optional: Aktualisieren Sie den aktuellen Kanalstatus

Erstellen Sie einen Datensatz mit dem Status der aktuellen und gespeicherten Kanäle, die Clusterübertragungswarteschlangen bedienen. Mit den folgenden Befehlen wird der Status angezeigt, der den Systemclusterübertragungswarteschlangen zugeordnet ist. Fügen Sie eigene Befehle hinzu, um den Status anzuzeigen, der den von Ihnen definierten Clusterübertragungswarteschlangen zugeordnet ist. Verwenden Sie eine Konvention wie XMITQ. *ChannelName*, um Clusterübertragungswarteschlangen zu benennen, die Sie definieren, damit der Kanalstatus für diese Übertragungswarteschlangen einfach angezeigt werden kann.

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

2. Übertragungswarteschlangen wechseln.
   - Tun Sie nichts. Der Warteschlangenmanager wechselt beim Neustart die Clustersenderkanäle, wenn er nach dem Stoppen oder Inaktiv erneut gestartet wird.

     Wählen Sie diese Option aus, wenn Sie keine Regeln oder Bedenken zum Ändern einer Warteschlangenmanagerkonfiguration haben. Aktive Anwendungen sind von den Änderungen nicht betroffen.
   - Starten Sie den Warteschlangenmanager erneut. Alle Clustersenderkanäle werden bei Bedarf automatisch gestoppt und erneut gestartet.

     Wählen Sie diese Option aus, um alle Änderungen sofort einzuleiten. Aktive Anwendungen werden durch den Warteschlangenmanager unterbrochen, da er heruntergefahren und erneut gestartet wird.
   - Stoppen Sie einzelne Clustersenderkanäle, und starten Sie sie erneut.

     Wählen Sie diese Option aus, um ein paar Kanäle sofort zu ändern. Beim Ausführen von Anwendungen wird eine kurze Verzögerung bei der Nachrichtenübertragung zwischen dem Stoppen und dem erneuten Starten des Nachrichtenkanals angezeigt. Der Clustersenderkanal bleibt aktiv, außer während der Zeit, in der Sie ihn gestoppt haben. Während der Vermittlung werden Nachrichten an die alte Übertragungswarteschlange zugestellt, durch den Vermittlungsvorgang in die neue Übertragungswarteschlange übertragen und vom Clustersenderkanal aus der neuen Übertragungswarteschlange weitergeleitet.
3. Optional: Kanäle überwachen, während sie umschalten

   Zeigen Sie den Kanalstatus und die Übertragungswarteschlangentiefe während des Switchs an. Im folgenden Beispiel wird der Status der Übertragungswarteschlangen des Systemclusters angezeigt.

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

4. Optional: Überwachen Sie die Nachrichten AMQ7341 `Die Übertragungswarteschlange für den Kanal` *`ChannelName`*`, die von der Warteschlange` *`QueueName`* `in` *`QueueName`* umgeschaltet wurde, die in das Fehlerprotokoll des Warteschlangenmanagers geschrieben werden.

Gestoppten Clustersenderkanal in eine andere Clusterübertragungswarteschlange wechseln
Wenn Sie die Änderungen manuell vornehmen möchten, nehmen Sie die Änderungen an einem Clustersenderkanal vor, wenn er gestoppt ist, und übertragen ihn von einer Clusterübertragungswarteschlange in eine andere, bevor der Clustersenderkanal gestartet wird.

Vorbereitende Schritte

Sie können einige Konfigurationsänderungen vornehmen und sie jetzt wirksam werden lassen, ohne die betroffenen Clustersenderkanäle zu starten. Alternativ können Sie die Konfigurationsänderungen vornehmen, die Sie als einen der Schritte in der Task benötigen.

Wenn Sie die Anzahl der Nachrichten reduzieren, die der Switching-Prozess in die neue Übertragungswarteschlange übertragen muss, wird der Wechsel schneller abgeschlossen. Lesen Sie den Abschnitt Wie der Prozess zum Umschalten des Clustersenderkanals in eine andere Übertragungswarteschlange

funktioniert aus den Gründen, die versucht haben, die Übertragungswarteschlange zu leeren, bevor Sie fortfahren.

Informationen zu diesem Vorgang

Diese Task schaltet die Übertragungswarteschlangen, die von gestoppten oder inaktiven Clustersenderkanälen bereitgestellt werden. Sie können diese Task ausführen, da ein Clustersenderkanal gestoppt ist und Sie die Übertragungswarteschlange sofort umschalten möchten. Dies ist beispielsweise der Fall, wenn ein Clustersenderkanal nicht gestartet wird oder ein anderes Konfigurationsproblem vorliegt. Um das Problem zu beheben, müssen Sie einen Clustersenderkanal erstellen und die Übertragungswarteschlange für den alten Clustersenderkanal mit dem neuen Clustersenderkanal verknüpfen, den Sie definiert haben.

Ein wahrscheinlicher Fall ist, dass Sie die Steuerung steuern wollen, wenn die Neukonfiguration von Clusterübertragungswarteschlangen ausgeführt wird. Um die Rekonfiguration vollständig zu steuern, stoppen Sie die Kanäle, ändern die Konfiguration und wechseln dann die Übertragungswarteschlangen.

Vorgehensweise

1. Stoppen Sie die Kanäle, die Sie wechseln möchten.
   a) Stoppen Sie alle aktiven oder inaktiven Kanäle, die Sie wechseln möchten. Wenn Sie einen inaktiven Clustersenderkanal stoppen, wird dieser verhindert, während Sie Konfigurationsänderungen vornehmen.

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

2. Optional: Nehmen Sie die Konfigurationsänderungen vor.

   Informationen zum Beispiel finden Sie unter „Clustering: Beispielkonfiguration mehrerer Clusterübertragungswarteschlangen” auf Seite 59.
3. Schalten Sie die Clustersenderkanäle in die neuen Clusterübertragungswarteschlangen um.

   Multi Geben Sie unter Multiplatforms folgenden Befehl aus:

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

   z/OS Verwenden Sie unter z/OS die Funktion SWITCH des Befehls CSQUTIL, um die Nachrichten zu Wechseln oder die Vorgänge zu überwachen. Verwenden Sie den folgenden Befehl.

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

   Weitere Informationen finden Sie im Abschnitt SWITCH-Funktion .

   Der Befehl **runswchl**oder CSQUTIL SWITCH überträgt alle Nachrichten in der alten Übertragungswarteschlange an die neue Übertragungswarteschlange. Wenn die Anzahl der Nachrichten in der alten Übertragungswarteschlange für diesen Kanal den Wert null erreicht, wird der Switch abgeschlossen. Der Befehl ist synchron. Der Befehl schreibt während des Umschaltvorgangs Statusnachrichten in das Fenster.

   Während der Übertragungsphase werden vorhandene und neue Nachrichten, die für den Clustersenderkanal bestimmt sind, in die neue Übertragungswarteschlange übertragen.

   Da der Clustersenderkanal gestoppt ist, werden die Nachrichten in der neuen Übertragungswarteschlange erstellt. Vergleichen Sie den gestoppten Clustersenderkanal mit dem Schritt „2” auf Seite 70 in „Aktive Clustersenderkanäle zu einer anderen Gruppe von Clusterübertragungswarteschlangen wechseln” auf Seite 69. In diesem Schritt wird der Clustersenderkanal ausgeführt, sodass Nachrichten nicht notwendigerweise in der neuen Übertragungswarteschlange erstellt werden müssen.
4. Optional: Kanäle überwachen, während sie umschalten

Zeigen Sie in einem anderen Befehlsfenster die Länge der Übertragungswarteschlange während des Switchs an. Im folgenden Beispiel wird der Status der Übertragungswarteschlangen des Systemclusters angezeigt.

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

5. Optional: Überwachen Sie die Nachrichten AMQ7341 `Die Übertragungswarteschlange für den Kanal` *`ChannelName`*`, die von der Warteschlange` *`QueueName`* `in` *`QueueName`* umgeschaltet wurde, die in das Fehlerprotokoll des Warteschlangenmanagers geschrieben werden.
6. Starten Sie die Clustersenderkanäle erneut, die Sie gestoppt haben.

   Die Kanäle werden nicht automatisch gestartet, wenn Sie sie gestoppt haben, indem Sie sie in den Status STOPPED stellen.

   ```
   START CHANNEL(ChannelName)
   ```

Zugehörige Verweise
runswchl
GELÖST-CHANNEL
STOP CHANNEL

Clustering: Best Practices für Migration und Änderung
Dieser Abschnitt enthält eine Anleitung zur Planung und Verwaltung von IBM MQ-Clustern. Diese Informationen sind ein Leitfaden, der auf Tests und Feedback von Kunden basiert.

1. „Objekte in einem Cluster verschieben" auf Seite 72 (Bewährte Verfahren für das Verschieben von Objekten innerhalb eines Clusters ohne Installation von Fixpacks oder neuen Versionen von IBM MQ).
2. „Upgrades und Wartungsinstallationen" auf Seite 74 (Bewährte Verfahren für die Aufrechterhaltung einer betriebsweisen Clusterarchitektur und die Ausführung von Wartungs-oder Upgrades und Tests der neuen Architektur).

Objekte in einem Cluster verschieben

Anwendungen und ihre Warteschlangen

Wenn Sie eine Warteschlangeninstanz, die auf einem WS-Manager gehostet wird, in einem anderen Warteschlangenmanager verschieben müssen, können Sie mit den Parametern für die Lastverteilung arbeiten, um einen reibungslosen Übergang zu gewährleisten.

Erstellen Sie eine Instanz der Warteschlange, in der sie neu gehostet werden soll, verwenden Sie jedoch die Einstellungen für die Lastverteilung im Cluster, um das Senden von Nachrichten an die ursprüngliche Instanz fortzusetzen, bis Ihre Anwendung bereit ist, zu wechseln. Dies wird mit den folgenden Schritten erreicht:

1. Setzen Sie die **CLWLRANK**-Eigenschaft der vorhandenen Warteschlange auf einen hohen Wert, z. B. auf 5.
2. Erstellen Sie die neue Instanz der Warteschlange und setzen Sie die **CLWLRANK**-Eigenschaft auf 0.
3. Führen Sie eine beliebige weitere Konfiguration des neuen Systems aus, z. B. die Implementierung und das Starten von Anwendungen für die neue Instanz der Warteschlange.
4. Setzen Sie die **CLWLRANK**-Eigenschaft der neuen Warteschlangeninstanz auf einen höheren Wert als die ursprüngliche Instanz, z. B. auf 9.
5. Zulassen, dass die ursprüngliche Warteschlangeninstanz alle in der Warteschlange befindlichen Nachrichten im System verarbeitet und dann die Warteschlange löscht.

Verschieben ganzer WS-Manager

Wenn sich der WS-Manager auf demselben Host befindet, die IP-Adresse jedoch geändert wird, lautet der Prozess wie folgt:

- DNS, wenn es korrekt verwendet wird, kann die Vereinfachung des Prozesses vereinfachen. Informationen zur Verwendung von DNS durch Festlegen des Kanalattributs Verbindungsname (CONNAME) finden Sie unter ALTER CHANNEL.
- Wenn Sie ein vollständiges Repository verschieben, müssen Sie sicherstellen, dass Sie mindestens ein anderes vollständiges Repository haben, das problemlos ausgeführt wird (z. B. keine Probleme mit dem Kanalstatus), bevor Sie Änderungen vornehmen.
- Setzen Sie den Warteschlangenmanager mit dem Befehl SUSPEND QMGR aus, um die Datenverkehrsaufbauung zu vermeiden.
- Ändern Sie die IP-Adresse des Computers. Wenn Ihre CLUSRCVR-Kanaldefinition im Feld CONNAME eine IP-Adresse verwendet, ändern Sie diesen IP-Adresseneintrag. Der DNS-Cache muss möglicherweise durchgebürstet werden, um sicherzustellen, dass Aktualisierungen überall verfügbar sind.
- Wenn der Warteschlangenmanager die Verbindung zu den vollständigen Repositorys wiederherstellt, lösen sich automatisch die Kanalautodefinitionen selbst auf.
- Wenn der Warteschlangenmanager ein vollständiges Repository und die Änderungen an der IP-Adresse befindet, muss sichergestellt werden, dass die Teilpartien so bald wie möglich umgestellt werden, um manuell definierte CLUSSDR-Kanäle an die neue Position zu verweisen. Solange dieser Schalter nicht ausgeführt wird, können diese WS-Manager möglicherweise nur die verbleibenden (unveränderten) vollständigen Repositorys in Verbindung setzen, und es werden Warnungen angezeigt, die sich auf die falsche Kanaldefinition richten.
- Den WS-Manager mit dem Befehl RESUME QMGR wiederaufnehmen.

Wenn der Warteschlangenmanager auf einen neuen Host verschoben werden muss, ist es möglich, die Daten des Warteschlangenmanagers zu kopieren und aus einer Sicherung zurückzuschreiben. Dieser Prozess wird jedoch nicht empfohlen, es sei denn, es gibt keine anderen Optionen; es kann jedoch besser sein, einen Warteschlangenmanager auf einer neuen Maschine zu erstellen und Warteschlangen und Anwendungen zu replizieren, wie im vorherigen Abschnitt beschrieben. Diese Situation bietet einen reibungslosen Rollover/Rollback-Mechanismus.

Wenn Sie entschlossen sind, einen vollständigen Warteschlangenmanager unter Verwendung der Sicherung zu verschieben, befolgen Sie die folgenden bewährten Verfahren:

- Behandeln Sie den gesamten Prozess als Restore des Warteschlangenmanagers von der Sicherung, wobei Sie alle Prozesse anwenden, die Sie in der Regel für die Systemwiederherstellung verwenden, die für Ihre Betriebssystemumgebung geeignet sind.
- Verwenden Sie den Befehl **REFRESH CLUSTER** nach der Migration, um alle lokal gespeicherten Clusterinformationen (einschließlich aller unbestätigten, automatisch definierten Kanäle) zu löschen und die erneute Erstellung zu erzwingen.

  Anmerkung: Bei großen Clustern kann der Befehl **REFRESH CLUSTER** während seiner Ausführung und danach in 27-Tage-Intervallen, wenn die Clusterobjekte ihre Statusaktualisierungen automatisch an alle interessierten Warteschlangenmanager senden, zu Unterbrechungen führen. Nähere Informationen hierzu erhalten Sie im Abschnitt Die Aktualisierung in einem großen Cluster kann sich auf die Leistung und Verfügbarkeit auswirken.

Wenn Sie einen Warteschlangenmanager erstellen und die Konfiguration von einem vorhandenen WS-Manager im Cluster replizieren (wie in diesem Abschnitt beschrieben), behandeln Sie die beiden verschiedenen Warteschlangenmanager niemals so, wie sie eigentlich identisch sind. Geben Sie insbesondere keinen neuen Warteschlangenmanager denselben Warteschlangenmanagernamen und die gleiche IP-Adresse an. Der Versuch, auf diese Weise schnell zu einem Ersatz-Warteschlangenmanager zu wechseln, ist eine häufige Ursache für Probleme in IBM MQ-Clustern. Der Cache erwartet Aktualisierungen, einschließlich des Attributs **QMID**, und der Status kann beschädigt sein.

Wenn versehentlich zwei verschiedene Warteschlangenmanager mit demselben Namen erstellt werden, wird empfohlen, den Befehl RESET CLUSTER **QMID** zu verwenden, um den falschen Eintrag aus dem Cluster zu entfernen.

Upgrades und Wartungsinstallationen

Vermeiden Sie das so genannte Big-Bang-Szenario (z. B. das Stoppen aller Cluster-und WS-Manageraktivitäten, das Anwenden aller Upgrades und Wartungsarbeiten auf alle WS-Manager und das anschließende Starten aller Aktivitäten). Cluster sind so konzipiert, dass sie immer noch mit mehreren Versionen des Warteschlangenmanagers zusammenarbeiten, so dass ein gut geplanter, gestaffter Wartungsansatz empfohlen wird.

Haben Sie einen Backup-Plan:

- Haben Sie Sicherungen erstellt?
- Vermeiden Sie sofort die Verwendung der neuen Clusterfunktionalität: Warten Sie, bis Sie sicher sind, dass alle WS-Manager auf die neue Version aufgerüstet sind, und sind sicher, dass Sie keinen Rollback rückgängig machen werden. Die Verwendung einer neuen Clusterfunktion in einem Cluster, in dem einige WS-Manager noch auf einer früheren Version stehen, kann zu undefiniertem Verhalten führen.

  Ein Repository speichert die empfangenen Datensätze in der eigenen Version. Hat der empfangene Datensatz eine neuere Version, werden beim Speichern die Attribute, die zu der späteren Version gehören, gelöscht. Ein IBM MQ 9.3-Warteschlangenmanager, der Informationen zu einem IBM MQ 9.4-MQ-Warteschlangenmanager empfängt, speichert nur IBM MQ 9.3-Informationen. Ein IBM MQ 9.4-Repository, das einen IBM MQ 9.3-Datensatz empfängt, speichert Standardwerte für Attribute, die in der neueren Version eingeführt wurden. Diese Standardwerte werden für Attribute übernommen, die nicht in dem empfangenen Datensatz enthalten sind.

Migrieren Sie zuerst die vollständigen Repositorys. Obwohl sie Informationen weiterleiten können, die sie nicht verstehen, können sie sie nicht fortbestehen, so dass es nicht der empfohlene Ansatz ist, es sei denn, es ist absolut notwendig. Weitere Informationen finden Sie im Abschnitt Migration des WS-Managers-Clusters .

Clustering: Best Practices für REFRESH CLUSTER verwenden
Sie verwenden den Befehl **REFRESH CLUSTER** , um alle lokal gespeicherten Informationen zu einem Cluster zu löschen und diese Informationen aus den vollständigen Repositorys im Cluster erneut zu erstellen. Sie sollten diesen Befehl nicht verwenden, außer in außergewöhnlichen Umständen. Wenn Sie es verwenden müssen, gibt es besondere Hinweise darauf, wie Sie es verwenden. Diese Informationen sind ein Leitfaden, der auf Tests und Feedback von Kunden basiert.

Führen Sie nur den Befehl REFRESH CLUSTER aus, wenn Sie dies wirklich tun müssen.

Die IBM MQ-Clustertechnologie stellt sicher, dass jede Änderung an der Clusterkonfiguration, z. B. die Änderung an einer Clusterwarteschlange, automatisch allen Mitgliedern des Clusters bekannt gegeben wird, die diese Information benötigen. Es ist nicht notwendig, weitere administrative Schritte zu unternehmen, um diese Weitergabe von Informationen zu erreichen.

Wenn solche Informationen nicht zu den Warteschlangenmanagern im Cluster gelangen, in denen sie erforderlich sind, z. B. eine Clusterwarteschlange, die von einem anderen Warteschlangenmanager im Cluster nicht bekannt ist, wenn eine Anwendung versucht, sie zum ersten Mal zu öffnen, bedeutet dies ein Problem in der Clusterinfrastruktur. Es ist beispielsweise möglich, dass ein Kanal nicht zwischen einem WS-Manager und einem vollständigen WS-Manager-Repository gestartet werden kann. Daher müssen alle Situationen, in denen Inkonsistenzen beobachtet werden, untersucht werden. Lösen Sie die Situation nach Möglichkeit ohne Verwendung des Befehls **REFRESH CLUSTER** auf.

In seltenen Fällen, die an anderer Stelle in dieser Produktdokumentation oder auf Anforderung des IBM Support dokumentiert sind, können Sie den Befehl **REFRESH CLUSTER** verwenden, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen und diese Informationen aus den vollständigen Repositorys im Cluster erneut zu erstellen.

Die Neuerung in einem großen Cluster kann die Leistung und Verfügbarkeit des Clusters beeinträchtigen.

Die Verwendung des Befehls **REFRESH CLUSTER** kann während der Ausführung des Clusters zu Unterbrechungen führen, z. B. durch eine plötzliche Zunahme der Arbeit für die vollständigen Repositorys, wenn sie die erneute Weitergabe von Clusterressourcen des Warteschlangenmanagers verarbeiten. Wenn Sie in einem großen Cluster (d. h. viele Hunderte von Warteschlangenmanagern) aktualisieren, sollten Sie die Verwendung des Befehls in der täglichen Arbeit vermeiden, wenn möglich, und alternative Methoden verwenden, um spezifische Inkonsistenzen zu korrigieren. Wenn beispielsweise eine Clusterwarteschlange im Cluster nicht ordnungsgemäß weitergegeben wird, wird die Warteschlangenkonfiguration im gesamten Cluster von einem ersten Untersuchungsverfahren aktualisiert, das die Clusterwarteschlangendefinition aktualisiert, z. B. die Beschreibung der Clusterwarteschlange, die die Warteschlangenkonfiguration ändert. Dieser Prozess kann dazu beitragen, das Problem zu identifizieren und möglicherweise eine temporäre Inkonsistenz zu beheben.

Wenn alternative Methoden nicht verwendet werden können und Sie **REFRESH CLUSTER** in einem großen Cluster ausführen müssen, sollten Sie dies zu Zeiten geringer Systemauslastung oder während eines Wartungsfensters tun, um Auswirkungen auf Benutzerworkloads zu vermeiden. Sie sollten auch vermeiden, einen großen Cluster in einem einzigen Stapel zu aktualisieren und stattdessen die Aktivität wie in „Leistungs-und Verfügbarkeitsprobleme vermeiden, wenn Clusterobjekte automatische Aktualisierungen senden" auf Seite 75 erläutert zu stagnihen.

Leistungs-und Verfügbarkeitsprobleme vermeiden, wenn Clusterobjekte automatische Aktualisierungen senden

Nachdem ein neues Clusterobjekt in einem Warteschlangenmanager definiert ist, wird eine Aktualisierung für dieses Objekt alle 27 Tage ab der Definitionierungszeit generiert und an alle vollständigen Repositorys im Cluster und an alle anderen interessierten Warteschlangenmanager gesendet. Wenn Sie den Befehl **REFRESH CLUSTER** an einen Warteschlangenmanager ausgeben, setzen Sie die Systemzeit für diese automatische Aktualisierung für alle Objekte zurück, die lokal im angegebenen Cluster definiert sind.

Wenn Sie einen großen Cluster (d. a. viele Hunderte von Warteschlangenmanagern) in einem einzigen Stapel oder unter anderen Umständen aktualisieren, z. B. ein System aus der Konfigurationssicherung erneut erstellen, werden alle diese Warteschlangenmanager nach 27 Tagen alle ihre Objektdefinitionen erneut für die vollständigen Repositorys bekannt machen. Dies könnte wiederum dazu führen, dass das System erheblich langsamer oder gar nicht mehr verfügbar ist, bis alle Aktualisierungen abgeschlossen sind. Wenn Sie also mehrere Warteschlangenmanager in einem großen Cluster aktualisieren oder erneut erstellen müssen, sollten Sie die Aktivität über mehrere Stunden oder mehrere Tage stagniern, sodass nachfolgende automatische Aktualisierungen die Systemleistung nicht regelmäßig beeinträchtigen.

Die Systemclusterprotokollwarteschlange

Wenn eine **REFRESH CLUSTER** ausgeführt wird, erstellt der WS-Manager eine Momentaufnahme des Clusterstatus vor der Aktualisierung und speichert sie in `SYSTEM.CLUSTER.HISTORY.QUEUE` (SCHQ), wenn sie auf dem Warteschlangenmanager definiert ist. Diese Momentaufnahme dient ausschließlich zu Servicezwecken für IBM im Falle von späteren Problemen mit dem System.

Der SCHQ wird standardmäßig auf verteilten WS-Managern beim Start definiert. Bei der Migration von z/OS muss SCHQ manuell definiert werden.

Die Nachrichten in der SCHQ laufen nach drei Monaten ab.

Zugehörige Konzepte

„Hinweise zu REFRESH CLUSTER für Publish/Subscribe-Cluster" auf Seite 113
Die Ausgabe des Befehls **REFRESH CLUSTER** führt dazu, dass der Warteschlangenmanager vorübergehend lokal gehaltene Informationen zu einem Cluster löscht, einschließlich aller Clusterthemen und der zugehörigen Proxy-Subskriptionen.

Zugehörige Verweise

Anwendungsprobleme bei der Ausführung von REFRESH CLUSTER

MQSC-Befehlsreferenz: REFRESH CLUSTER

Clustering: Verfügbarkeit, Multi-Instanz und Wiederherstellung nach einem Katastrophenfall
Dieser Abschnitt enthält eine Anleitung zur Planung und Verwaltung von IBM MQ-Clustern. Diese Informationen sind ein Leitfaden, der auf Tests und Feedback von Kunden basiert.

IBM MQ-Clustering ist eigentlich keine Hochverfügbarkeitslösung, aber unter bestimmten Umständen kann sie verwendet werden, um die Verfügbarkeit von Services unter Verwendung von IBM MQ zu verbessern, z. B. wenn mehrere Instanzen einer Warteschlange auf verschiedenen Warteschlangenmanagern vorhanden sind. Dieser Abschnitt enthält Anleitungen um sicherzustellen, dass die IBM MQ-Infrastruktur eine möglichst hohe Verfügbarkeit aufweist, damit sie in einer solchen Architektur verwendet werden kann.

Anmerkung: Weitere Hochverfügbarkeits-und Disaster-Recovery-Lösungen sind für IBM MQverfügbar (siehe Hochverfügbarkeit, Wiederherstellung und Neustart konfigurieren).

Verfügbarkeit von Clusterressourcen
Der Grund für die übliche Empfehlung, zwei vollständige Repositorys zu verwalten, besteht darin, dass der Verlust von einem nicht kritisch für die reibungslose Ausführung des Clusters ist. Selbst wenn beide nicht mehr verfügbar sind, gibt es eine 60-Tage-Karenzzeit für vorhandene Kenntnisse, die von Teilrepositorys gehalten werden, obwohl neue oder nicht zuvor aufgerufene Ressourcen (z. B. Warteschlangen) in diesem Ereignis nicht verfügbar sind.

Verwenden von Clustern zur Verbesserung der Anwendungsverfügbarkeit
Ein Cluster kann Ihnen bei der Entwicklung hoch verfügbarer Anwendungen (z. B. einer Serveranwendung für Anforderungen/Antworttyp) helfen, indem Sie mehrere Instanzen der Warteschlange und der Anwendung verwenden. Falls erforderlich, können Prioritätsattribute die Anwendung "live" bevorzugen, es sei denn, ein WS-Manager oder Kanal ist beispielsweise nicht verfügbar. Dies ist leistungsfähig, um schnell umschalten zu können, um die Verarbeitung neuer Nachrichten fortzusetzen, wenn ein Problem auftritt.

Nachrichten, die einem bestimmten Warteschlangenmanager in einem Cluster zugestellt wurden, werden jedoch nur in dieser Warteschlangeninstanz gehalten und stehen erst dann zur Verarbeitung zur Verfügung, wenn dieser WS-Manager wiederhergestellt wird. Aus diesem Grund sollten Sie für die hohe Verfügbarkeit von Daten möglicherweise andere Technologien, wie z. B. Warteschlangenmanager mit mehreren Instanzen, berücksichtigen.

Warteschlangenmanager mit mehreren Instanzen
Software mit hoher Verfügbarkeit (mehrere Instanzen) ist ein integriertes Angebot, damit Ihre vorhandenen Nachrichten verfügbar bleiben. Weitere Informationen finden Sie unter IBM MQ mit Konfigurationen mit hoher Verfügbarkeit verwenden, Erstellen eines Multi-Instanz-Warteschlangenmanagers und im folgenden Abschnitt. Für jeden Warteschlangenmanager in einem Cluster kann mit diesem Verfahren eine hohe Verfügbarkeit erreicht werden, sofern auf allen Warteschlangenmanagern im Cluster mindestens IBM WebSphere MQ 7.0.1 ausgeführt wird. Wenn sich die WS-Manager im Cluster auf früheren Ebenen befinden, verlieren sie möglicherweise die Verbindung zu den Multi-Instanz-WS-Managern, wenn sie zu einem sekundären IP-System umschlagen.

Wie bereits in diesem Artikel beschrieben, solange zwei vollständige Repositorys konfiguriert sind, sind sie fast von ihrer Natur hoch verfügbar. Falls IBM MQ mit hoher Verfügbarkeit benötigt wird, können Multi-Instanz-Warteschlangenmanager für vollständige Repositorys verwendet werden. Es gibt keinen Grund, diese Methoden zu verwenden, und in der Tat für temporäre Ausfälle können diese Methoden während des Failover zusätzliche Leistungskosten verursachen. Wenn Sie die Software HA verwenden, anstatt zwei vollständige Repositorys auszuführen, wird davon abgeraten, weil z. B. bei einem einzelnen Kanalausfall nicht notwendigerweise ein Fehler fehlschlagen würde, aber es kann vorkommen, dass Teilrepositorys nicht in der Lage sind, Clusterressourcen abzufragen.

Wiederherstellung nach einem Katastrophenfall
Eine Disaster-Recovery, z. B. die Wiederherstellung nach einer Beschädigung der Platten, auf denen die Daten eines Warteschlangenmanagers gespeichert waren, ist zwar schwierig, wird von IBM MQ aber unterstützt, obwohl der Vorgang nicht automatisch ausgeführt werden kann. Die einzige "wahre" Disaster-Revovery-Option in IBM MQ (abgesehen von den durch das Betriebssystem bereitgestellten oder sonstigen grundlegenden Replikationstechnologien) ist die Wiederherstellung aus einer Sicherung. Es gibt einige Cluster-spezifische Punkte, die in diesen Situationen zu berücksichtigen sind:

- Gehen Sie beim Testen von Szenarios zur Notfallwiederherstellung sorgfältig vor. Wenn Sie beispielsweise den Betrieb von Sicherungswarteschlangenmanagern testen, müssen Sie darauf achten, dass sie in demselben Netz online sind, da es möglich ist, versehentlich den Live-Cluster zu verbinden und die 'Stealing' -Nachrichten zu starten, indem sie dieselben benannten Warteschlangen wie in den Live-Cluster-WS-Managern enthalten.
- Der Test zur Wiederherstellung nach einem Katastrophenfall darf nicht in einen aktiven Live-Cluster eingreifen. Zu den Techniken zur Vermeidung von Kollisionen gehören:
  - Vollständige Netztrennung oder Trennung auf Firewall-Ebene.
  - z/OS Kanalinitialisierung oder z/OS **chinit** -Adressraum wird nicht gestartet.
  - Es wird kein Live-TLS-Zertifikat für das System zur Wiederherstellung nach einem Katastrophenfall ausgegeben, oder es sei denn, es tritt ein tatsächliches Fehlerbehebungsszenario
- Wenn Sie eine Sicherung eines Warteschlangenmanagers im Cluster wiederherstellen, ist es möglich, dass die Sicherung nicht mehr mit dem Rest des Clusters synchronisiert ist. Der Befehl **REFRESH CLUSTER** kann Aktualisierungen auflösen und mit dem Cluster synchronisieren, aber der Befehl **REFRESH CLUSTER** muss als letzte Möglichkeit verwendet werden. Siehe „Clustering: Best Practices für REFRESH CLUSTER verwenden” auf Seite 74. Bevor Sie den Befehl verwenden, überprüfen Sie in der unternehmensinternen Dokumentation und der IBM MQ-Dokumentation, ob irgendein einfacher Schritt versäumt wurde.
- Wie bei jeder Wiederherstellung müssen die Anwendungen mit der Wiedergabe und dem Verlust von Daten umgehen. Es muss entschieden werden, ob die Warteschlangen in einem bekannten Status gelöscht werden sollen oder ob genügend Informationen vorhanden sind, um die Replays zu verwalten.

Verteiltes Publish/Subscribe-Netz planen

Sie können ein Netz von Warteschlangenmanagern erstellen, in denen Subskriptionen, die auf einem Warteschlangenmanager erstellt wurden, übereinstimmende Nachrichten empfangen, die von einer Anwendung veröffentlicht werden, die mit einem anderen Warteschlangenmanager im Netz verbunden ist. Um eine geeignete Topologie auszuwählen, müssen Sie Ihre Anforderungen für die manuelle Steuerung, die Netzgröße, die Häufigkeit von Änderungen, die Verfügbarkeit und die Skalierbarkeit in Betracht ziehen.

Vorbereitende Schritte

Diese Task setzt voraus, dass Sie wissen, was verteilte Publish/Subscribe-Netze sind und wie sie funktionieren. Eine technische Übersicht finden Sie unter Verteilte Publish/Subscribe-Netzwerke.

Informationen zu diesem Vorgang

Es gibt drei grundlegende Topologien für ein Publish/Subscribe-Netzwerk:

- Direkt geroutete Cluster
- Topic-Host-Routing-Cluster
- Hierarchie

Bei den ersten beiden Topologien ist der Ausgangspunkt eine IBM MQ-Clusterkonfiguration. Die dritte Topologie kann mit oder ohne Cluster erstellt werden. Weitere Informationen zur Planung des zugrundeliegenden Warteschlangenmanagernetzes finden Sie unter „Verteilte Warteschlangen und Cluster planen” auf Seite 21.

Ein *Direkt-Routing-Cluster* ist die einfachste Topologie, die konfiguriert werden soll, wenn ein Cluster bereits vorhanden ist. Jedes Thema, das Sie in jedem WS-Manager definieren, wird automatisch auf jedem WS-Manager im Cluster zur Verfügung gestellt, und die Veröffentlichungen werden direkt von jedem Warteschlangenmanager weitergeleitet, auf dem eine Veröffentlichungsanwendung eine Verbindung herstellt, zu jedem der Warteschlangenmanager, für die übereinstimmende Subskriptionen vorhanden sind. Voraussetzung für eine solch einfache Konfiguration ist das hohe Maß an gemeinsamer Nutzung von Informationen und an Konnektivität zwischen den Warteschlangenmanagern eines Cluster, das IBM MQ ermöglicht. Für kleine und einfache Netze (d. a. eine kleine Anzahl von Warteschlangenmanagern

und eine relativ statische Gruppe von Publishern und Subskribenten) ist dies akzeptabel. Wenn der Systemaufwand jedoch in größeren oder dynamischeren Umgebungen verwendet wird, ist dies möglicherweise untragbar. Siehe„Direktes Routing in Publish/Subscribe-Clustern" auf Seite 83.

Ein *Topic-Host-Routing-Cluster* bietet denselben Vorteil wie ein direkter weitergeleitete Cluster, indem Sie jedes Thema, das Sie in jedem WS-Manager im Cluster definieren, automatisch auf jedem WS-Manager im Cluster verfügbar machen. Bei den Host-Routing-Clustern müssen Sie jedoch die Warteschlangenmanager, die die einzelnen Themen enthalten, sorgfältig auswählen, da alle Informationen und Veröffentlichungen zu diesem Thema diese Topic-Host-WS-Manager durchlaufen. Dies bedeutet, dass das System die Kanäle und Informationsflüsse nicht zwischen allen Warteschlangenmanagern verwalten muss. Dies bedeutet jedoch auch, dass Veröffentlichungen möglicherweise nicht mehr direkt an Subskribenten gesendet werden, sondern möglicherweise über einen Topic-Host-Warteschlangenmanager weitergeleitet werden. Aus diesen Gründen kann es zu einer zusätzlichen Belastung des Systems kommen, insbesondere auf den Warteschlangenmanagern, die die Themen hosten, so dass eine sorgfältige Planung der Topologie erforderlich ist. Diese Topologie ist besonders effektiv für Netze, die viele Warteschlangenmanager enthalten, oder die eine dynamische Gruppe von Publishern und Subskribenten (d. a. veröffentlichende Stellen oder Subskribenten, die häufig hinzugefügt oder entfernt werden) enthalten. Es können zusätzliche Themenhosts definiert werden, um die Verfügbarkeit von Routen zu verbessern und die Publikationsworkload horizontal skalieren zu lassen. Siehe„Thema Host-Routing in Publish/Subscribe-Clustern" auf Seite 88.

Eine *Hierarchie* erfordert die Konfiguration der meisten manuellen Konfigurationen und ist die schwierigste Topologie, die geändert werden soll. Sie müssen die Beziehungen zwischen den einzelnen Warteschlangenmanagern in der Hierarchie und ihren direkten Beziehungen manuell konfigurieren. Nach der Konfiguration von Beziehungen werden die Veröffentlichungen (wie bei den vorherigen beiden Topologien) an Subskriptionen auf anderen Warteschlangenmanagern in der Hierarchie weitergeleitet. Veröffentlichungen werden unter Verwendung der Hierarchiebeziehungen weitergeleitet. Dadurch können sehr spezifische Topologien so konfiguriert werden, dass sie unterschiedlichen Anforderungen entsprechen, aber es kann auch dazu führen, dass Veröffentlichungen, die viele " Hops " erfordern, über temporäre Warteschlangenmanager die Subskriptionen erreichen. Es gibt immer nur eine Route durch eine Hierarchie für eine Veröffentlichung, so dass die Verfügbarkeit jedes Warteschlangenmanagers kritisch ist. Hierarchien sind in der Regel nur dann vorzuziehen, wenn ein einzelner Cluster nicht konfiguriert werden kann, z. B. wenn er mehrere Organisationen umfasst. Siehe„Routing in Publish/Subscribe-Hierarchien" auf Seite 114.

Bei Bedarf können die oben genannten drei Topologien kombiniert werden, um spezifische topographische Anforderungen zu lösen. Ein Beispiel finden Sie unter Kombinieren der Topic-Bereiche mehrerer Cluster .

Wenn Sie eine geeignete Topologie für Ihr verteiltes Publish/Subscribe-Netz auswählen möchten, müssen Sie die folgenden allgemeinen Fragen berücksichtigen:

- Wie groß wird Ihr Netzwerk sein?
- Wie viel manuelle Steuerung benötigen Sie über die Konfiguration?
- Wie dynamisch wird das System sowohl in Bezug auf Themen als auch in Bezug auf die Subskriptionen und in Bezug auf die Warteschlangenmanager sein?
- Was sind Ihre Verfügbarkeits-und Skalierbarkeitsanforderungen?
- Können alle WS-Manager direkt miteinander verbunden werden?

Prozedur

- Schätzen Sie, wie groß Ihr Netzwerk sein muss.
  a) Schätzen Sie, wie viele Themen Sie benötigen.
  b) Schätzen Sie, wie viele Publisher und Abonnenten Sie erwarten.
  c) Schätzen Sie, wie viele WS-Manager an Publish/Subscribe-Aktivitäten beteiligt sein werden.

  Weitere Informationen finden Sie unter „Publish/Subscribe-Clustering: Bewährte Verfahren" auf Seite 98, insbesondere in folgenden Abschnitten:

- Vorgehensweise zum Anpassen der Größe Ihres Systems
- Gründe für die Begrenzung der Anzahl von an Publish/Subscribe-Aktivitäten beteiligten Clusterwarteschlangenmanagern
- Vorgehensweise bei der Entscheidung, welche Themen zu einem Cluster gehören sollen

Wenn Ihr Netz über viele Warteschlangenmanager verfügt und viele Publisher und Subskribenten verarbeiten kann, müssen Sie wahrscheinlich einen Topic-Host-Routing-Cluster oder eine Hierarchie verwenden. Direkt verlegte Cluster erfordern fast keine manuelle Konfiguration und können eine gute Lösung für kleine oder statische Netzwerke sein.

- Überlegen Sie, wie viel manuelle Steuerung Sie benötigen, über welchen Warteschlangenmanager die einzelnen Themen, Bereitsteller oder Subskribenten gehostet werden.

  a) Überlegen Sie, ob einige Ihrer WS-Manager weniger fähig sind als andere.

  b) Überlegen Sie, ob die Kommunikationsverbindungen zu einigen Ihrer WS-Manager empfindlicher als andere sind.

  c) Geben Sie Fälle an, in denen Sie ein Thema mit vielen Veröffentlichungen und nur wenigen Subskribenten erwarten.

  d) Geben Sie Fälle an, in denen Sie ein Thema erwarten, das viele Subskribenten und nur wenige Veröffentlichungen enthält.

  In allen Topologien werden Veröffentlichungen an Subskriptionen auf anderen Warteschlangenmanagern zugestellt. In einem direkt weitergeleiteten Cluster nehmen diese Veröffentlichungen den kürzesten Pfad zu den Subskriptionen an. In einem Topic-Host-Routing-Cluster oder einer Hierarchie steuern Sie die Route, die von den Veröffentlichungen ausgeführt wird. Wenn sich Ihre Warteschlangenmanager in ihrer Funktionalität unterscheiden oder unterschiedliche Verfügbarkeits-und Konnektivitätsstufen aufweisen, möchten Sie bestimmte Workloads bestimmten Warteschlangenmanagern zuordnen. Sie können dies entweder mit einem Topic-Host-Routing-Cluster oder mit einer Hierarchie ausführen.

  In allen Topologien ist es möglich, die Veröffentlichungsanwendungen auf demselben Warteschlangenmanager wie die Subskriptionen zu lokalisieren, wenn dies möglich ist, um die Leistung zu minimieren und die Leistung zu maximieren. Für Topic-Host-Routing-Cluster können Sie Publisher oder Subskribenten in den Warteschlangenmanagern, die das Thema hosten, in Betracht ziehen. Dadurch werden alle zusätzlichen " Hops " zwischen WS-Managern entfernt, um eine Veröffentlichung an einen Subskribenten zu übergeben. Dieser Ansatz ist besonders effektiv in Fällen, in denen ein Thema viele veröffentlichende Stellen und wenige Abonnenten hat, oder viele Abonnenten und nur wenige Verlage. Siehe z. B. Topic-Host-Routing mit zentralisierten Publishern oder Subskribenten .

  Weitere Informationen finden Sie unter „Publish/Subscribe-Clustering: Bewährte Verfahren” auf Seite 98, insbesondere in folgenden Abschnitten:

  - Vorgehensweise bei der Entscheidung, welche Themen zu einem Cluster gehören sollen
  - Publisher- und Subskriptionspositionen

- Überlegen Sie, wie dynamisch die Netzaktivität sein wird.

  a) Schätzen Sie, wie häufig Subskribenten zu verschiedenen Themen hinzugefügt und entfernt werden.

  Immer wenn eine Subskription aus einem Warteschlangenmanager hinzugefügt oder aus einem Warteschlangenmanager entfernt wird und die erste oder letzte Subskription für diese bestimmte Themenzeichenfolge ist, werden diese Informationen anderen Warteschlangenmanagern in der Topologie mitgeteilt. In einem direkt weitergeleiteten Cluster und einer Hierarchie werden diese Subskriptionsinformationen an alle Warteschlangenmanager in der Topologie weitergegeben, unabhängig davon, ob sie über Publisher für das Thema verfügen. Wenn die Topologie aus vielen Warteschlangenmanagern besteht, kann dies zu einem erheblichen Leistungsaufwand führen. In einem Host-Routing-Cluster werden diese Informationen nur an die WS-Manager weitergegeben, die ein Clusterthema enthalten, das der Themenzeichenfolge der Subskription zugeordnet ist.

  Weitere Informationen finden Sie unter „Publish/Subscribe-Clustering: Bewährte Verfahren” auf Seite 98 im Abschnitt Subskriptionsänderung und dynamische Themenzeichenfolgen.

Anmerkung: In sehr dynamischen Systemen, bei denen die Gruppe vieler eindeutiger Themenzeichenfolgen schnell und ständig geändert wird, kann es am besten sein, das Modell in den Modus " publish überall " umzuschalten. Weitere Informationen finden Sie im Abschnitt Subskriptionsleistung in Publish/Subscribe-Netzen.

b) Überlegen Sie, wie dynamisch die WS-Manager in der Topologie sind.

Eine Hierarchie erfordert jede Änderung des Warteschlangenmanagers in der Topologie, die manuell in die Hierarchie eingefügt oder aus der Hierarchie entfernt werden muss, wobei die Änderungen beim Ändern von Warteschlangenmanagern auf höheren Ebenen in der Hierarchie berücksichtigt werden müssen. WS-Manager in einer Hierarchie verwenden normalerweise auch manuell konfigurierte Kanalverbindungen. Sie müssen diese Verbindungen verwalten, Kanäle hinzufügen und entfernen, da Warteschlangenmanager hinzugefügt und aus der Hierarchie entfernt werden.

In einem Publish/Subscribe-Cluster werden WS-Manager automatisch mit jedem anderen Warteschlangenmanager verbunden, der erforderlich ist, wenn er zum ersten Mitglied des Clusters ist, und wird automatisch über Themen und Subskriptionen informiert.

- Berücksichtigen Sie die Anforderungen an die Verfügbarkeit und die Skalierbarkeit des Veröffentlichungsverkehrs.
  a) Entscheiden Sie, ob Sie immer eine verfügbare Route von einem Veröffentlichungswarteschlangenmanager zu einem subskribierenden Warteschlangenmanager haben müssen, selbst wenn ein Warteschlangenmanager nicht verfügbar ist.
  b) Berücksichtigen Sie, wie skalierbar Sie das Netz benötigen. Entscheiden Sie, ob die Ebene des Veröffentlichungsdatenverkehrs zu hoch ist, um durch einen einzelnen Warteschlangenmanager oder Kanal weitergeleitet zu werden, und ob diese Ebene des Veröffentlichungsdatenverkehrs von einem einzelnen Topic-Zweig bearbeitet werden muss oder sich über mehrere Topic-Verzweigungen verteilen kann.
  c) Überlegen Sie, ob die Nachrichtenreihenfolge beibehalten werden muss.

  Da ein direkter Routing-Cluster Nachrichten direkt von Veröffentlichungswarteschlangenmanagern zum Subskribieren von Warteschlangenmanagern sendet, müssen Sie die Verfügbarkeit von temporären Warteschlangenmanagern entlang der Route nicht berücksichtigen. Ebenso wird die Skalierung auf die temporären WS-Manager nicht berücksichtigt. Wie bereits erwähnt, kann der Aufwand für die automatische Verwaltung von Kanälen und Informationsflüssen zwischen allen Warteschlangenmanagern im Cluster jedoch erhebliche Auswirkungen auf die Leistung haben, insbesondere in einer großen oder dynamischen Umgebung.

  Ein Topic-Host-Routing-Cluster kann für einzelne Themen optimiert werden. Sie können sicherstellen, dass jede Verzweigung der Themenstruktur, die über eine beachtliche Veröffentlichungsworkload verfügt, auf einem anderen Warteschlangenmanager definiert ist und dass jeder Warteschlangenmanager ausreichend performant ist und für die erwartete Auslastung für diese Verzweigung der Themenstruktur verfügbar ist. Sie können die Verfügbarkeit und die horizontale Skalierung auch verbessern, indem Sie die einzelnen Themen auf mehreren Warteschlangenmanagern definieren. Auf diese Weise kann das System den Host-WS-Managern des Topics nicht mehr verfügbar machen und die Auslastung des Publikationsdatenverkehrs in der Lastverteilung auf sie Wenn Sie jedoch ein bestimmtes Thema auf mehreren Warteschlangenmanagern definieren, führen Sie außerdem die folgenden Einschränkungen durch:

  – Sie verlieren die Nachrichtenreihenfolge in allen Veröffentlichungen.
  – Es ist nicht möglich, ständige Veröffentlichungen zu verwenden. Siehe„Designüberlegungen zu ständigen Veröffentlichungen in Publish/Subscribe-Clustern” auf Seite 111.

  Sie können keine hohe Verfügbarkeit oder Skalierbarkeit von Routing in einer Hierarchie über mehrere Routen konfigurieren.

  Weitere Informationen finden Sie unter „Publish/Subscribe-Clustering: Bewährte Verfahren” auf Seite 98 im Abschnitt Veröffentlichungsdatenverkehr.
- Basierend auf diesen Berechnungen verwenden Sie die Links, die Ihnen bei der Entscheidung helfen, ob ein Topic-Host-Routing-Cluster, ein direkter Routing-Cluster, eine Hierarchie oder eine Mischung dieser Topologien verwendet werden soll.

Nächste Schritte

Sie können jetzt Ihr verteiltes Publish/Subscribe-Netz konfigurieren.

Zugehörige Tasks

WS-Manager-Cluster konfigurieren

Verteilte Warteschlangensteuerung konfigurieren

Publish/Subscribe-Cluster konfigurieren

WS-Manager mit einer Publish/Subscribe-Hierarchie verbinden

Publish/Subscribe-Cluster entwerfen

Es gibt zwei grundlegende Publish/Subscribe-Clustertopologien: *direktes Routing* und *Topic-Host-Routing* . Jeder hat unterschiedliche Vorteile. Wenn Sie Ihren Publish/Subscribe-Cluster entwerfen, wählen Sie die Topologie aus, die am besten zu den erwarteten Netzanforderungen passt.

Eine Übersicht über die beiden Publish/Subscribe-Clustertopologien finden Sie unter Publish/Subscribe-Cluster. Informationen zum Auswerten der Netzanforderungen finden Sie in „Verteiltes Publish/Subscribe-Netz planen” auf Seite 77 und „Publish/Subscribe-Clustering: Bewährte Verfahren” auf Seite 98.

Im Allgemeinen bieten beide Clustertopologien die folgenden Vorteile:

- Einfache Konfiguration über eine Punkt-zu-Punkt-Clustertopologie.
- Automatischer Umgang mit Warteschlangenmanagern, die den Cluster verbinden und verlassen.
- Skalieren Sie die Skalierung für zusätzliche Subskriptionen und Publisher, indem Sie zusätzliche Warteschlangenmanager hinzufügen und die zusätzlichen Subskriptionen und Publisher auf diese verteilen.

Allerdings weisen die beiden Topologien unterschiedliche Vorteile auf, da die Anforderungen spezifischer werden.

Direkt weitergeleitete Publish/Subscribe-Cluster

Bei einem direkten Routing sendet jeder WS-Manager im Cluster Veröffentlichungen von verbundenen Anwendungen direkt an jeden anderen Warteschlangenmanager im Cluster mit einer übereinstimmenden Subskription.

Ein Publish/Subscribe-Cluster mit direkter Weiterleitung bietet die folgenden Vorteile:

- Nachrichten, die für eine Subskription auf einem bestimmten Warteschlangenmanager im selben Cluster bestimmt sind, werden direkt zu diesem Warteschlangenmanager transportiert und müssen keinen temporären Warteschlangenmanager durchlaufen. Dies kann die Leistung im Vergleich zu einer Topologie mit Topic-Host oder einer hierarchischen Topologie verbessern.
- Da alle WS-Manager direkt miteinander verbunden sind, gibt es in der Routing-Infrastruktur dieser Topologie keinen Single Point of Failure. Wenn ein Warteschlangenmanager nicht verfügbar ist, können Subskriptionen auf anderen WS-Managern im Cluster weiterhin Nachrichten von Publishern auf verfügbaren Warteschlangenmanagern empfangen.
- Es ist sehr einfach zu konfigurieren, insbesondere auf einem vorhandenen Cluster.

Bei Verwendung eines direkt weitergeleiteten Publish/Subscribe-Clusters ist Folgendes zu beachten:

- Alle WS-Manager im Cluster werden von allen anderen Warteschlangenmanagern im Cluster informiert.
- Warteschlangenmanager in einem Cluster, die eine oder mehrere Subskriptionen für ein Clusterthema hosten, erstellen automatisch Clustersenderkanäle zu allen anderen WS-Managern im Cluster, auch wenn diese Warteschlangenmanager keine Nachrichten in einem Clusterthema veröffentlichen.
- Die erste Subskription auf einem WS-Manager in einer Themenzeichenfolge unter einem Clusterthema führt dazu, dass eine Nachricht an alle anderen Warteschlangenmanager im Cluster gesendet wird. In ähnlicher Weise wird auch die letzte Subskription für eine zu löschende Themenzeichenfolge in einer Nachricht angezeigt. Je mehr einzelne Themenzeichenfolgen in einem Clusterthema verwendet werden und um so höher ist die Änderungsrate der Subskriptionen, so dass die Kommunikation zwischen WS-Managern mehr stattfindet.

- Jeder WS-Manager im Cluster behält die Kenntnis der subskribierten Themenzeichenfolgen, über die er informiert wird, selbst wenn der Warteschlangenmanager diese Themen weder veröffentlicht noch subskribiert.

Aus den oben genannten Gründen wird für alle Warteschlangenmanager in einem Cluster mit einem direkt geleiteten Thema ein zusätzlicher Systemaufwand entstehen. Je mehr WS-Manager in dem Cluster vorhanden sind, um so größer ist der Systemaufwand. Auch die mehr Themenzeichenfolgen subskribiert und um so größer ist ihr Änderungsrate, um so größer der Aufwand. Dies kann zu einer zu hohen Auslastung von Warteschlangenmanagern führen, die auf kleinen Systemen in einem großen oder dynamischen Direct-Routing-Publish/Subscribe-Cluster ausgeführt werden. Weitere Informationen finden Sie unter Direct routed Publish/Subscribe performance .

Wenn Sie wissen, dass ein Cluster die Overheads von Direct-Routing-Publish/Subscribe nicht aufnehmen kann, können Sie stattdessen topic host routed publish/subscribe verwenden. Alternativ dazu können Sie die Cluster-Publish/Subscribe-Funktionalität auch in extremen Situationen vollständig inaktivieren, indem Sie das Warteschlangenmanager-Attribut **PSCLUS** auf jedem WS-Manager im Cluster auf `DISABLED` setzen. Siehe „Clusterveröffentlichungs-/Subskriptionssubskribieren” auf Seite 109. Dadurch wird verhindert, dass ein Cluster-Topic erstellt wird, und stellt daher sicher, dass Ihrem Netz keine Überleitung zugeordnet ist, die mit einem Cluster-Publish/Subscribe verknüpft sind.

Topic-Host-Verlegte Publish/Subscribe-Cluster

Bei Topic-Host-Routing werden die Warteschlangenmanager, in denen Clusterthemen administrativ definiert sind, zu Routern für Veröffentlichungen. Veröffentlichungen von Nicht-Hosting-WS-Managern im Cluster werden über den Host-WS-Manager an jeden Warteschlangenmanager im Cluster mit einer übereinstimmenden Subskription weitergeleitet.

Ein Publish/Subscribe-Cluster mit Topic-Host-Routing bietet die folgenden zusätzlichen Vorteile für einen direkt weitergeleiteten Publish/Subscribe-Cluster:

- Nur WS-Manager, auf denen Topic-Host-Routing-Themen definiert sind, werden von allen anderen Warteschlangenmanagern im Cluster informiert.
- Nur der Topic-Host-Warteschlangenmanager muss in der Lage sein, eine Verbindung zu allen anderen Warteschlangenmanagern im Cluster herzustellen. In der Regel wird nur die Verbindung zu den Warteschlangenmanagern hergestellt, in denen Subskriptionen vorhanden sind. Daher gibt es deutlich weniger Kanäle, die zwischen WS-Managern ausgeführt werden.
- Cluster-WS-Manager, die eine oder mehrere Subskriptionen für ein Clusterthema hosten, erstellen Clustersenderkanäle automatisch nur zu Warteschlangenmanagern, die ein Clusterthema enthalten, das der Themenzeichenfolge der Subskription zugeordnet ist.
- Die erste Subskription auf einem WS-Manager in einer Themenzeichenfolge unter einem Clusterthema führt dazu, dass eine Nachricht an einen Warteschlangenmanager im Cluster gesendet wird, in dem das Clusterthema gehostet wird. In ähnlicher Weise wird auch die letzte Subskription für eine zu löschende Themenzeichenfolge in einer Nachricht angezeigt. Je mehr einzelne Themenzeichenfolgen in einem Clusterthema verwendet werden, und je höher die Änderungsrate der Subskriptionen ist, wird die Kommunikation zwischen den WS-Managern, aber nur zwischen Subskriptionshosts und Themenhosts, durchgeführt.
- Größere Kontrollmöglichkeiten bei der physischen Konfiguration. Bei direkter Weiterleitung müssen alle WS-Manager am Publish/Subscribe-Cluster teilnehmen und ihre Overheads erhöhen. Bei der Weiterleitung von Topic-Hosts sind nur die Warteschlangenmanager des Topic-Hosts von anderen Warteschlangenmanagern und deren Subskriptionen bekannt. Sie wählen die Topic-Host-Warteschlangenmanager explizit aus. Daher können Sie sicherstellen, dass diese WS-Manager auf einer geeigneten Ausrüstung ausgeführt werden, und Sie können weniger leistungsfähige Systeme für die anderen Warteschlangenmanager verwenden.

Beim Verwenden eines Publish/Subscribe-Clusters für einen Topic-Host ist Folgendes zu beachten:

- Ein zusätzlicher " Hop " zwischen einem Veröffentlichungswarteschlangenmanager und einem subskribierenden Warteschlangenmanager wird eingeführt, wenn der Bereitsteller oder der Subskribent sich nicht auf einem Topic-Hosting-Warteschlangenmanager befindet. Die Latenzzeit, die durch den zusätz-

lichen " Hop " verursacht wird, kann bedeuten, dass die Routing-Weiterleitung durch das Topic-Host weniger effizient ist.

- Bei großen Clustern vereinfacht das Thema Host-Routing die wichtigen Leistungs-und Skalierungsprobleme, die Sie mit der direkten Weiterleitung erreichen können.
- Sie können alle Themen in einem einzigen Warteschlangenmanager oder in einer sehr kleinen Anzahl von Warteschlangenmanagern definieren. Wenn Sie dies tun, stellen Sie sicher, dass die Host-WS-Manager auf leistungsfähigen Systemen mit guter Konnektivität gehostet werden.
- Sie können dasselbe Thema in mehr als einem Warteschlangenmanager definieren. Dadurch wird die Verfügbarkeit des Themas erhöht und die Skalierbarkeit verbessert, da die IBM MQ-Workload die Veröffentlichungen zu einem Thema auf alle Hosts für dieses Thema verteilt. Beachten Sie jedoch, dass die Definition desselben Themas in mehr als einem Warteschlangenmanager die Nachrichtenreihenfolge für dieses Thema verliert.
- Wenn Sie verschiedene Themen auf verschiedenen Warteschlangenmanagern hosten, können Sie die Skalierbarkeit verbessern, ohne die Nachrichtenreihenfolge zu verlieren.

Zugehörige Tasks

Szenario: Publish/Subscribe-Cluster erstellen

Publish/Subscribe-Cluster konfigurieren

Verteilte Publish/Subscribe-Netze optimieren

Fehlerbehebung bei Problemen mit verteiltem Publish/Subscribe

Direktes Routing in Publish/Subscribe-Clustern

Veröffentlichungen von einem beliebigen Veröffentlichungswarteschlangenmanager werden direkt an alle anderen Warteschlangenmanager im Cluster weitergeleitet, die eine übereinstimmende Subskription aufweisen.

Eine Einführung dazu, wie Nachrichten zwischen Warteschlangenmanagern in Publish/Subscribe-Hierarchien und Clustern weitergeleitet werden, finden Sie unter Verteilte Publish/Subscribe-Netze.

Ein Direct-Routing-Publish/Subscribe-Cluster verhält sich wie folgt:

- Alle WS-Manager kennen automatisch alle anderen Warteschlangenmanager.
- Alle WS-Manager mit Subskriptionen für Clusterthemen erstellen Kanäle zu allen anderen WS-Managern im Cluster und informieren sie über ihre Subskriptionen.
- Nachrichten, die von einer Anwendung veröffentlicht werden, werden von dem Warteschlangenmanager weitergeleitet, mit dem sie verbunden ist, direkt zu jedem Warteschlangenmanager, auf dem eine übereinstimmende Subskription vorhanden ist.

Das folgende Diagramm zeigt einen WS-Manager-Cluster, der derzeit nicht für Publish/Subscribe- oder Punkt-zu-Punkt-Aktivitäten verwendet wird. Es ist zu beachten, dass Warteschlangenmanager im Cluster jeweils nur eine Verbindung zu und von den Warteschlangenmanagern mit vollständigem Repository herstellt.

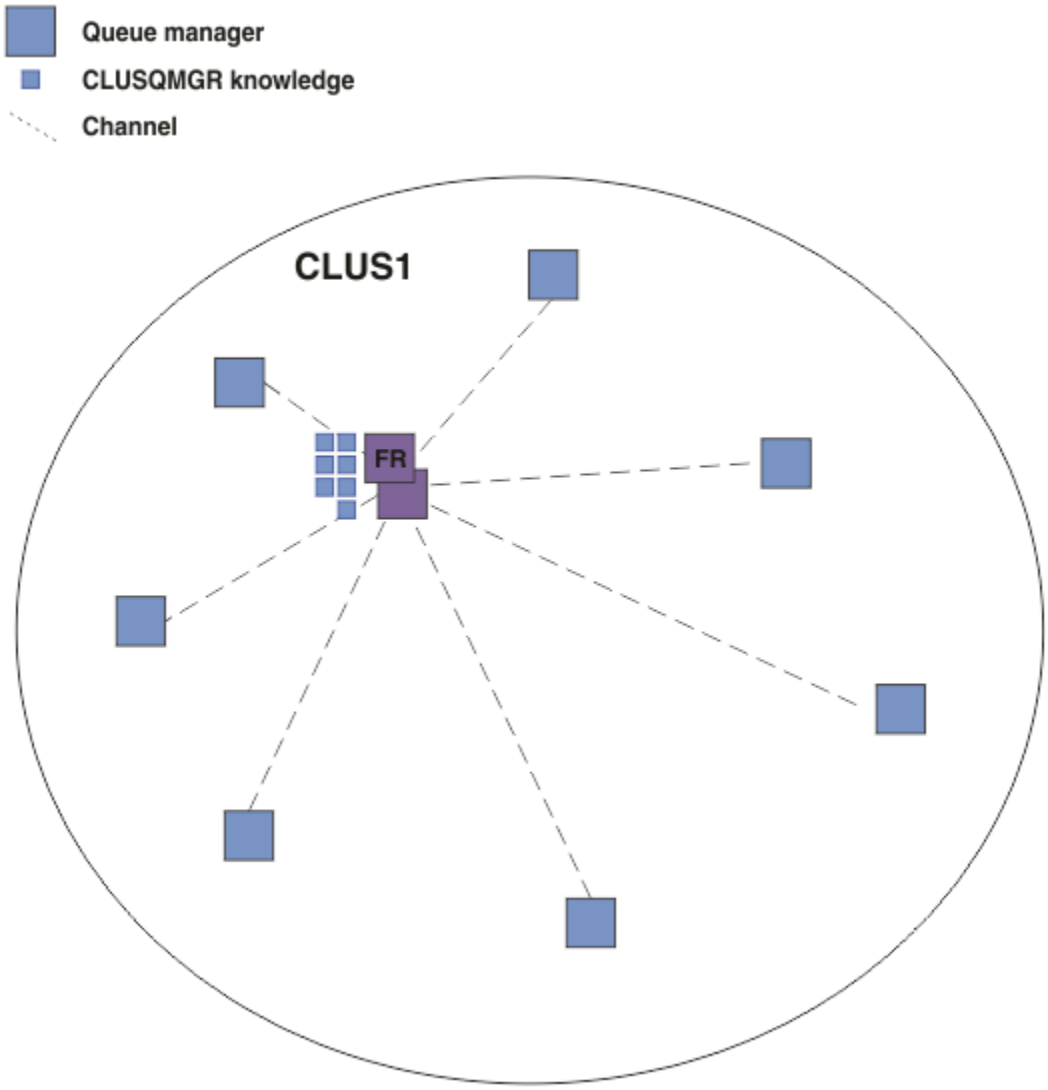


Abbildung 16. Ein WS-Manager-Cluster

Damit Veröffentlichungen zwischen Warteschlangenmanagern in einem direkt weitergeleiteten Cluster fließen können, müssen Sie eine Verzweigung der Themenstruktur wie im Abschnitt Publish/Subscribe-Cluster konfigurieren beschrieben und *direktes Routing* (Standardeinstellung) angeben.

In einem direkt weitergeleiteten Publish/Subscribe-Cluster definieren Sie das Themenobjekt auf jedem WS-Manager im Cluster. Wenn Sie dies tun, werden die Kenntnisse über das Objekt und die Kenntnisse aller anderen Warteschlangenmanager im Cluster automatisch von den vollständigen WS-Managern in den WS-Managern in alle Warteschlangenmanager des Clusters übertragen. Dies geschieht, bevor ein WS-Manager auf das Thema verweist:

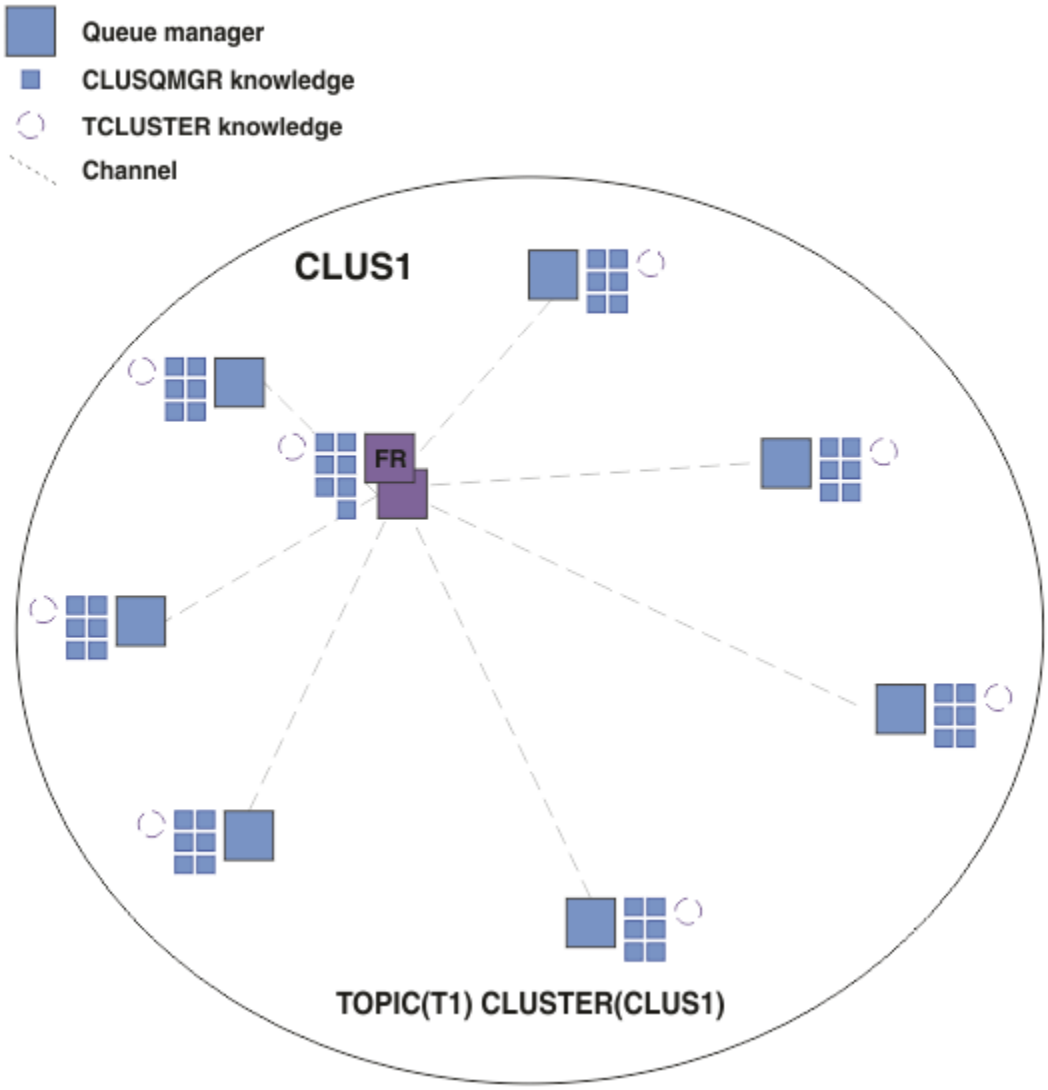


Abbildung 17. Ein Direct-Routing-Publish/Subscribe-Cluster

Wenn eine Subskription erstellt wird, erstellt der Warteschlangenmanager, der die Subskription enthält, einen Kanal zu jedem WS-Manager im Cluster und sendet Details zur Subskription. Dieses verteilte Wissen über Abonnements wird durch ein Proxy-Abonnement auf jedem Warteschlangenmanager dargestellt. Wenn eine Veröffentlichung auf einem Warteschlangenmanager im Cluster erstellt wird, der mit der Themenzeichenfolge dieser Proxy-Subskription übereinstimmt, wird ein Clusterkanal vom Publisher-Warteschlangenmanager zu jedem Warteschlangenmanager eingerichtet, der eine Subskription hostet, und die Nachricht wird an jeden Warteschlangenmanager gesendet.

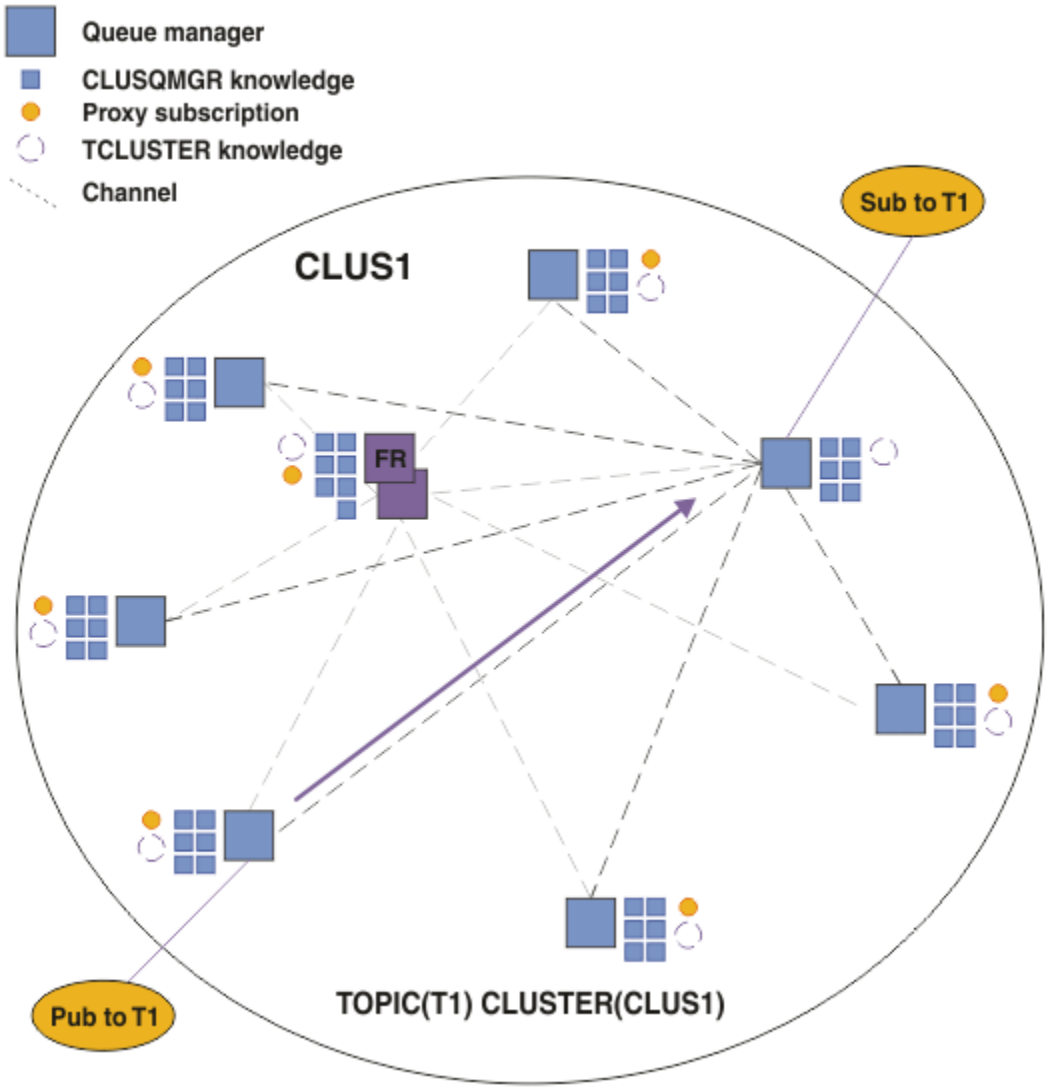


Abbildung 18. Ein direkt weitergeleitete Publish/Subscribe-Cluster mit einem Publisher und einem Subskribenten zu einem Cluster-Thema

Die direkte Weiterleitung von Veröffentlichungen an Subskriptionswarteschlangenmanager vereinfacht die Konfiguration und minimiert die Latenzzeit bei der Bereitstellung von Veröffentlichungen zu Subskriptionen.

Abhängig von der Position der Subskriptionen und Publisher kann Ihr Cluster jedoch schnell vollständig miteinander verbunden werden, wobei jeder WS-Manager eine direkte Verbindung zu jedem anderen Warteschlangenmanager hat. Dies kann in Ihrer Umgebung akzeptabel sein oder nicht. Auch wenn die Gruppe der Themenzeichenfolgen, die subskribiert werden, häufig geändert wird, kann sich der Systemaufwand für die Weitergabe dieser Informationen zwischen allen Warteschlangenmanagern ebenfalls erheblich ändern. Alle Warteschlangenmanager in einem direkt weitergeleiteten Publish/Subscribe-Cluster müssen in der Lage sein, diese Overheads zu bewältigen.

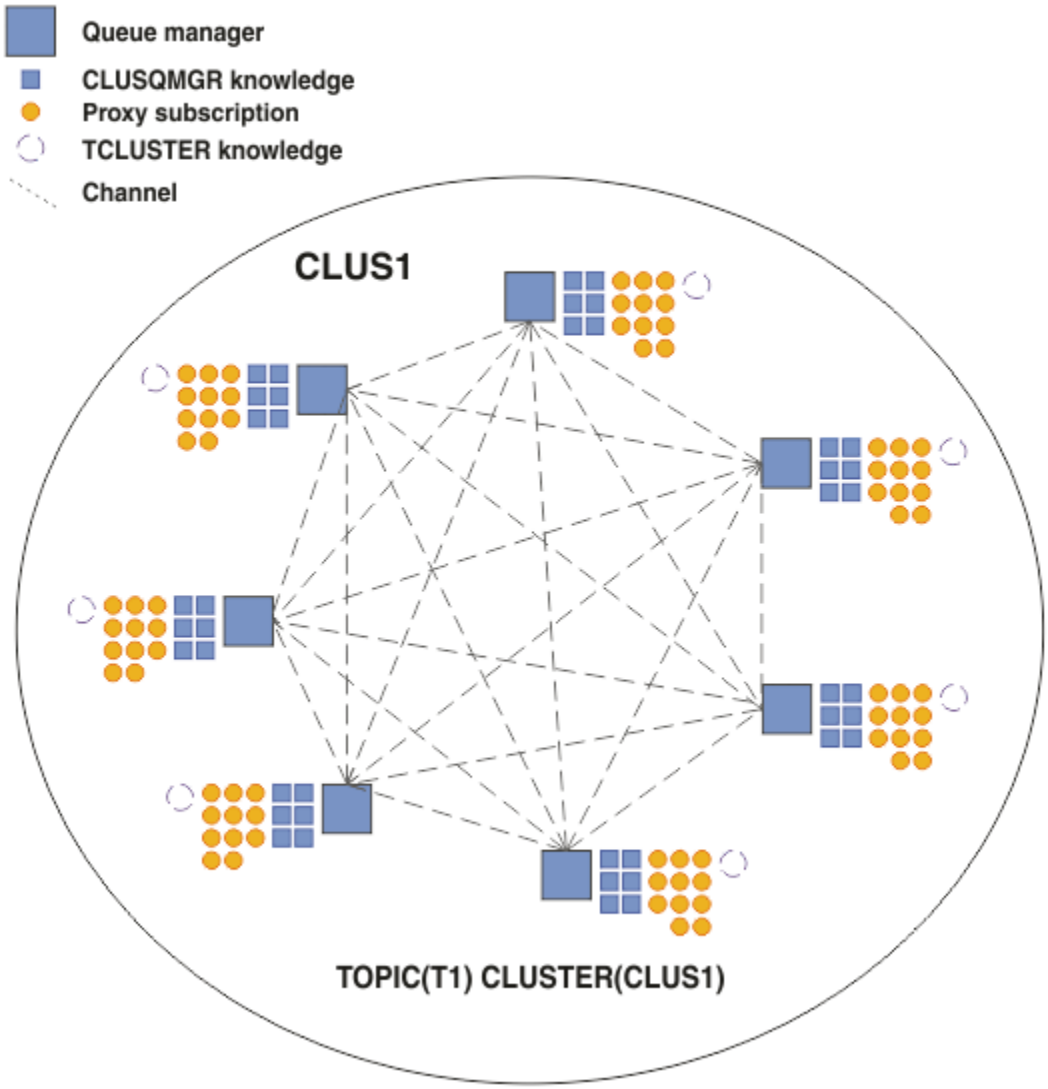


Abbildung 19. Ein direkt weitergeleitete Publish/Subscribe-Cluster, die vollständig miteinander verbunden sind.

Zusammenfassung und weitere Hinweise

Ein direkt weitergeleitete Publish/Subscribe-Cluster benötigt wenig manuellen Eingriff zum Erstellen oder Verwalten von und ermöglicht direktes Routing zwischen Publishern und Subskribenten. Bei bestimmten Konfigurationen ist es in der Regel die am besten geeignete Topologie, insbesondere Cluster mit wenigen Warteschlangenmanagern, oder wenn eine hohe Warteschlangenmanagerkonnektivität akzeptabel ist, und die Subskriptionen selten geändert werden. Es gibt jedoch auch bestimmte Einschränkungen auf Ihrem System:

- Die Auslastung der einzelnen WS-Manager ist proportional zur Gesamtzahl der Warteschlangenmanager im Cluster. Daher können in größeren Clustern einzelne WS-Manager und das System als Ganzes Leistungsprobleme erfahren.
- Standardmäßig werden alle im Cluster subskribierten Themenzeichenfolgen im gesamten Cluster weitergegeben, und die Veröffentlichungen werden nur an ferne Warteschlangenmanager weitergegeben, die über eine Subskription für das zugeordnete Thema verfügen. Daher können schnelle Änderungen an der Gruppe von Subskriptionen zu einem Begrenzungsfaktor werden. Sie können dieses Standardverhalten ändern und stattdessen alle Publizierungsveröffentlichungen an alle Warteschlangenmanager weitergeben, wodurch die Notwendigkeit von Proxy-Subskriptionen entfällt. Dadurch reduziert sich der Austausch der Subskriptionsdaten, der durch Veröffentlichungen bewirkte Datenverkehr sowie mögli-

cherweise die Anzahl der von den Warteschlangenmanagern eingerichteten Kanäle erhöht sich hingegen. Weitere Informationen finden Sie im Abschnitt Subskriptionsleistung in Publish/Subscribe-Netzen.

Anmerkung: Eine ähnliche Einschränkung gilt auch für Hierarchien.

- Aufgrund der Vernetzung von Publish/Subscribe-Queue-Managern dauert es, bis Proxy-Subskriptionen sich über alle Knoten im Netz ausbreiten. Ferne Veröffentlichungen beginnen nicht unbedingt sofort, wenn sie sofort subskribiert werden, so dass frühzeitige Veröffentlichungen möglicherweise nicht nach einer Subskription für eine neue Themenzeichenfolge gesendet werden. Sie können die Probleme, die durch die Subskriptionsverzögerung verursacht werden, entfernen, indem alle Veröffentlichungen an alle Warteschlangenmanager weitergegeben werden, wodurch die Notwendigkeit von Proxy-Subskriptionen entfernt wird. Weitere Informationen finden Sie im Abschnitt Subskriptionsleistung in Publish/Subscribe-Netzen.

  Anmerkung: Diese Einschränkung gilt auch für Hierarchien.

Bevor Sie direktes Routing verwenden, untersuchen Sie die alternativen Ansätze, die in „Thema Host-Routing in Publish/Subscribe-Clustern" auf Seite 88 und „Routing in Publish/Subscribe-Hierarchien" auf Seite 114 detailliert beschrieben sind.

Thema Host-Routing in Publish/Subscribe-Clustern

Veröffentlichungen von Nicht-Hosting-WS-Managern im Cluster werden über den Host-WS-Manager an jeden Warteschlangenmanager im Cluster mit einer übereinstimmenden Subskription weitergeleitet.

Eine Einführung dazu, wie Nachrichten zwischen Warteschlangenmanagern in Publish/Subscribe-Hierarchien und Clustern weitergeleitet werden, finden Sie unter Verteilte Publish/Subscribe-Netze.

Um das Verhalten und die Vorteile von Topic-Host-Routing zu verstehen, ist es am besten, „Direktes Routing in Publish/Subscribe-Clustern" auf Seite 83 zu verstehen.

Der Publish/Subscribe-Cluster eines Topic-Hosts verhält sich wie folgt:

- Gruppierte verwaltete Themenobjekte werden manuell auf einzelnen Warteschlangenmanagern im Cluster definiert. Diese werden als *Topic-Host-Warteschlangenmanager* bezeichnet.
- Wenn eine Subskription für einen Cluster-WS-Manager ausgeführt wird, werden Kanäle vom Subskriptionshostwarteschlangenmanager zum Topic-Host-Warteschlangenmanager erstellt, und Proxy-Subskriptionen werden nur auf den Warteschlangenmanagern erstellt, die das Thema enthalten.
- Wenn eine Anwendung Informationen zu einem Thema veröffentlicht, leitet der verbundene Warteschlangenmanager die Veröffentlichung immer an einen Warteschlangenmanager weiter, der das Thema hostet, das es an alle WS-Manager im Cluster weiterleitet, die übereinstimmende Subskriptionen für das Thema haben.

Dieser Vorgang wird in den folgenden Beispielen näher erläutert.

Thema Hostweiterleitung unter Verwendung eines einzelnen Themenhosts

Damit Veröffentlichungen zwischen Warteschlangenmanagern in einem Topic-Host-Routing-Cluster fließen können, müssen Sie eine Verzweigung der Themenstruktur wie im Abschnitt Publish/Subscribe-Cluster konfigurieren beschrieben erstellen und *Topic-Host-Routing* angeben.

Es gibt eine Reihe von Gründen, ein Topic-Host-Topic-Objekt auf mehreren Warteschlangenmanagern in einem Cluster zu definieren. Der Einfachheit jedoch beginnen wir mit einem einzigen Themenhost zu beginnen.

Das folgende Diagramm zeigt einen WS-Manager-Cluster, der derzeit nicht für Publish/Subscribe- oder Punkt-zu-Punkt-Aktivitäten verwendet wird. Es ist zu beachten, dass Warteschlangenmanager im Cluster jeweils nur eine Verbindung zu und von den Warteschlangenmanagern mit vollständigem Repository herstellt.

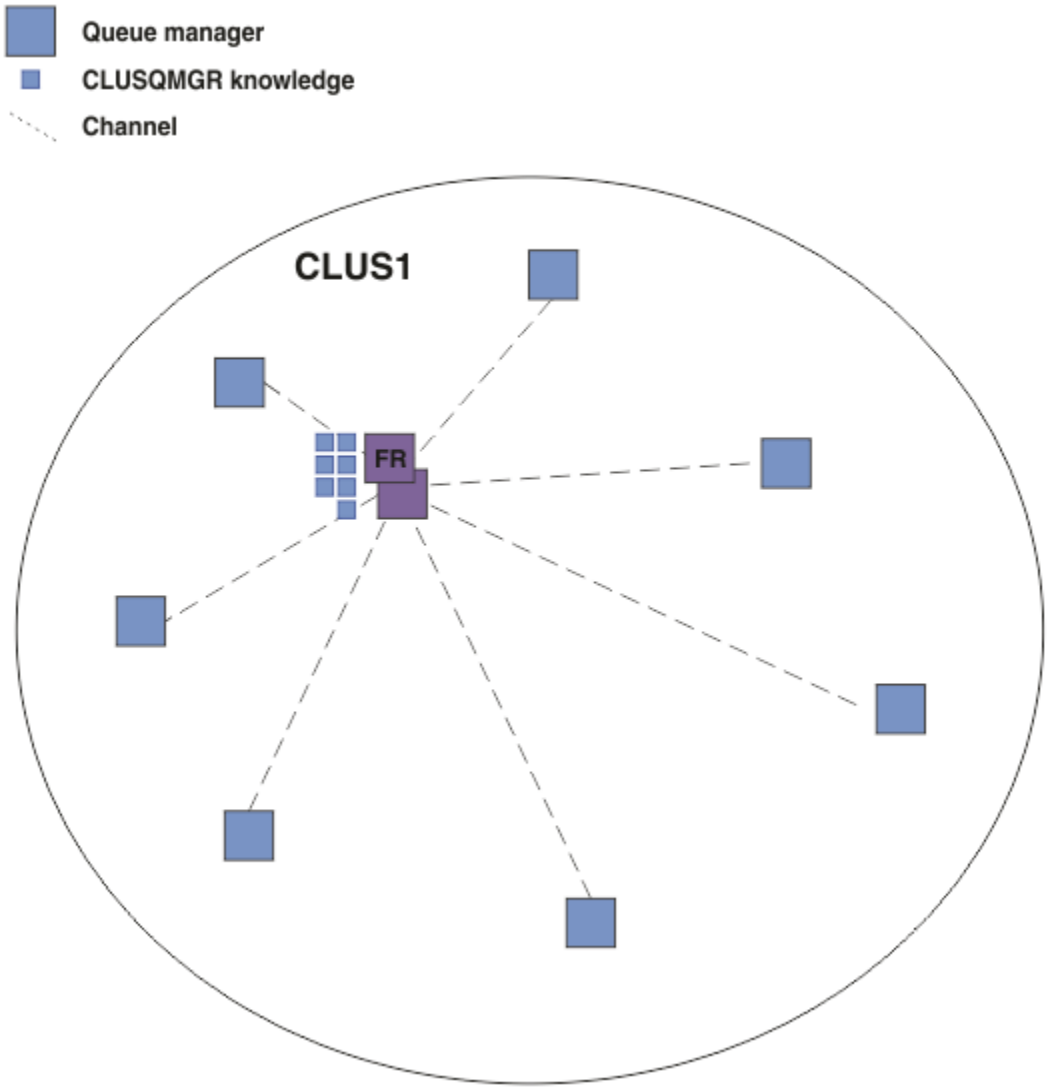


Abbildung 20. Ein WS-Manager-Cluster

In einem Publish/Subscribe-Cluster des Topic-Hosts definieren Sie das Themenobjekt auf einem bestimmten Warteschlangenmanager im Cluster. Der Publish/Subscribe-Datenverkehr fließt dann durch diesen Warteschlangenmanager und macht ihn zu einem kritischen Warteschlangenmanager im Cluster und erhöht seine Auslastung. Aus diesen Gründen wird es nicht empfohlen, einen vollständigen WS-Manager-Repository zu verwenden, aber einen anderen WS-Manager im Cluster zu verwenden. Wenn Sie das Themenobjekt auf dem Host-WS-Manager definieren, wird die Kenntnis des Objekts und seines Hosts automatisch von den vollständigen WS-Managern des Repositorys an alle anderen WS-Manager im Cluster übertragen. Beachten Sie, dass im Gegensatz zu *direktes Routing* jeder Warteschlangenmanager nicht zu jedem anderen WS-Manager im Cluster gehört.

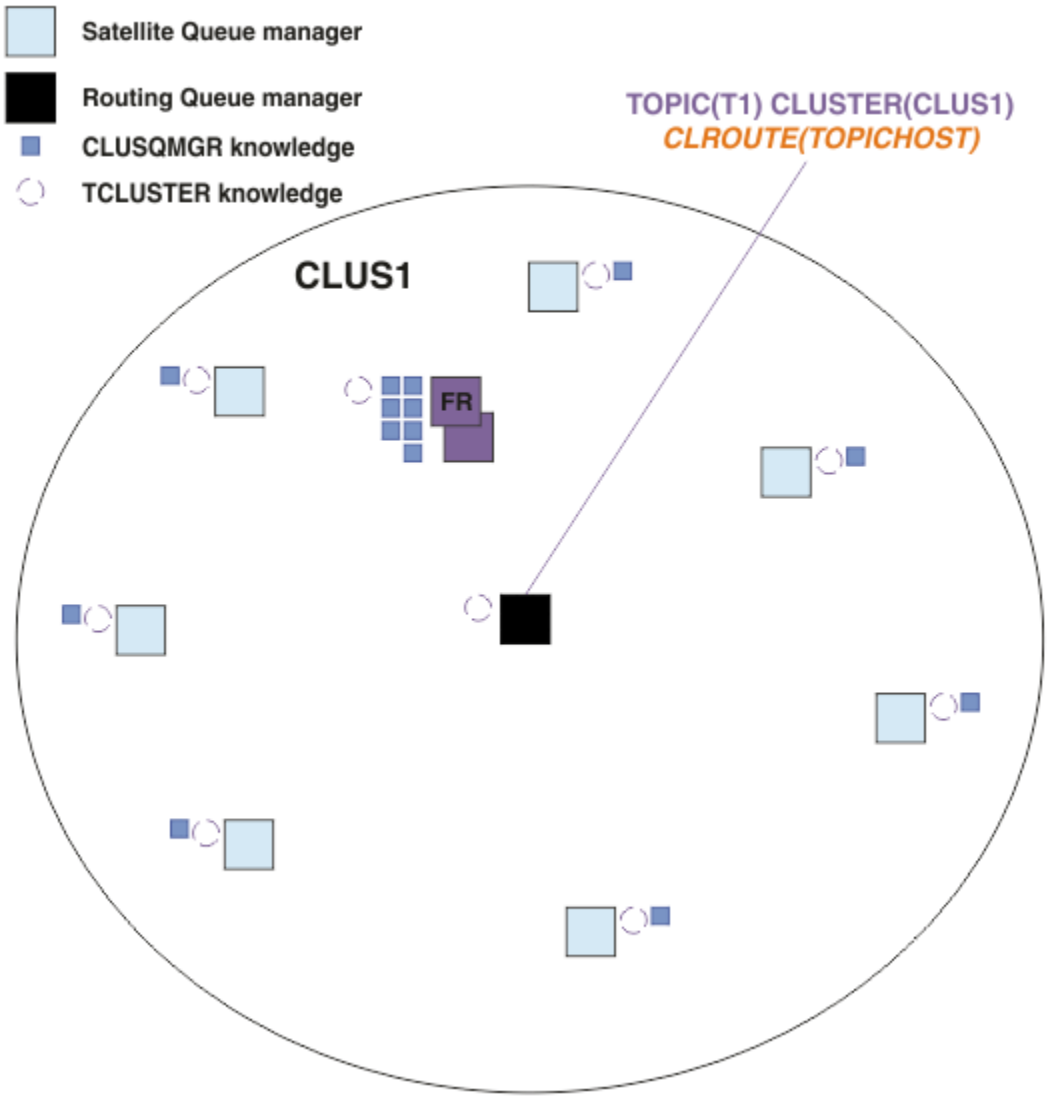


Abbildung 21. Ein Topic-Host-Publish/Subscribe-Cluster mit einem Thema, das auf einem Topic-Host definiert ist

Wenn eine Subskription auf einem Warteschlangenmanager erstellt wird, wird ein Kanal zwischen dem subskribierenden Warteschlangenmanager und dem Topic-Host-Warteschlangenmanager erstellt. Der subskribierende Warteschlangenmanager stellt nur eine Verbindung zum Topic-Host-Warteschlangenmanager her und sendet die Details der Subskription (in Form einer *Proxy-Subskription*). Der Topic-Host-WS-Manager leitet diese Subskriptionsinformationen nicht an alle weiteren Warteschlangenmanager im Cluster weiter.

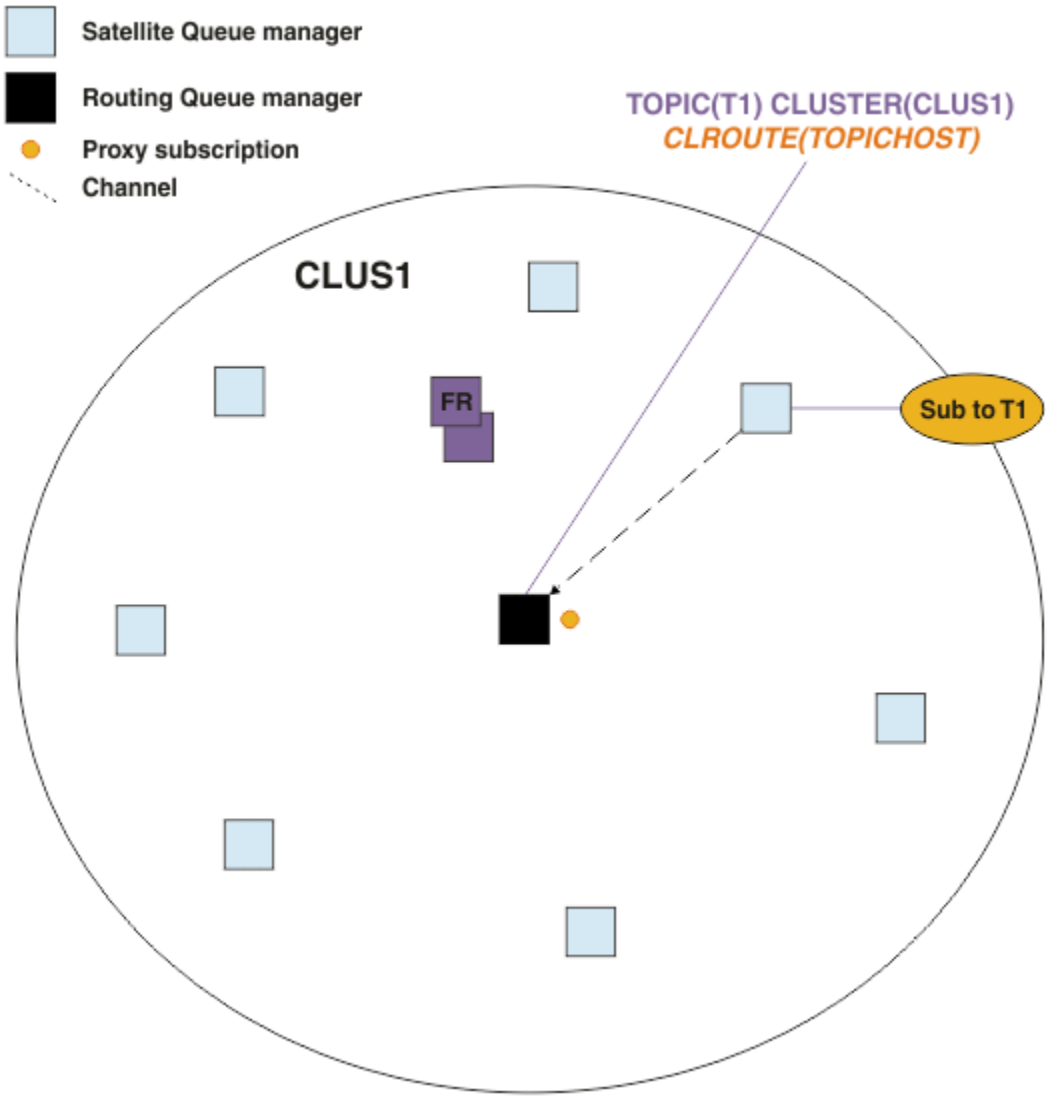


Abbildung 22. Ein Topic-Host-Publish/Subscribe-Cluster mit einem Thema, das auf einem Topic-Host definiert ist, und einen Subskribenten

Wenn eine Veröffentlichungsanwendung eine Verbindung zu einem anderen Warteschlangenmanager herstellt und eine Nachricht veröffentlicht wird, wird ein Kanal zwischen dem Veröffentlichungswarteschlangenmanager und dem Topic-Host-Warteschlangenmanager erstellt, und die Nachricht wird an diesen Warteschlangenmanager weitergeleitet. Der Veröffentlichungswarteschlangenmanager hat keine Kenntnis von Subskriptionen auf anderen WS-Managern im Cluster, daher wird die Nachricht auch dann an den Topic-Host-Warteschlangenmanager weitergeleitet, wenn es keine Subskribenten zu diesem Thema im Cluster gibt. Der Veröffentlichungswarteschlangenmanager stellt eine Verbindung nur mit dem Topic-Host-Warteschlangenmanager her. Veröffentlichungen werden, falls vorhanden, über den Themenhost an die subskribierenden Warteschlangenmanager weitergeleitet.

Subskriptionen auf demselben Warteschlangenmanager wie der Bereitsteller werden direkt erfüllt, ohne dass die Nachrichten zuerst an einen Topic-Host-Warteschlangenmanager gesendet werden.

Beachten Sie, dass Sie aufgrund der kritischen Rolle, die jeder Topic-Host-Warteschlangenmanager gespielt hat, Warteschlangenmanager auswählen müssen, die die Anforderungen zum Laden, Verfügbarkeits-und Konnektivitätsanforderungen des Topic-Hosts verarbeiten können.

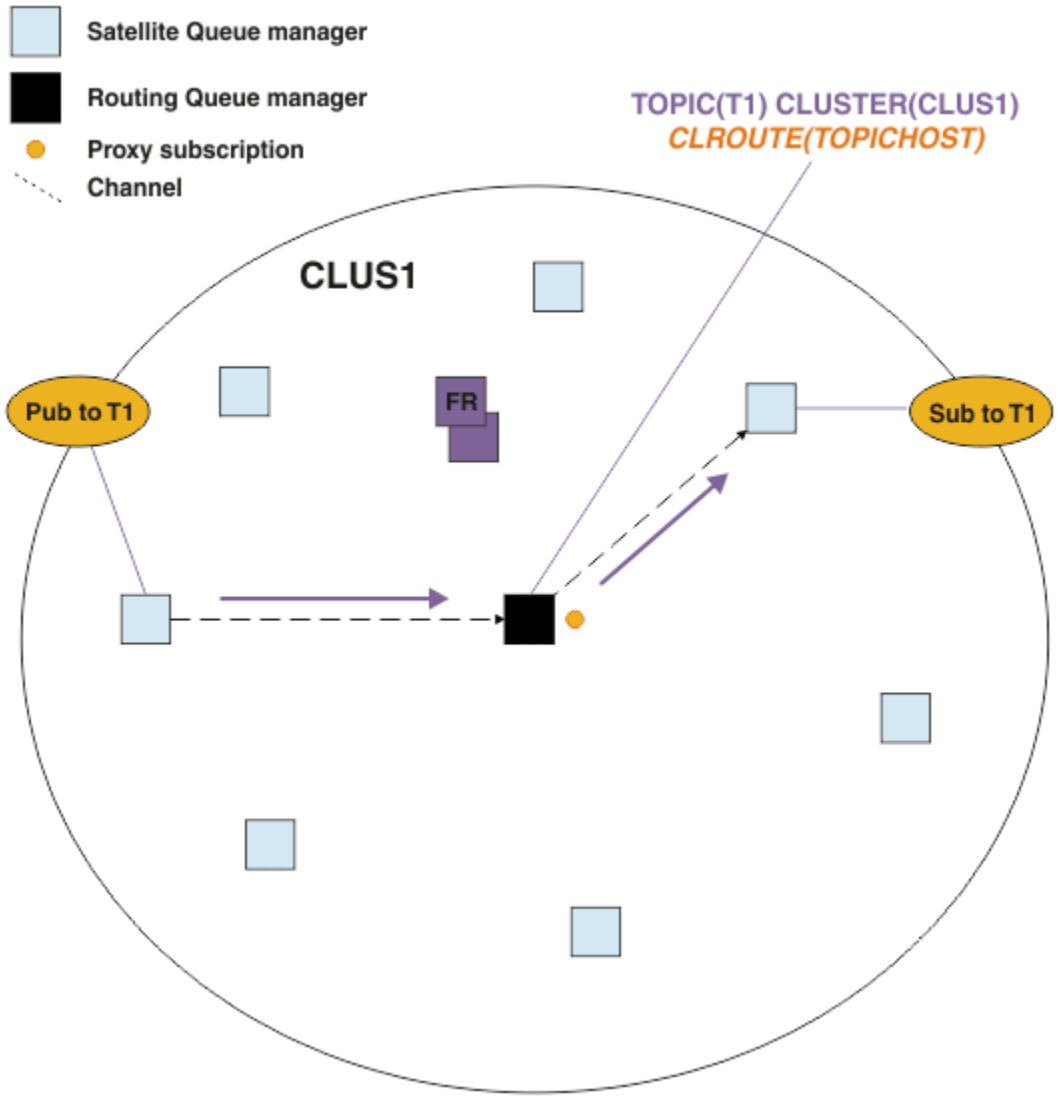


Abbildung 23. Ein Topic-Host-Publish/Subscribe-Cluster mit einem Topic, einem Subskribenten und einem Publisher

Unterteilung der Themenstruktur auf mehrere Warteschlangenmanager

Ein weitergeleitete Topic-Host-Warteschlangenmanager ist nur für die Subskriptionswissens-und Veröffentlichungsnachrichten zuständig, die sich auf die Verzweigung der Themenstruktur beziehen, für die das verwaltete Themenobjekt konfiguriert ist. Wenn verschiedene Publish/Subscribe-Anwendungen im Cluster verwendet werden, können Sie verschiedene WS-Manager für die verschiedenen Clusterverzweigungen der Themenstruktur konfigurieren. Dies ermöglicht die Skalierung, indem der Veröffentlichungsdatenverkehr, die Subskriptionskenntnisse und die Kanäle auf den einzelnen Topic-Host-WS-Managern im Cluster reduziert werden. Sie sollten diese Methode für unterschiedliche Bereiche mit hohem Datenvolumen verwenden, die in der Themenstruktur enthalten sind:

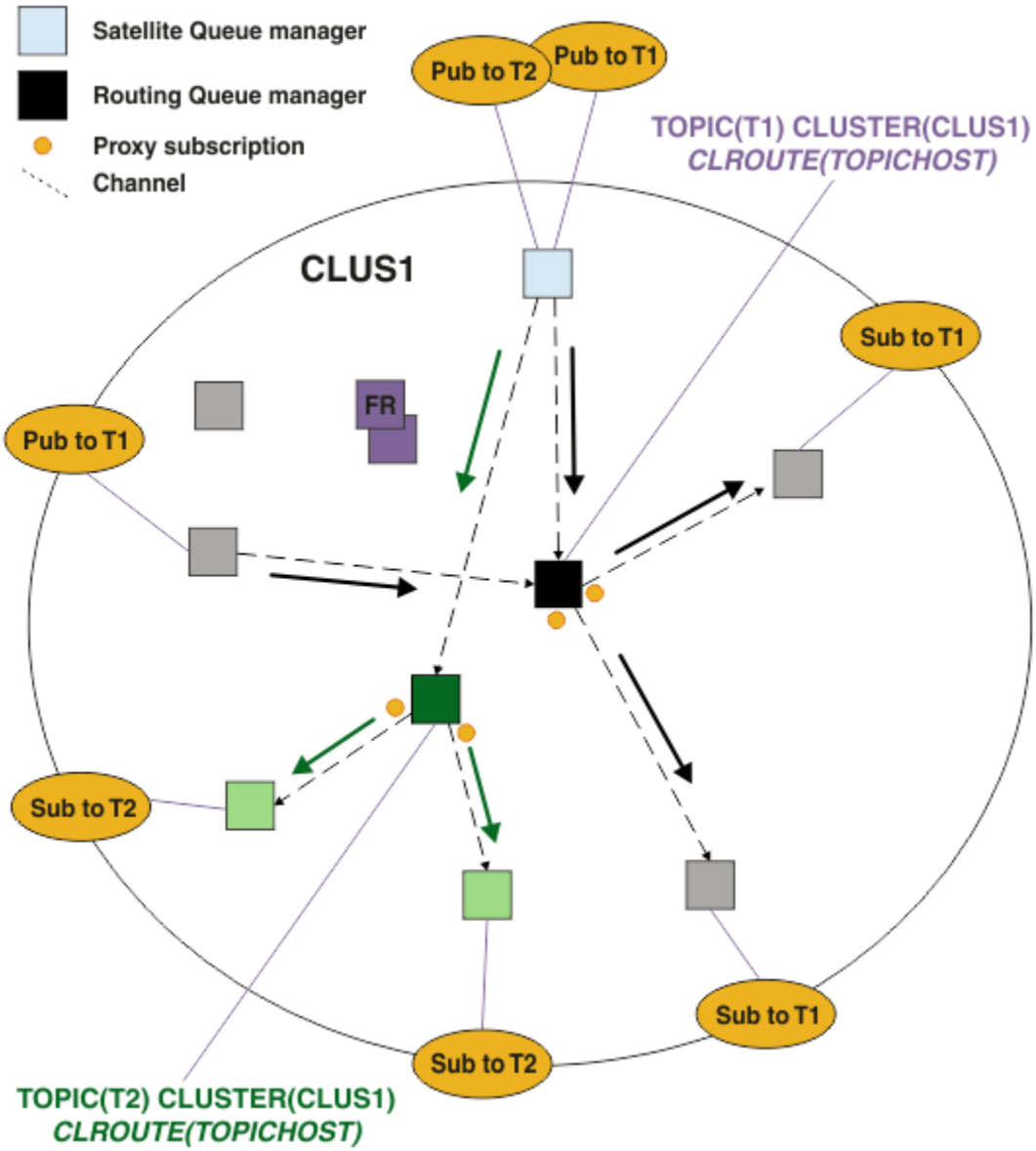


Abbildung 24. Ein Topic-Host-Publish/Subscribe-Cluster mit zwei Themen, die jeweils auf einem Themen-host definiert sind

Wenn Sie beispielsweise die in Themenstrukturen beschriebenen Themen verwenden und das Thema T1 mit der Themenzeichenfolge `/USA/Alabama` und das Thema T2 mit der Themenzeichenfolge `/USA/Alaska` konfiguriert wurde, wird eine in `/USA/Alabama/Mobile` veröffentlichte Nachricht über den Warteschlangenmanager, auf dem sich T1 befindet, und eine in `/USA/Alaska/Juneau` veröffentlichte Nachricht über den Warteschlangenmanager, auf dem sich T2 befindet, weitergeleitet.

Anmerkung: Sie können keine einzelne Subskription über mehrere Clusterverzweigungen der Themenstruktur erstellen, indem Sie in der Themenstruktur ein höheres Platzhalterzeichen verwenden als die Punkte, die in Gruppen zusammengefasst sind. Siehe Platzhalter für Platzhalterzeichen .

Thema Hostweiterleitung unter Verwendung mehrerer Topic-Hosts für ein einzelnes Thema

Wenn ein einzelner Warteschlangenmanager die Verantwortung für die Weiterleitung eines Themas hat und dass der Warteschlangenmanager nicht mehr verfügbar oder nicht in der Lage ist, die Auslastung zu verarbeiten, werden die Veröffentlichungen nicht zeitnah an die Subskriptionen weitergeleitet.

Wenn Sie mehr Ausfallsicherheit, Skalierbarkeit und Lastausgleich benötigen, als Sie bei der Definition eines Themas in nur einem Warteschlangenmanager ein Thema definieren, können Sie ein Thema in mehr als einem Warteschlangenmanager definieren. Jede einzelne veröffentlichte Nachricht wird über einen einzigen Themenhost weitergeleitet. Wenn mehrere übereinstimmende Topic-Hostdefinitionen vorhanden sind, wird einer der Themenhosts ausgewählt. Die Auswahl erfolgt auf die gleiche Weise wie für Clusterwarteschlangen. Dadurch können Nachrichten an verfügbare Topic-Hosts weitergeleitet werden, so dass keine vorhanden sind, die nicht verfügbar sind, und es ermöglicht, dass die Nachrichtenlast auf mehrere Topic-Host-WS-Manager und -Kanäle verteilt wird. Wenn Sie mehrere Topic-Hosts für dasselbe Thema im Cluster verwenden, wird die Sortierung über mehrere Nachrichten jedoch nicht aufrechterhalten.

Das folgende Diagramm zeigt einen Topic-Host-Routing-Cluster, in dem das gleiche Thema auf zwei Warteschlangenmanagern definiert wurde. In diesem Beispiel senden die subskribierenden Warteschlangenmanager Informationen zu dem subskribierten Thema an beide Topic-Host-Warteschlangenmanager in Form einer Proxy-Subskription:

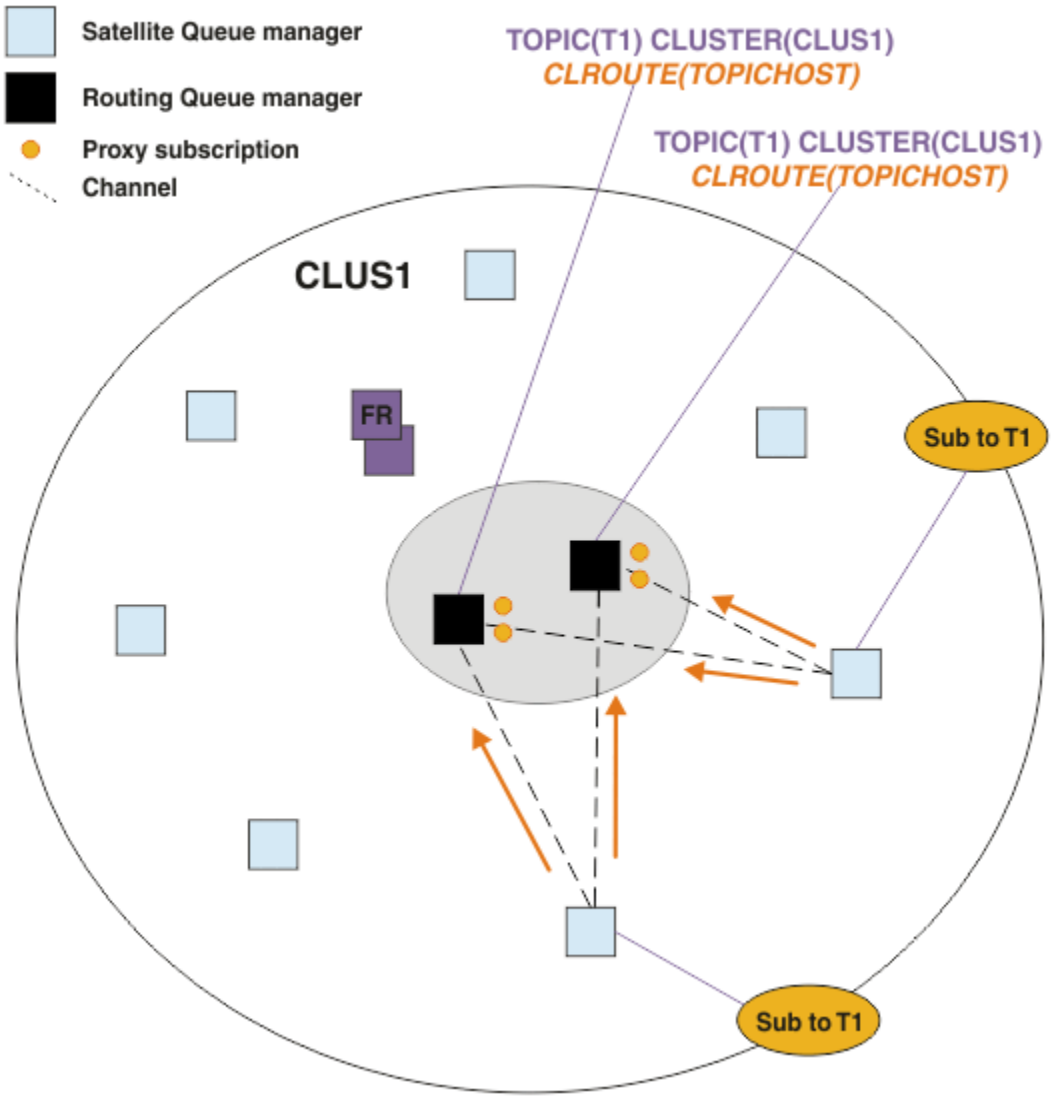


Abbildung 25. Proxy-Subskriptionen in einem Publish/Subscribe-Cluster mit mehreren Topics erstellen

Wenn eine Veröffentlichung von einem Nicht-Hosting-Warteschlangenmanager erstellt wird, sendet der Warteschlangenmanager eine Kopie der Veröffentlichung an *einen* des Topic-Host-WS-Managers für dieses Thema. Das System wählt den Host basierend auf dem Standardverhalten des Algorithmus für die Clusterauslastungsverwaltung aus. In einem typischen System nähert sich dies einer Round-Robin-Verteilung über die einzelnen Themenhostwarteschlangenmanager an. Es gibt keine Affinität zwischen

Nachrichten aus derselben Veröffentlichungsanwendung. Dies entspricht der Verwendung eines Clusterbindungs-Typs `NOTFIXED` .

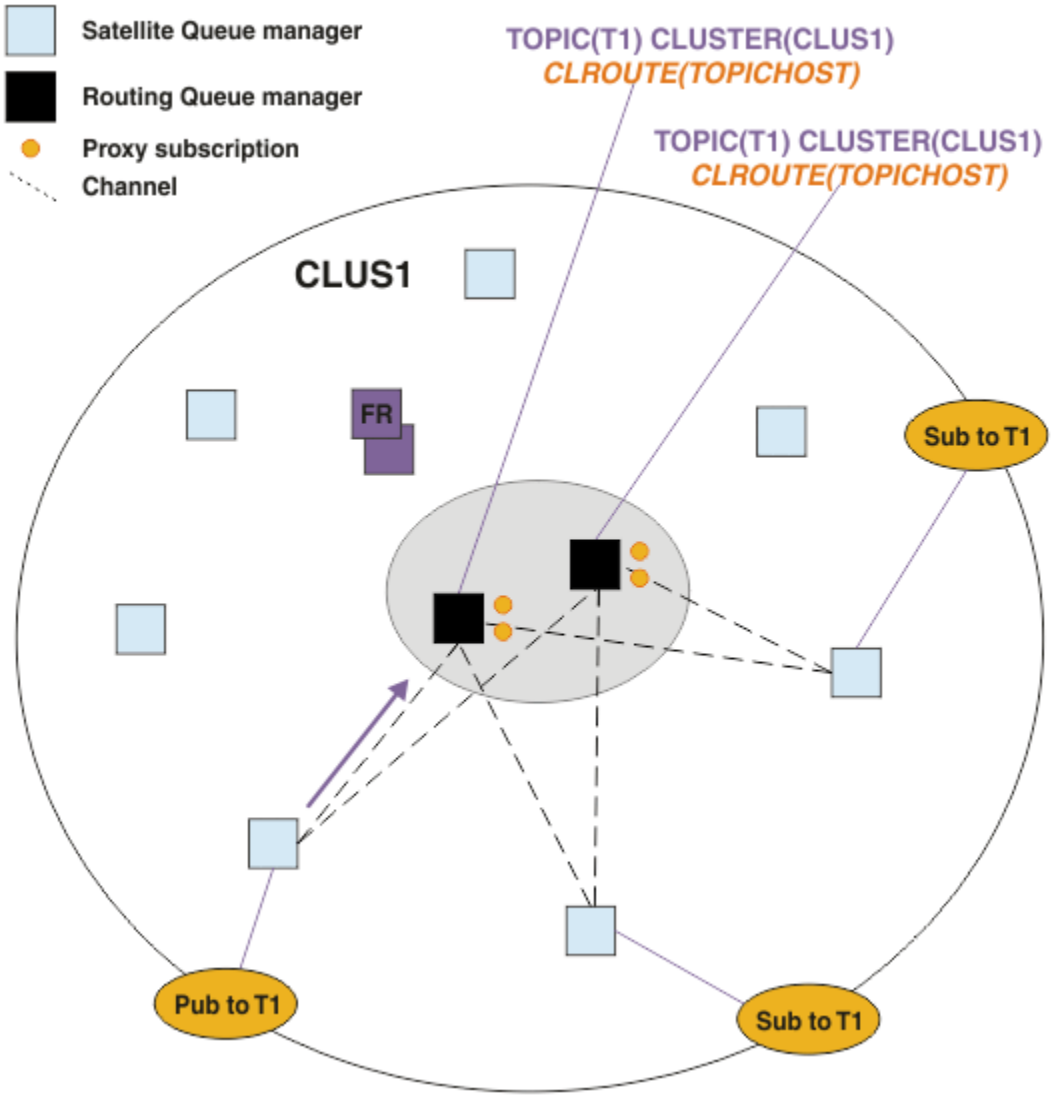


Abbildung 26. Empfangen von Veröffentlichungen in einem Publish/Subscribe-Cluster mit mehreren Themenhost

Inbound-Veröffentlichungen zum ausgewählten Topic-Host-Warteschlangenmanager werden dann an alle WS-Manager weitergeleitet, die eine übereinstimmende Proxy-Subskription registriert haben:

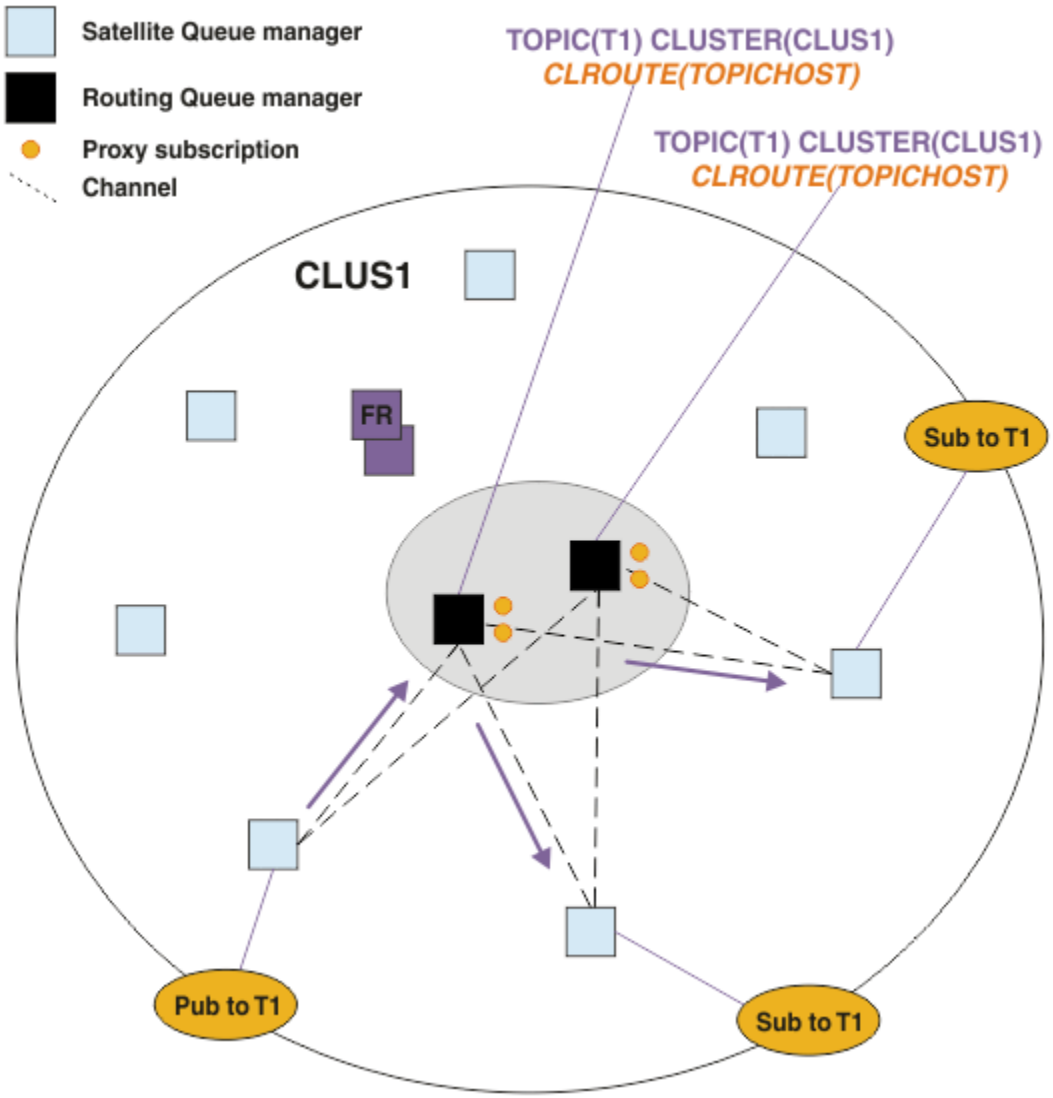


Abbildung 27. Weiterleiten von Veröffentlichungen an Subskribenten in einem Publish/Subscribe-Cluster mit mehreren Themen

Subskriptionen und Publiziern lokal für einen Topic-Host-Warteschlangenmanager erstellen

Die oben genannten Beispiele zeigen das Routing zwischen Publishern und Subskribenten auf WS-Managern, die keine verwalteten Themenobjekte im Host verwalten. In diesen Topologien benötigen Nachrichten mehrere *Hops* , um die Subskriptionen zu erreichen.

Wenn der zusätzliche Hop nicht erwünscht ist, kann es sinnvoll sein, wichtige Publisher mit Topic-Hosting-Warteschlangenmanagern zu verbinden. Wenn es jedoch mehrere Themenhosts für ein Thema und nur einen Bereitsteller gibt, wird der gesamte Veröffentlichungsdatenverkehr über den Topic-Host-Warteschlangenmanager weitergeleitet, mit dem der Bereitsteller verbunden ist.

In ähnlicher Weise, wenn es Schlüsselsubskriptionen gibt, können diese sich auf einem Topic-Host-Warteschlangenmanager befinden. Wenn es jedoch mehrere Hosts des weitergeleiteten Themas gibt, wird der zusätzliche Hop nur durch einen Teil der Veröffentlichungen vermieden, wobei der Rest zuerst durch die anderen Topic-Host-Warteschlangenmanager weitergeleitet wird.

Topologien wie diese werden hier weiter beschrieben: Topic-Host-Routing mit zentralisierten Publishern oder Subskribenten .

Anmerkung: Eine spezielle Planung ist erforderlich, wenn die Konfiguration des weitergeleiteten Topics geändert wird, wenn Publisher oder Subskriptionen mit Routing-Topic-Hosts gemeinsam lokalisiert werden. Weitere Informationen finden Sie im Abschnitt Einem Topic-Host-Cluster zusätzliche Topic-Hosts hinzufügen .

Zusammenfassung und weitere Hinweise

Ein Topic-Host-Publish/Subscribe-Cluster gibt Ihnen präzise Kontrolle darüber, welche Warteschlangenmanager die einzelnen Themen enthalten, und diese Warteschlangenmanager werden zu den *Routing* -Warteschlangenmanagern für diese Verzweigung in der Themenstruktur. Außerdem müssen Warteschlangenmanager ohne Subskriptionen oder Publisher keine Verbindung zu den Topic-Host-Warteschlangenmanagern herstellen, und Warteschlangenmanager mit Subskriptionen müssen keine Verbindung zu Warteschlangenmanagern herstellen, die kein Thema enthalten. Diese Konfiguration kann die Anzahl der Verbindungen zwischen WS-Managern im Cluster und die Menge der Informationen, die zwischen WS-Managern übertragen werden, erheblich reduzieren. Dies gilt insbesondere für große Cluster, in denen nur eine Untergruppe von Warteschlangenmanagern Publish/Subscribe-Arbeiten ausführt. Diese Konfiguration gibt Ihnen auch die Möglichkeit, die Auslastung einzelner Warteschlangenmanager im Cluster zu steuern, so dass Sie z. B. hochaktive Themen auf leistungsfähigeren und ausfallsicheren Systemen hosten können. Für bestimmte Konfigurationen-insbesondere größere Cluster-ist es in der Regel eine besser geeignete Topologie als *direktes Routing* .

Beim TOPICHOST-Routing gelten für Ihr System jedoch bestimmte Einschränkungen:

- Die Systemkonfiguration und -wartung muss sorgfältiger geplant werden als dies beim DIRECT-Routing erforderlich ist. Sie müssen in der Themenstruktur die Punkte festlegen, die zu Clustern zusammengeschlossen werden sollen; ebenso müssen Sie festlegen, wo im Cluster sich die Themendefinitionen befinden.
- Wenn ein neues Thema mit TOPICHOST-Routing definiert wird, werden die Informationen wie bei Themen, für die DIRECT-Routing definiert ist, an die Warteschlangenmanager mit vollständigem Repository und von dort direkt an alle Clustermitglieder übertragen. Dadurch werden von den vollständigen Repositorys aus Kanäle zu jedem Clustermitglied gestartet (sofern dies noch nicht geschehen ist).
- Veröffentlichungen werden immer von einem Warteschlangenmanager, bei dem es sich nicht um einen Themenhost handelt, an einen Warteschlangenmanager, der als Themenhost dient, gesendet; dies ist auch dann der Fall, wenn im Cluster keine Subskriptionen vorhanden sind. Wenn daher davon ausgegangen werden kann, dass Subskriptionen vorliegen, oder wenn der Aufwand für die globale Konnektivität und für globale Informationen höher ist als der eventuell zusätzliche Datenverkehr beim Übertragen von Veröffentlichungen, sollten Sie Themen-Routing verwenden.

  Anmerkung: Wie bereits beschrieben, kann das Veröffentlichen von Publishern auf einem Topic-Host dieses Risiko mindern.
- Nachrichten, die auf Warteschlangenmanagern veröffentlicht werden, die keine Hosts sind, gehen nicht direkt an den Warteschlangen, der die Subskription hostet, sondern werden immer über einen TOPICHOST-Warteschlangenmanager weitergeleitet. Dadurch erhöhen sich der Gesamtaufwand für den Cluster und die Latenzzeit bei der Nachrichtenübertragung, wodurch sich die Leistung verschlechtert.

  Anmerkung: Wie bereits beschrieben, können Subskriptionen oder Publisher, die für einen Topic-Host lokal sind, dieses Risiko mindern.
- Die Verwendung nur eines Warteschlangenmanagers als Themenhost stellt einen SPoF (Single Point of Failure) für alle Nachrichten dar, die zu einem Thema veröffentlicht werden. Durch eine Definition mehrerer Themenhosts wird ein solcher SPoF ausgeschlossen. Bei Verwendung mehrerer Hosts ändert sich allerdings die Reihenfolge, in der veröffentlichte Nachrichten für Subskriptionen empfangen werden.
- TOPICHOST-Warteschlangenmanager erzeugen ein zusätzliches Nachrichtenvolumen, weil sie Veröffentlichungsdatenverkehr von mehreren Warteschlangenmanagern verarbeiten müssen. Dieses Volumen kann verringert werden, indem mehrere TOPICHOSTs für ein einzelnes Thema verwendet werden (wobei die Reihenfolge der Nachrichten nicht beibehalten wird) oder indem verschiedene Warteschlangenmanager als Hosts für weitergeleitete Themen für verschiedene Zweige der Themenstruktur verwendet werden.

Bevor Sie Topic-Host-Routing verwenden, untersuchen Sie die alternativen Ansätze in „Direktes Routing in Publish/Subscribe-Clustern” auf Seite 83 und „Routing in Publish/Subscribe-Hierarchien” auf Seite 114.

Publish/Subscribe-Clustering: Bewährte Verfahren

Durch die Verwendung von Clusterthemen wird die Publish/Subscribe-Domäne zwischen der Warteschlange erweitert. Manager einfach, kann aber zu Problemen führen, wenn die Mechanik und die Auswirkungen nicht vollständig sind verstanden. Es gibt zwei Modelle für die gemeinsame Nutzung von Informationen und das Routing von Veröffentlichungen. Implementieren Sie das Modell, das Ihren individuellen Geschäftsanforderungen am besten entspricht, und führt Sie am besten für Ihre Auswahl aus. Cluster.

Die Best-Practice-Informationen in den folgenden Abschnitten bieten nicht eine Gesamtgröße für alle Lösungen an, sondern gemeinsame Ansätze zur Lösung allgemeiner Probleme. Es wird vorausgesetzt, dass Sie ein grundlegendes Verständnis von IBM MQ-Clustern und von Publish/Subscribe-Nachrichtenübermittlung haben und dass Sie mit den Informationen aus Verteilte Publish/Subscribe-Netze und „Publish/Subscribe-Cluster entwerfen” auf Seite 81 vertraut sind.

Wenn Sie einen Cluster für Punkt-zu-Punkt-Messaging verwenden, arbeitet jeder WS-Manager im Cluster auf einer Basis-zu-Know-Basis. Dies bedeutet, dass es nur Informationen zu anderen Clusterressourcen, wie z. B. anderen WS-Managern im Cluster und in Clusterwarteschlangen, findet, wenn Anwendungen, die eine Verbindung zu ihnen herstellen, sie verwenden möchten. Wenn Sie Publish/Subscribe-Messaging zu einem Cluster hinzufügen, wird ein erhöhtes Maß an Informationsaustausch und Konnektivität zwischen Cluster-WS-Managern eingeführt. Um die Best Practices für Publish/Subscribe-Cluster verfolgen zu können, müssen Sie die Auswirkungen dieser Änderung im Verhalten umfassend verstehen.

Um Ihnen die beste Architektur zu ermöglichen, basierend auf Ihren präzisen Bedürfnissen, gibt es zwei Modelle Informationen zur gemeinsamen Nutzung und Veröffentlichung von Veröffentlichungen in Publish/Subscribe-Clustern: *direktes Routing* und *Topic-Host-Routing* . Um das richtige zu machen Auswahl, müssen beide Modelle und die unterschiedlichen Anforderungen, die jeweils Modell genügt. Diese Anforderungen werden in den folgenden Abschnitten erläutert: Verbindung mit „Verteiltes Publish/Subscribe-Netz planen” auf Seite 77:

- „Gründe für die Begrenzung der Anzahl der Cluster-WS-Manager, die in Publish/Subscribe-Aktivität” auf Seite 98
- „Vorgehensweise bei der Entscheidung, welche Themen in einem Cluster” auf Seite 99
- „Wie Sie Ihr System als Größe” auf Seite 100
- „Bereitsteller-und Subskriptionsposition” auf Seite 101
- „Veröffentlichungsdatenverkehr” auf Seite 101
- „Subskriptionsänderung und dynamische Themenzeichenfolgen” auf Seite 102

Gründe für die Begrenzung der Anzahl der Cluster-WS-Manager, die in Publish/Subscribe-Aktivität

Wenn Sie Publish/Subscribe-Messaging in einem Cluster verwenden, sind Kapazitätserwägungen und Leistungsaspekte zu beachten. Daher ist es die beste Praxis, die die Publish/Subscribe-Aktivität über Warteschlangenmanager hinweg erforderlich ist, und sie nur auf die Anzahl der Warteschlangenmanager, die diese erfordern. Nach der Mindestanzahl an Warteschlangen Manager, die Themen veröffentlichen und subskribieren müssen, werden identifiziert. Sie können Mitglieder eines Clusters sein, der nur sie enthält, und keine anderen Warteschlangenmanager.

Dieser Ansatz ist besonders nützlich, wenn Sie bereits einen etablierten Cluster haben. funktionieren gut für Punkt-zu-Punkt-Messaging. Wenn Sie eine vorhandene Großer Cluster in einem Publish/Subscribe-Cluster. Es ist eine bessere Methode, zunächst einen separaten Cluster für die Publish/Subscribe-Arbeit zu erstellen, in dem die Anwendungen nicht mit dem aktuellen Cluster versucht werden. Sie können eine Teilmenge der vorhandenen WS-Manager, die sich bereits in einem oder mehreren Punkt-zu-Punkt-Clustern befinden, und Diese Teilmenge der Member des neuen Publish/Subscribe-Clusters. Allerdings ist

die vollständige Repository-WS-Manager für Ihren neuen Cluster dürfen keine Member eines anderen Clusters sein. Dies isoliert die zusätzliche Last aus dem vorhandenen Cluster. Repositorys.

Wenn Sie keinen neuen Cluster erstellen können, müssen Sie einen vorhandenen großen Cluster in einen Publish/Subscribe-Cluster, verwenden Sie kein direktes Routing-Modell. Der Topic-Host, der weitergeleitet -Modell wird in der Regel besser in größeren Clustern ausgeführt, da es im Allgemeinen die Publish/Subscribe-gemeinsame Nutzung von Informationen und Konnektivität mit der Gruppe von Warteschlangenmanagern die aktiv Publish/Subscribe-Arbeiten ausführen und sich auf die Warteschlange konzentrieren Manager, die die Themen enthalten. Die Ausnahme ist die, wenn eine manuelle Aktualisierung der Subskriptionsinformationen werden auf einem Warteschlangenmanager aufgerufen, auf dem eine Themendefinition gehostet wird. An diesem Punkt wird der Topic-Host-Warteschlangenmanager mit jedem WS-Manager in Verbindung der Cluster. Lesen Sie hierzu den Abschnitt Resynchronisation von Proxy-Subskriptionen.

Wenn Sie feststellen, dass ein Cluster aufgrund seiner Größe nicht für Publish/Subscribe verwendet werden kann oder die aktuelle Last, es ist eine gute Methode, um zu verhindern, dass dieser Cluster unerwartet ausgeführt wird. in einen Publish/Subscribe-Cluster. WS-Manager von **PSCLUS** verwenden Eigenschaft zum Stoppen von jedem, der ein Clusterthema in einem WS-Manager in der Cluster. Siehe „Clusterveröffentlichungs-/Subskriptionssubskribieren” auf Seite 109.

Vorgehensweise bei der Entscheidung, welche Themen in einem Cluster

Es ist wichtig, sorgfältig auszuwählen, welche Themen dem Cluster hinzugefügt werden: Je höher die Themenstruktur in diesen Themen ist, desto breiter wird ihre Verwendung. Dies kann Sie führen dazu, dass mehr Subskriptionsinformationen und Veröffentlichungen weitergegeben werden, als dies erforderlich ist. Wenn es mehrere, eindeutige Zweige der Themenstruktur gibt, wo einige müssen in Gruppen zusammengefasst sein, andere nicht, erstellen Sie verwaltete Themenobjekte im Stammverzeichnis. von jeder Verzweigung, die Clustering benötigt, und die dem Cluster hinzufügen. Wenn z. B. Verzweigungen `/A`, `/B` und `/C` benötigen Clustering, definieren eine separate Clusterthemenobjekte für jede Verzweigung.

Anmerkung: Die Das System verhindert das Verschachtzen von Clusterthemendefinitionen in der Themenstruktur. Sie dürfen nur Topics an einem Punkt in der Themenstruktur für die einzelnen Unterverzweigung. So können Sie beispielsweise keine Clusterthemenobjekte für `/A` und für `/A/B` definieren. Die Verschachtelung von Clusterthemen kann zu Verwechslungen dabei führen, welches Clusterobjekt für welches Abonnement gilt, insbesondere wenn Abonnements Wildcards verwenden. Das ist noch wichtiger, bei Verwendung von Topic-Host-Routing, bei dem Routing-Entscheidungen genau definiert sind durch Ihre Zuordnung von Themenhosts.

Wenn Clusterthemen hoch in der Themenstruktur hinzugefügt werden müssen, aber einige Verzweigungen der Baum unter dem Clusterpunkt erfordert nicht das Clustering-Verhalten, das Sie verwenden können. die Attribute der Subskriptions-und Veröffentlichungsbereiche, um die Ebene von Abonnements-und Publikationsfreigabe für weitere Themen.

Sie sollten den Topic-Root-Knoten nicht in den Cluster stellen, ohne die Verhalten, das angezeigt wird. Globale Themen möglichst naheliegend machen, z. B. durch mit einem übergeordneten Qualifikationsmerkmal in der Themenzeichenfolge: `/global` oder `/cluster`.

Es gibt einen weiteren Grund, warum der Root-Topic-Knoten nicht in einem Cluster zusammengefasst werden soll. Dies liegt daran, dass jeder Warteschlangenmanager über eine lokale Definition für den Stammknoten, das Themenobjekt `SYSTEM.BASE.TOPIC` , verfügt. Wenn dieses Objekt in Gruppen zusammengefasst ist Auf einem WS-Manager im Cluster werden alle anderen WS-Manager auf diese Warteschlangenmanager aufmerksam gemacht. Wenn jedoch eine lokale Definition desselben Objekts vorhanden ist, wird die zugehörige Eigenschaft überschrieben. das Clusterobjekt. Dies führt dazu, dass diese WS-Manager so handeln, als wäre das Thema nicht gruppiert. Um dies zu beheben, müssten Sie jede Definition von `SYSTEM.BASE.TOPIC`in einem Cluster zusammenfassen. Sie könnten dies für direkte Weiterleitung tun. Definitionen, aber nicht für Topic-Host-Routing-Definitionen, da sie alle WS-Manager zu einem Themenhost.

Wie Sie Ihr System als Größe

Publish/Subscribe-Cluster führen in der Regel zu einem anderen Cluster-Muster. Kanäle zu Punkt-zu-Punkt-Messaging in einem Cluster. Das Punkt-zu-Punkt-Modell ist ein 'opt in' eine, aber Publish/Subscribe-Cluster haben eine mehr wahllose Natur mit Abonnement-Fan-out, insbesondere bei Verwendung von direkt weitergeleiteten Themen. Daher ist es Wichtige Angabe, welche WS-Manager in einem Publish/Subscribe-Cluster verwendet werden sollen Clusterkanäle, um eine Verbindung zu anderen Warteschlangenmanagern herzustellen und unter welchen Umständen.

In der folgenden Tabelle ist die typische Gruppe der Clustersender-und -Empfängerkanäle aufgeführt. für jeden WS-Manager in einem Publish/Subscribe-Cluster unter normaler Ausführung erwartet wird, in Abhängigkeit von der Rolle des WS-Managers im Publish/Subscribe-Cluster.

Tabelle 5. Clustersender-und Empfängerkanäle für jede Routing-Methode.

WS-Manager-Rolle	Direkte Clusterempfänger	Direkte Clustersender	Topic-Clusterempfänger	Topic-Clustersender
Vollständiger Repository	AllQmgrs	AllQmgrs	AllQmgrs	AllQMgrs
Host der Themendefinition	nicht zutreffend	nicht zutreffend	AllSubs + AllPubs (1)	Alle Subs (1)
Subskriptionen erstellt	AllPubs (1)	AllQMgrs	AllHosts	AllHosts
Bereitgeschaltete Veröffentlichungskomponenten	Alle Subs (1)	Alle Subs (1)	AllHosts	AllHosts
Keine Publisher oder Subskribenten	Alle Subs (1)	Keiner (1)	Keiner (2)	Keiner (2)

Schlüssel:

AllQmgrs
Ein Kanal zu und von jedem WS-Manager im Cluster.

AllSubs
Ein Kanal zu und von jedem WS-Manager, in dem eine Subskription erstellt.

AllPubs
Ein Kanal zu und von jedem WS-Manager, an den eine Veröffentlichungsanwendung angeschlossen wurde.

AllHosts
Ein Kanal zu und von jedem Warteschlangenmanager, in dem eine Definition des s Clusterthemenobjekt wurde konfiguriert.

--
Keine Kanäle zu oder von anderen Warteschlangenmanagern im Cluster für die einzige Zweck des Publish/Subscribe-Messaging.

Anmerkungen:

1. Wenn eine WS-Manageraktualisierung von Proxy-Subskriptionen aus dieser Warteschlange erstellt wird Manager, ein Kanal zu und von allen anderen WS-Managern im Cluster automatisch erstellt werden.
2. Wenn eine WS-Manageraktualisierung von Proxy-Subskriptionen aus dieser Warteschlange erstellt wird Manager, einen Kanal zu und von allen anderen WS-Managern im Cluster, die host kann eine Definition eines Clusterthemas automatisch erstellt werden.

In der vorherigen Tabelle wird gezeigt, dass das Thema Host Routing in der Regel deutlich weniger verwendet. Clustersender-und Empfängerkanäle als direktes Routing. Wenn die Kanalkonnektivität ein

Problem für bestimmte Warteschlangenmanager in einem Cluster, aus Gründen der Kapazität oder der Fähigkeit, bestimmte Kanäle einzurichten (z. B. über Firewalls), Topic-Host Die Weiterleitung ist daher eine bevorzugte Lösung.

Bereitsteller-und Subskriptionsposition

Ein Cluster-Publish/Subscribe aktiviert Nachrichten, die auf einem WS-Manager veröffentlicht werden. an Subskriptionen für alle anderen WS-Manager im Cluster zugestellt werden. Wie für Punkt-zu-Punkt-Messaging, die Kosten für die Übertragung von Nachrichten zwischen Warteschlangenmanagern kann sich negativ auf die Leistung auswirken. Daher sollten Sie die Erstellung von Subskriptionen für Themen in denselben Warteschlangenmanagern, in denen Nachrichten angezeigt werden veröffentlicht.

Bei der Verwendung von Topic-Host-Routing in einem Cluster ist es wichtig, dass auch die Position der Subskriptionen und Publisher in Bezug auf die Topic-Hosting-Warteschlange Manager. Wenn der Publisher nicht mit einem Warteschlangenmanager verbunden ist, der ein Host von ist Das Clusterthema Nachrichten, die veröffentlicht werden, werden immer an einen Topic-Hosting-Warteschlangenmanager gesendet. Gleichermaßen, wenn eine Subskription auf einem Warteschlangenmanager erstellt wird, der nicht Topic-Host für ein Clusterthema, Nachrichten, die von anderen WS-Managern in veröffentlicht werden Der Cluster wird immer zuerst an einen Topic-Host-Warteschlangenmanager gesendet. Mehr wenn sich die Subskription auf einem WS-Manager befindet, auf dem sich die -Thema, aber es gibt einen oder mehrere andere Warteschlangenmanager, die ebenfalls dieses Thema hosten, Ein Teil der Veröffentlichungen von anderen Warteschlangenmanagern wird durch diese weitergeleitet. andere Topic-Hosting-WS-Manager. Weitere Informationen finden Sie unter Topic-Host-Routing mit zentralen Publishern oder Subskribenten Informationen zum Entwerfen eines Topic-Host-Publish/Subscribe-Clusters zum Minimieren Abstand zwischen veröffentlichenden Stellen und Abonnements.

Veröffentlichungsdatenverkehr

Nachrichten, die von einer Anwendung veröffentlicht werden, die mit einem Warteschlangenmanager in einem Cluster verbunden ist Übertragung an Subskriptionen auf anderen WS-Managern mit Clustersender Kanäle.

Wenn Sie direktes Routing verwenden, nehmen die veröffentlichten Nachrichten den kürzesten Pfad zwischen WS-Manager. Das heißt, sie gehen direkt vom Veröffentlichungswarteschlangenmanager zu jedem der folgenden die Warteschlangenmanager mit Subskriptionen. Nachrichten werden nicht an die Warteschlange übertragen Manager, die keine Subskriptionen für das Thema haben. Siehe Proxy-Abonnements in einem Publish/Subscribe-Netz.

Gibt die Rate der Veröffentlichungsnachrichten zwischen einem Warteschlangenmanager und einem anderen in Der Cluster ist hoch, die Cluster-Channel-Infrastruktur zwischen diesen beiden Punkten. muss in der Lage sein, die Rate beizubehalten. Dies kann die Optimierung der Kanäle und Übertragungswarteschlange wird verwendet.

Wenn Sie Topic-Host-Routing verwenden, wird jede Nachricht, die in einem Warteschlangenmanager veröffentlicht wird, Es wird kein Themenhost an einen Topic-Host-WS-Manager übertragen. Dies ist unabhängig Gibt an, ob eine oder mehrere Subskriptionen an einer anderen Stelle im Cluster vorhanden sind. Dies enthält weitere Faktoren, die bei der Planung berücksichtigt werden müssen:

- Ist die zusätzliche Latenzzeit des ersten Sendens jeder Veröffentlichung an einen Topic-Host WS-Manager akzeptabel?
- Kann jeder Topic-Host-Warteschlangenmanager die eingehende und abgehende Veröffentlichung unterstützen rate? Betrachten Sie ein System mit Publishern auf vielen verschiedenen WS-Managern. Wenn sie senden ihre Nachrichten an eine sehr kleine Gruppe von Topic-Hosting-Warteschlangen Manager, werden diese Themenhosts zu einem Engpass bei der Verarbeitung dieser -Nachrichten und Routing-Nachrichten an die Subskribements von Warteschlangenmanagern.
- Erwartet wird, dass ein erheblicher Teil der veröffentlichten Nachrichten nicht einen übereinstimmenden Subskribenten haben? Ist dies der Fall, und ist die Veröffentlichungsrate dieser Nachrichten hoch ist, kann es am besten sein, den Warteschlangenmanager des Publishers zu einem Topic-Host zu

machen. In Eine veröffentlichte Nachricht, in der keine Subskriptionen im Cluster vorhanden sind, wird nicht an andere WS-Manager übertragen.

Diese Probleme können auch durch die Einführung mehrerer Themenhosts gelockert werden, um die Publikationslast auf sie:

- Wenn es mehrere unterschiedliche Themen gibt, die jeweils einen Teil der Veröffentlichungsdatenverkehr in Betracht ziehen, in Betracht ziehen, sie auf verschiedenen Warteschlangenmanagern
- Wenn die Themen nicht auf verschiedene Themenhosts getrennt werden können, sollten Sie Definieren desselben Themenobjekts auf mehreren Warteschlangenmanagern. Dies führt zu -Veröffentlichungen, die für die Weiterleitung auf die einzelnen von ihnen verteilt werden. Dies ist jedoch nur dann sinnvoll, wenn die Reihenfolge der Publizistnachrichten nicht sortiert ist. erforderlich.

Subskriptionsänderung und dynamische Themenzeichenfolgen

Eine weitere Überlegung ist die Auswirkung auf die Leistung des Systems für die Weitergabe. Proxy-Subskriptionen. Gewöhnlich sendet ein Warteschlangenmanager eine Proxy-Subskriptionsnachricht. bei bestimmten anderen WS-Managern im Cluster, wenn die erste Subskription für einen Eine bestimmte Clusterthemenzeichenfolge (nicht nur ein konfiguriertes Themenobjekt) wird unter erstellt. dieser WS-Manager. In ähnlicher Weise wird eine Proxy-Abonnementlöschungsnachricht gesendet, wenn Die letzte Subskription für eine bestimmte Clusterthemenzeichenfolge wird gelöscht.

Für direktes Routing sendet jeder WS-Manager mit Subskriptionen diese Proxy-Server Subskriptionen für alle anderen WS-Manager im Cluster. Bei Topic-Host-Routing sendet jeder WS-Manager mit Subskriptionen nur die Proxy-Subskriptionen an jede WS-Manager, der eine Definition für dieses Clusterthema enthält. Daher wird mit Direktes Routing, die mehr Warteschlangenmanager, die sich im Cluster befinden, um so höher ist die Der Systemaufwand für die Verwaltung von Proxy-Subskriptionen in allen In der Erwägung, dass der Themenhost Routing, die Anzahl der Warteschlangenmanager im Cluster ist kein Faktor.

Bei beiden Routing-Modellen, wenn eine Publish/Subscribe-Lösung aus vielen eindeutigen Themenzeichenfolgen, die subskribiert werden, oder die Themen in einem WS-Manager im Cluster häufig subskribiert und nicht subskribiert sind, wird ein erheblicher Systemaufwand in diesem Warteschlangenmanager zu sehen ist, verursacht durch die ständige Generierung von Nachrichten, die die Verteilung und die Proxy-Subskriptionen löschen. Bei direkter Weiterleitung wird dies durch die Notwendigkeit verbunden, diese Nachrichten an jeden WS-Manager im Cluster zu senden.

Wenn die Änderungsrate der Subskriptionen zu hoch ist, um sie aufnehmen zu können, selbst in einem Topic-Host-Routing-System, siehe Subskriptionsleistung in Publish/Subscribe-Netzen für Informationen über Möglichkeiten zum Reduzieren des Overhead des Proxy-Abonnements

Clusterthemen definieren

Clusterthemen sind Verwaltungsthemen, für die das Attribut **cluster** definiert ist. Informationen zu Clusterthemen werden an alle Clustermitglieder übertragen und mit lokalen Themen zu warteschlangenmanagerübergreifenden Thementeilbereichen verbunden. Damit können Nachrichten, die auf einem Warteschlangenmanager zu einem Thema veröffentlicht werden, an die Subskriptionen anderer Warteschlangenmanager im Cluster übermittelt werden.

Wenn Sie ein Clusterthema für einen Warteschlangenmanager definieren, wird diese Clusterthemendefinition an die Warteschlangenmanager mit vollständigem Repository gesendet. Anschließend leiten die vollständigen Repositorys die Clusterthemendefinition an alle Warteschlangenmanager im Cluster weiter, sodass das Clusterthema für alle Bereitsteller und Subskribenten verfügbar ist, die in einem Warteschlangenmanager im Cluster vorhanden sind. Der Warteschlangenmanager, in dem ein Clusterthema erstellt wird, wird als Clusterthemenhost bezeichnet. Das Clusterthema kann von allen Warteschlangenmanagern im Cluster verwendet werden; alle Änderungen an einem Clusterthema müssen jedoch in dem Warteschlangenmanager vorgenommen werden, in dem das Thema definiert ist (d. h. im Clusterthemenhost); anschließend wird die Änderung über die vollständigen Repositorys an alle Clustermitglieder weitergegeben.

Wenn Sie direktes Routing verwenden, wird die Position der Clusterthemendefinition nicht wirkt sich direkt auf das Verhalten des Systems aus, da alle Warteschlangenmanager im Cluster Verwenden Sie die

Themendefinition auf die gleiche Weise. Daher sollten Sie das Thema in jedem beliebigen Warteschlangenmanager, der ein Mitglied des Clusters sein wird, solange das Thema benötigt wird, und das ist auf einem System zuverlässig genug, um regelmäßig in Kontakt mit der vollen Repository-WS-Manager.

Wenn Sie Topic-Host-Routing verwenden, ist die Position der Clusterthemendefinition sehr wichtig. wichtig, da andere WS-Manager im Cluster Kanäle zu dieser Warteschlange erstellen Verwalter und Abonnements-Informationen und Veröffentlichungen an sie senden. So wählen Sie die beste WS-Manager zum Hosten der Themendefinition, müssen Sie das Thema Host-Routing verstehen. Siehe „Thema Host-Routing in Publish/Subscribe-Clustern" auf Seite 88.

Wenn Sie über ein Clusterthema und ein lokales Themenobjekt verfügen, hat das lokale Thema Vorrang. Siehe „Mehrere Cluster-Topic-Definitionen mit demselben Namen" auf Seite 105.

Informationen zu den Befehlen, mit denen Clusterthemen angezeigt werden, finden Sie in den zugehörigen Informationen.

Vererbung von Clustern

In der Regel erwarten Veröffentlichungs-und Subskribierungsanwendungen in einer Publish/Subscribe-Clustertopologie die gleiche Arbeit, unabhängig davon, welcher WS-Manager im Cluster sie enthält. sind verbunden mit. Aus diesem Grund werden die verwalteten Themenobjekte in Clustern an die jeden WS-Manager im Cluster.

Ein verwaltungs-Topic-Objekt übernimmt sein Verhalten von einem anderen verwalteten Thema. Objekte höher in der Themenstruktur. Diese Vererbung tritt auf, wenn ein expliziter Wert wurde nicht für einen Themenparameter festgelegt.

Im Falle eines in Gruppen zusammengefassten Publish/Subscribe ist es wichtig, eine solche zu berücksichtigen. Vererbung, weil sie die Möglichkeit bietet, dass Publisher und Abonnenten verhält sich abhängig von dem Warteschlangenmanager, zu dem sie eine Verbindung herstellen, unterschiedlich. Wenn ein Das Clusterthemenobjekt hinterlässt Parameter, die von höheren Themenobjekten übernommen werden. Das Thema kann sich auf verschiedenen Warteschlangenmanagern im Cluster unterschiedlich verhalten. Ebenso werden lokal definierte Themenobjekte, die unter einem Clusterthemenobjekt definiert sind, in Die Themenstruktur bedeutet, dass die niedrigeren Themen noch in Gruppen zusammengefasst sind, aber die lokale -Objekt kann sein Verhalten in einer Weise ändern, die sich von anderen Warteschlangenmanagern unterscheidet. im Cluster.

Platzhaltersubskriptionen

Proxy-Subskriptionen werden erstellt, wenn lokale Subskriptionen an eine Themenzeichenfolge vorgenommen werden. der in einem Clusterthemenobjekt aufgelöst wird oder darunter ist. Bei einer Subskription mit Platzhalterzeichen höher in der Themenhierarchie als ein beliebnises Clusterthema erstellt hat, hat es keinen Proxy Subskriptionen, die um den Cluster für das übereinstimmende Clusterthema gesendet werden, und daher keine Veröffentlichungen von anderen Membern des Clusters empfängt. Sie erhält jedoch Veröffentlichungen aus dem lokalen WS-Manager.

Wenn eine andere Anwendung jedoch eine Themenzeichenfolge subskribiert, die in oder aufgelöst wird, unterhalb des Clusterthemas werden Proxy-Subskriptionen generiert und Veröffentlichungen an diesen WS-Manager weitergegeben werden. Bei Ankunft das ursprüngliche, höhere Wildcard Subskription wird als rechtmäßiger Empfänger dieser Veröffentlichungen betrachtet und erhält eine Kopie. Wenn dieses Verhalten nicht erforderlich ist, legen Sie **`WILDCARD(BLOCK)`** im Clusterthema fest. Dadurch wird das ursprüngliche Platzhalterzeichen nicht als legitimes Platzhalterzeichen betrachtet. -Subskription und stoppt das Empfangen von Veröffentlichungen (lokal oder an anderer Stelle in den Cluster) im Clusterthema oder in dessen Unterthemen.

Zugehörige Konzepte

Mit Verwaltungsthemen arbeiten

Mit Subskriptionen arbeiten

Zugehörige Verweise

ANZEIGETHEMA

ANZEIGETPSTATUS
ANZEIGEUNTERGEORDNET

Clusterthemenattribute
Wenn ein Themenobjekt das Attribut "Clustername" definiert hat, wird die Themendefinition auf alle Warteschlangenmanager im Cluster verteilt. Jeder WS-Manager verwendet die weitergegebenen Themenattribute, um das Verhalten von Publish/Subscribe-Anwendungen zu steuern.

Ein Themenobjekt verfügt über eine Reihe von Attributen, die für Publish/Subscribe-Cluster gelten. Einige steuern das allgemeine Verhalten der Veröffentlichungs-und Subskribierungsanwendungen und steuern, wie das Thema im gesamten Cluster verwendet wird.

Eine Clusterthemenobjektdefinition muss so konfiguriert werden, dass sie alle Warteschlangenmanager im Cluster korrekt verwenden kann.

Wenn beispielsweise die Modellwarteschlangen für verwaltete Subskriptionen (`MDURMDL` und `MNDURMDL`) auf einen nicht standardmäßigen Warteschlangennamen gesetzt werden, muss diese benannte Modellwarteschlange auf allen Warteschlangenmanagern definiert werden, in denen verwaltete Subskriptionen erstellt werden.

Wenn ein Attribut auf ASPARENT gesetzt ist, hängt das Verhalten des Abschnitts in ähnlicher Weise von den höheren Knoten in der Themenstruktur (siehe Verwaltungsthemenobjekte) auf jedem einzelnen Queue-Manager im Cluster ab. Dies kann zu einem anderen Verhalten beim Veröffentlichen oder Subskribieren von verschiedenen Warteschlangenmanagern führen.

Die Hauptattribute, die sich direkt auf das Publish/Subscribe-Verhalten im Cluster beziehen, lauten wie folgt:

CLROUTE
Dieser Parameter steuert das Routing von Nachrichten zwischen WS-Managern, in denen Publisher verbunden sind, und Warteschlangenmanagern, in denen übereinstimmende Subskriptionen vorhanden sind.

- Sie konfigurieren die Route entweder direkt zwischen diesen WS-Managern oder über einen Warteschlangenmanager, der eine Definition des Clusterthemas enthält. Weitere Informationen dazu finden Sie im Artikel Publish/Subscribe-Cluster.
- Sie können den **CLROUTE** nicht ändern, solange der Parameter **CLUSTER** festgelegt ist. Wenn Sie den **CLROUTE** ändern möchten, müssen Sie zunächst die Eigenschaft **CLUSTER** auf leer setzen. Dies stoppt Anwendungen, die das Thema in einer Cluster-Art verwenden. Dies führt wiederum zu einer Unterbrechung der Veröffentlichungen, die an Subskriptionen zugestellt werden, so dass Sie auch die Publish/Subscribe-Messaging während der Änderung in den Quiescemodus versetzt haben sollten.

PROXYSUB
Dieser Parameter steuert, wann Proxy-Subskriptionen erstellt werden.

- `FIRSTUSE` ist der Standardwert und bewirkt, dass Proxy-Subskriptionen als Antwort auf lokale Subskriptionen auf einem Warteschlangenmanager in einer verteilten Publish/Subscribe-Topologie gesendet werden und abgebrochen werden, wenn sie nicht mehr benötigt werden. Weitere Informationen darüber, warum Sie dieses Attribut möglicherweise vom Standardwert `FIRSTUSE` ändern möchten, finden Sie im Abschnitt Individuelle Proxy-Subskriptionsweiterleitung und *Veröffentlichung überall* .
- Um *publish überall* zu aktivieren, setzen Sie den Parameter **PROXYSUB** auf `FORCE` für ein übergeordneter Themenobjekt. Dies führt zu einer einzelnen Proxy-Subskription mit Platzhalterzeichen, die alle Topics unter diesem Themenobjekt in der Themenstruktur abgleicht.

Anmerkung: Wenn Sie das Attribut **PROXYSUB(FORCE)** in einem großen Publish/Subscribe-Cluster festlegen, kann es zu einer übermäßigen Auslastung der Systemressourcen kommen. Das Attribut **PROXYSUB(FORCE)** wird an jeden Warteschlangenmanager weitergegeben, nicht nur an den Warteschlangenmanager, auf dem das Thema definiert wurde. Dies bewirkt, dass jeder WS-Manager im Cluster ein Platzhalterzeichen für ein Platzhalterzeichen erstellt.

Eine Kopie einer Nachricht zu diesem Thema, die auf einem beliebigen WS-Manager im Cluster veröffentlicht wird, wird abhängig von der Einstellung **CLROUTE** an jeden Warteschlangenmanager im Cluster gesendet-entweder direkt oder über einen Topic-Host-Warteschlangenmanager.

Wenn das Thema direkt weitergeleitet wird, erstellt jeder WS-Manager Clustersenderkanäle zu jedem anderen Warteschlangenmanager. Wenn der Topic-Host weitergeleitet wird, werden die Kanäle zu jedem Topic-Host-Warteschlangenmanager von jedem WS-Manager im Cluster erstellt.

Weitere Informationen zum **PROXYSUB** -Parameter bei Verwendung in Clustern finden Sie unter Direct routed Publish/Subscribe performance .

PUBSCOBE und SUBSCOPE

Diese Parameter legen fest, ob dieser Warteschlangenmanager Veröffentlichungen an Warteschlangenmanager in der Topologie (Publish/Subscribe-Cluster oder Hierarchie) weitergibt oder den Geltungsbereich nur auf den lokalen WS-Manager beschränkt. Sie können den entsprechenden Job über das Programm mit `MQPMO_SCOPE_QMGR` und `MQSO_SCOPE_QMGR` ausführen.

PUBSCOPE

Wenn ein Clusterthemenobjekt mit **PUBSCOPE(QMGR)** definiert wird, wird die Definition gemeinsam mit dem Cluster verwendet, aber der Umfang der Veröffentlichungen, die auf diesem Thema basieren, ist nur lokal und wird nicht an andere WS-Manager im Cluster gesendet.

SUBSCOPE

Wenn ein Clusterthemenobjekt mit **SUBSCOPE(QMGR)** definiert wird, wird die Definition gemeinsam mit dem Cluster gemeinsam genutzt, aber der Geltungsbereich von Subskriptionen, die auf diesem Thema basieren, ist nur lokal. Daher werden keine Proxy-Subskriptionen an andere Warteschlangenmanager im Cluster gesendet.

Diese beiden Attribute werden im Allgemeinen zusammen verwendet, um einen Warteschlangenmanager von der Interaktion mit anderen Membern des Clusters zu bestimmten Themen zu trennen. Der Warteschlangenmanager veröffentlicht oder empfängt keine Veröffentlichungen zu diesen Themen in und von anderen Membern des Clusters. Diese Situation verhindert nicht die Veröffentlichung oder Subskription, wenn Themenobjekte in Unterabschnitten definiert sind.

Wenn Sie **SUBSCOPE** in einer lokalen Definition eines Themas auf `QMGR` setzen, werden andere WS-Manager im Cluster nicht daran gehindert, ihre Proxy-Subskriptionen an den Warteschlangenmanager weiterzugeben, wenn sie eine Clusterversion des Themas mit **SUBSCOPE(ALL)** verwenden. Wenn die lokale Definition jedoch auch **PUBSCOPE** auf `QMGR` setzt, werden diese Proxy-Subskriptionen keine Veröffentlichungen von diesem Warteschlangenmanager gesendet.

Zugehörige Konzepte

Veröffentlichungsumfang

Subskriptionsumfang

Mehrere Cluster-Topic-Definitionen mit demselben Namen

Sie können dasselbe benannte Clusterthemenobjekt in mehreren Warteschlangenmanagern im Cluster definieren, und in bestimmten Szenarios kann dies ein bestimmtes Verhalten ermöglichen. Wenn mehrere Clusterthemendefinitionen mit demselben Namen vorhanden sind, sollte die Mehrzahl der Eigenschaften übereinstimmen. Wenn dies nicht der Fall ist, werden in Abhängigkeit von der Signifikanz der Abweichung Fehler oder Warnungen ausgegeben.

Wenn in den Eigenschaften mehrerer Cluster-Topic-Definitionen eine Diskrepanz vorliegt, werden Warnungen ausgegeben und eine der Themenobjektdefinitionen wird von jedem WS-Manager im Cluster verwendet. Welche Definition von jedem WS-Manager verwendet wird, ist nicht deterministisch oder konsistent über die Warteschlangenmanager im Cluster hinweg. Solche Diskrepanzen sollten so schnell wie möglich gelöst werden.

Bei der Clusterkonfiguration oder -wartung müssen Sie manchmal mehrere Clusterthemendefinitionen erstellen, die nicht identisch sind. Dies ist jedoch nur als vorübergehende Maßnahme sinnvoll und wird daher als Fehlerbedingung behandelt.

Wenn Diskrepanzen festgestellt werden, werden die folgenden Warnungen in jedes Fehlerprotokoll des Warteschlangenmanagers geschrieben:

- Multi Unter Multiplatforms, AMQ9465 und AMQ9466.
- z/OS Unter z/OS, CSQX465I und CSQX466I.

The chosen properties for any topic string on each queue manager can be determined by viewing topic status rather than the topic object definitions, for example by using **DISPLAY TPSTATUS**.

In einigen Situationen ist ein Konflikt in den Konfigurationseigenschaften so schwer wiegend, dass das zu erstellende Themenobjekt gestoppt wird, oder dass die falsch übereinstimmenden Objekte als ungültig markiert und nicht im Cluster weitergegeben werden (siehe **CLSTATE** in DISPLAY TOPIC). Diese Situationen treten auf, wenn ein Konflikt in der Eigenschaft der Clusterweiterleitung (**CLROUTE**) der Themendefinitionen auftritt. Darüber hinaus werden weitere Inkonsistenzen aufgrund der Bedeutung der Konsistenz zwischen den Themenhost-Routing-Definitionen wie in den nachfolgenden Abschnitten dieses Artikels abgelehnt.

Wenn der Konflikt zu dem Zeitpunkt erkannt wird, zu dem das Objekt definiert ist, wird die Konfigurationsänderung zurückgewiesen. Wenn später von den vollständigen Repository-WS-Managern festgestellt wird, werden die folgenden Warnungen in die Fehlerprotokolle der WS-Manager geschrieben:

- Multi Unter Multiplatforms: AMQ9879.
- z/OS Unter z/OS: CSQX879E.

Wenn mehrere Definitionen desselben Themenobjekts im Cluster definiert sind, hat eine lokal definierte Definition Vorrang vor einer fernen Definition, die über Remotezugriff definiert ist. Wenn also Unterschiede in den Definitionen vorhanden sind, verhalten sich die WS-Manager, die die verschiedenen Definitionen hosten, unterschiedlich.

Die Auswirkung der Definition eines Nicht-Cluster-Themas mit demselben Namen wie ein Clusterthema aus einem anderen Warteschlangenmanager.

Es ist möglich, ein verwalttes Themenobjekt zu definieren, das sich nicht auf einem Warteschlangenmanager befindet, der sich in einem Cluster befindet, und gleichzeitig dasselbe benannte Themenobjekt wie eine Clusterthemendefinition in einem anderen Warteschlangenmanager definieren. In diesem Fall hat das lokal definierte Themenobjekt Vorrang vor allen fernen Definitionen mit dem gleichen Namen.

Dadurch wird verhindert, dass das Clustering-Verhalten des Themas bei Verwendung dieses Warteschlangenmanagers verhindert wird. Dies bedeutet, dass Subskriptionen möglicherweise keine Veröffentlichungen von fernen Publishern empfangen, und Nachrichten von Publishern werden möglicherweise nicht an ferne Subskriptionen im Cluster weitergegeben.

Vor der Konfiguration eines solchen Systems sollte sorgfältig geprüft werden, da dies zu verwirrenden Verhaltensweisen führen kann.

Anmerkung: Wenn ein einzelner Warteschlangenmanager Veröffentlichungen und Subskriptionen von der Weitergabe an den Cluster verhindern muss, selbst wenn das Thema an anderer Stelle in einem Cluster zusammengefasst wurde, besteht ein alternativer Ansatz darin, die Veröffentlichungs-und Subskriptionsbereiche nur auf den lokalen WS-Manager zu setzen. Siehe „Clusterthemenattribute” auf Seite 104.

Mehrere Cluster-Topic-Definitionen in einem Cluster mit direktem Routing

Für direktes Routing definieren Sie in der Regel einen Clusterabschnitt nicht für mehrere Cluster-Queue-Manager. Dies liegt daran, dass das direkte Routing das Thema an allen Warteschlangenmanagern im Cluster verfügbar macht, unabhängig davon, auf welchem Warteschlangenmanager er definiert wurde. Darüber hinaus erhöht das Hinzufügen mehrerer Clusterthemendefinitionen die Systemaktivität und die Verwaltungskomplexität erheblich, und die Wahrscheinlichkeit, dass die Komplexität zunimmt, erhöht die Wahrscheinlichkeit eines Benutzerfehlers:

- Jede Definition führt dazu, dass ein weiteres Clusterthemenobjekt an die anderen WS-Manager im Cluster übertragen wird, einschließlich der anderen Cluster-Topic-Host-Warteschlangenmanager.

- Alle Definitionen für ein bestimmtes Thema in einem Cluster müssen identisch sein. Andernfalls ist es schwierig, herauszufinden, welche Themendefinition von einem WS-Manager verwendet wird.

Es ist außerdem nicht unbedingt erforderlich, dass der einzige Host-WS-Manager permanent für die ordnungsgemäße Funktion des Themas im Cluster verfügbar ist, da die Clusterthemendefinition von den vollständigen WS-Managern des Repositorys und von allen anderen Warteschlangenmanagern in den Teilclusterrepositorys zwischengespeichert wird. Weitere Informationen hierzu finden Sie im Abschnitt Verfügbarkeit von Topic-Host-WS-Managern, die direktes Routing verwenden .

Für eine Situation, in der Sie möglicherweise vorübergehend ein Clusterthema auf einem zweiten Warteschlangenmanager definieren müssen, z. B., wenn der vorhandene Host des Themas aus dem Cluster entfernt werden soll, finden Sie weitere Informationen unter Clusterthemendefinition in einen anderen Warteschlangenmanager verschieben .

Wenn Sie die Definition eines Cluster-Topics ändern müssen, achten Sie darauf, sie in dem Warteschlangenmanager zu ändern, in dem sie auch definiert wurde. Der Versuch, ihn von einem anderen WS-Manager zu ändern, kann versehentlich eine zweite Definition des Themas mit widersprüchlichen Themenattributen erstellen.

Mehrere Cluster-Topic-Definitionen in einem Cluster mit Topic-Host-Routing

Wenn ein Clusterthema mit einer Clusterroute von *topic host* definiert wird, wird das Thema in allen WS-Managern im Cluster genauso wie für *direkte* weitergeleitete Themen weitergegeben. Darüber hinaus wird das gesamte Publish/Subscribe-Messaging für dieses Thema über die Warteschlangenmanager weitergeleitet, in denen dieses Thema definiert ist. Daher wird die Position und die Anzahl der Definitionen des Themas im Cluster wichtig (siehe „Thema Host-Routing in Publish/Subscribe-Clustern” auf Seite 88).

Um eine ausreichende Verfügbarkeit und Skalierbarkeit zu gewährleisten, ist es sinnvoll, wenn möglich mehrere Themendefinitionen zu haben. Siehe Verfügbarkeit von Topic-Host-WS-Managern, die Topic-Host-Routing verwenden .

Beim Hinzufügen oder Entfernen zusätzlicher Definitionen eines *Topic-Host* -Themas in einem Cluster sollten Sie den Fluss der Nachrichten zum Zeitpunkt der Konfigurationsänderung berücksichtigen. Wenn zum Zeitpunkt der Änderung Nachrichten im Cluster zu dem Thema veröffentlicht werden, ist ein zwischengespeicherten Prozess erforderlich, um eine Themendefinition hinzuzufügen oder zu entfernen. Weitere Informationen hierzu finden Sie im Abschnitt Clusterthemendefinition in einen anderen Warteschlangenmanager verschieben und Weitere Themenhosts zu einem Topic-Host-Routing-Cluster hinzufügen .

Wie bereits erläutert, sollten die Eigenschaften der Mehrfachdefinitionen mit der möglichen Ausnahme des **PUB** -Parameters übereinstimmen, wie im nächsten Abschnitt beschrieben. Wenn Veröffentlichungen über Topic-Host-Warteschlangenmanager weitergeleitet werden, ist es sogar noch wichtiger, dass mehrere Definitionen konsistent sind. Daher wird eine Inkonsistenz, die in der Themenzeichenfolge oder dem Clusternamen festgestellt wurde, zurückgewiesen, wenn eine oder mehrere der Themendefinitionen für das Thema Host-Cluster-Routing konfiguriert wurden.

Anmerkung: Clusterthemendefinitionen werden auch zurückgewiesen, wenn versucht wird, sie oberhalb oder unterhalb eines anderen Themas in der Themenstruktur zu konfigurieren, in dem die vorhandene Clusterthemendefinition für das Thema Host-Routing konfiguriert ist. Dies verhindert die Mehrdeutigkeit bei der Weiterleitung von Veröffentlichungen in Bezug auf Platzhaltersubskriptionen.

Sonderbehandlung für den Parameter PUB

Der Parameter **PUB** wird verwendet, um zu steuern, wann Anwendungen in einem Thema veröffentlichen können. Im Fall des Topic-Host-Routing in einem Cluster kann er auch steuern, welche Topic-Host-Warteschlangenmanager verwendet werden, um Veröffentlichungen zu verlegen. Aus diesem Grund ist es zulässig, dass mehrere Definitionen desselben Themenobjekts im Cluster mit unterschiedlichen Einstellungen für den Parameter PUB vorhanden sind.

Wenn mehrere ferne Clusterdefinitionen eines Themas über unterschiedliche Einstellungen für diesen Parameter verfügen, ermöglicht das Thema, dass Veröffentlichungen an Subskriptionen gesendet und zugestellt werden, wenn die folgenden Bedingungen erfüllt sind:

- Es ist kein übereinstimmender Themenobjekt definiert, der auf dem Warteschlangenmanager definiert ist, mit dem der Publisher verbunden ist, der auf PUB(DISABLED) gesetzt ist.
- Mindestens eine der mehreren Themendefinitionen im Cluster ist auf PUB(ENABLED) gesetzt, oder es wird mindestens eine der Themendefinitionen auf PUB(ASPARENT) festgelegt und die lokalen Warteschlangenmanager, in denen der Bereitsteller verbunden ist, und die definierte Subskription auf PUB(ENABLED) an einem höheren Punkt in der Themenstruktur gesetzt.

Für Topic-Host-Routing, wenn Nachrichten von Anwendungen veröffentlicht werden, die mit Warteschlangenmanagern verbunden sind, die keine Topic-Hosts sind, werden Nachrichten nur an den Topic-Host-Warteschlangenmanager weitergeleitet, in dem der Parameter **PUB** nicht explizit auf DISABLED gesetzt wurde. Sie können daher die Einstellung PUB (DISABLED) verwenden, um den Nachrichtenverkehr über bestimmte Themenhosts in den Quiescemodus zu setzen. Möglicherweise möchten Sie dies tun, um die Wartung oder das Entfernen eines Warteschlangenmanagers vorzubereiten, oder aus den Gründen, die im Abschnitt Weitere Topic-Hosts zu einem Topic-Host-Routing-Cluster hinzufügen beschrieben werden.

Verfügbarkeit von Cluster-Topic-Host-Warteschlangenmanagern
Entwerfen Sie Ihren Publish/Subscribe-Cluster, um das Risiko zu minimieren, dass der Cluster nicht mehr in der Lage ist, den Datenverkehr für das Thema zu verarbeiten, wenn ein Topic-Host-Warteschlangenmanager nicht mehr verfügbar ist. Die Auswirkung eines Topic-Host-Warteschlangenmanagers, der nicht verfügbar wird, hängt davon ab, ob der Cluster Topic-Host-Routing oder direktes Routing verwendet.

Verfügbarkeit von Topic-Host-Warteschlangenmanagern, die direktes Routing verwenden

Für direktes Routing definieren Sie in der Regel einen Clusterabschnitt nicht für mehrere Cluster-Queue-Manager. Dies liegt daran, dass das direkte Routing das Thema an allen Warteschlangenmanagern im Cluster verfügbar macht, unabhängig davon, auf welchem Warteschlangenmanager er definiert wurde. Weitere Informationen finden Sie unter Mehrere Cluster-Topic-Definitionen in einem Cluster mit direkter Weiterleitung.

Wenn in einem Cluster der Host eines Clusterobjekts (z. B. eine Clusterwarteschlange oder ein Clusterthema) für einen längeren Zeitraum nicht mehr verfügbar ist, werden die anderen Mitglieder des Clusters schließlich die Kenntnis dieser Objekte verfallen lassen. Wenn der Cluster-Topic-Host-Warteschlangenmanager in einem Clusterthema nicht mehr verfügbar ist, verarbeiten die anderen Warteschlangenmanager weiterhin Publish/Subscribe-Anforderungen für das Thema in einer direkten Cluster-Methode (d. B. das Senden von Veröffentlichungen an Subskriptionen auf fernen Warteschlangenmanagern) für mindestens 60 Tage ab dem Zeitpunkt, ab dem der Topic-Hosting-Warteschlangenmanager zuletzt in der Kommunikation mit den vollständigen WS-Managern der Repository-WS-Managern stand. Wenn der Warteschlangenmanager, auf dem Sie das Clusterthemenobjekt definiert haben, nie wieder verfügbar gemacht wird, werden die zwischengespeicherten Themenobjekte auf den anderen Warteschlangenmanagern gelöscht, und das Thema wird auf ein lokales Thema zurückgesetzt, in dem die Subskriptionen nicht mehr Veröffentlichungen von Anwendungen empfangen, die mit fernen Warteschlangenmanagern verbunden sind.

Wenn der Warteschlangenmanager, auf dem Sie ein Clusterthemenobjekt definieren, mit dem 60-Tage-Zeitraum wiederhergestellt werden soll, müssen keine speziellen Maßnahmen ergriffen werden, um sicherzustellen, dass ein Clusterthemenhost verfügbar bleibt (beachten Sie jedoch, dass alle Subskriptionen, die auf dem nicht verfügbaren Clusterthemenhost definiert sind, nicht verfügbar bleiben). Der 60-Tage-Zeitraum reicht aus, um technische Probleme zu erfüllen, und wird wahrscheinlich nur aufgrund von Verwaltungsfehlern überschritten. Wenn der Cluster-Topic-Host nicht verfügbar ist, schreiben alle Member des Clusters stündlich Fehlerprotokollnachrichten, die stündlich angeben, dass ihr zwischengespeichertes Clusterthemenobjekt nicht aktualisiert wurde, um diese Möglichkeit zu beheben. Beantworten Sie diese Nachrichten, indem Sie sicherstellen, dass der WS-Manager, auf dem das Clusterthemenobjekt definiert ist, aktiv ist. Wenn es nicht möglich ist, den Cluster-Topic-Host-Warteschlangenmanager wieder verfügbar zu machen, definieren Sie die gleiche Clusterthemendefinition mit genau denselben Attributen in einem anderen Warteschlangenmanager im Cluster.

Verfügbarkeit von Topic-Host-Warteschlangenmanagern, die Topic-Host-Routing verwenden

Für das Topic-Host-Routing wird die gesamte Publish/Subscribe-Nachrichtenübertragung für ein Thema über die Warteschlangenmanager weitergeleitet, in denen dieses Thema definiert ist. Aus diesem Grund ist es sehr wichtig, dass die ständige Verfügbarkeit dieser WS-Manager im Cluster berücksichtigt wird. Wenn ein Themenhost nicht mehr verfügbar ist und kein anderer Host für das Thema vorhanden ist, wird der Datenverkehr von Publishern zu Subskribenten auf verschiedenen Warteschlangenmanagern im Cluster sofort für das Thema angehalten. Wenn weitere Topic-Hosts verfügbar sind, leiten die Cluster-WS-Manager den neuen Veröffentlichungsdatenverkehr durch diese Themenhosts, wodurch die kontinuierliche Verfügbarkeit von Nachrichtenrouten bereitgestellt wird.

Wie bei direkten Themen wird nach 60 Tagen, wenn der erste Themenhost noch nicht verfügbar ist, die Kenntnis des Topic-Hostthemas aus dem Cluster entfernt. Wenn es sich dabei um die letzte verbleibende Definition für dieses Thema im Cluster handelt, werden alle anderen Warteschlangenmanager die Weiterleitung von Veröffentlichungen an einen beliebigen Themenhost nicht mehr weiterleiten.

Um eine ausreichende Verfügbarkeit und Skalierbarkeit zu gewährleisten, ist es daher sinnvoll, wenn möglich, jedes Thema auf mindestens zwei Cluster-WS-Managern zu definieren. Dadurch wird der Schutz vor einem bestimmten Topic-Host-WS-Manager, der nicht mehr verfügbar ist Siehe auch Mehrere Clusterthemendefinitionen in einem Topic-Host-Routing-Cluster .

Wenn Sie nicht mehrere Themenhosts konfigurieren können (z. B. weil Sie die Nachrichtenreihenfolge beibehalten müssen) und Sie nicht nur einen Topic-Host konfigurieren können (weil die Verfügbarkeit eines einzelnen Warteschlangenmanagers den Fluss der Veröffentlichungen nicht auf Subskriptionen für alle Warteschlangenmanager im Cluster auswirken darf), ist es in Betracht zu ziehen, das Thema als direktes weitergeleitete Thema zu konfigurieren. Dadurch wird die Abhängigkeit von einem einzelnen Warteschlangenmanager für den gesamten Cluster vermieden, aber es ist immer noch erforderlich, dass jeder einzelne WS-Manager verfügbar ist, damit er lokal gehostete Subskriptionen und Publisher verarbeiten kann.

Clusterveröffentlichungs-/Subskriptionssubskribieren

Durch die Einführung des ersten direkt weitergeleiteten Clusterthemas in einen Cluster wird jeder WS-Manager im Cluster gezwungen, jeden anderen Warteschlangenmanager zu kennen und kann die Kanäle dazu bringen, Kanäle zu erstellen. Wenn dies nicht wünschenswert ist, sollten Sie stattdessen Topic-Host-Routing-Publish/Subscribe konfigurieren. Wenn das Vorhandensein eines direkt weitergeleiteten Clusterthemas die Stabilität des Clusters gefährden könnte, können Sie die Cluster-Publish/Subscribe-Funktionalität aufgrund von Skalierungsbedenken jedes Warteschlangenmanagers vollständig inaktivieren, indem Sie **PSCLUS** auf jedem WS-Manager im Cluster auf `DISABLED` setzen.

Wie unter „Direktes Routing in Publish/Subscribe-Clustern” auf Seite 83 beschrieben, werden bei der Einführung eines direkt weitergeleiteten Cluster-Topics im Cluster alle Teilrepositorys automatisch über alle anderen Mitglieder des Clusters benachrichtigt. Das Clusterthema kann auch Subskriptionen auf allen anderen Knoten erstellen (z. B. wo **PROXYSUB(FORCE)** angegeben ist) und verursacht eine große Anzahl von Kanälen, die von einem Warteschlangenmanager aus gestartet werden, auch wenn keine lokalen Subskriptionen vorhanden sind. Dadurch wird jedem WS-Manager im Cluster eine sofortige zusätzliche Belastung angezeigt. Für einen Cluster, der viele Warteschlangenmanager enthält, kann dies zu einer erheblichen Leistungsminderung führen. Daher muss die Einführung von Direct-Routing-Publish/Subscribe in einem Cluster sorgfältig geplant werden.

Wenn Sie wissen, dass ein Cluster die Overheads von Direct Routing Publish/Subscribe nicht aufnehmen kann, können Sie stattdessen Topic-Host-Routing-Publish/Subscribe verwenden. Eine Übersicht über die Unterschiede finden Sie in „Publish/Subscribe-Cluster entwerfen” auf Seite 81.

Wenn Sie es vorziehen, die Publish/Subscribe-Funktionalität für den Cluster vollständig zu inaktivieren, können Sie dies tun, indem Sie das WS-Managerattribut **PSCLUS** auf jedem WS-Manager im Cluster auf `DISABLED` setzen. Diese Einstellung inaktiviert die Publish/Subscribe-Publish/Subscribe im Cluster, indem drei Aspekte der WS-Manager-Funktionalität geändert werden:

- Ein Administrator dieses Warteschlangenmanagers ist nicht mehr in der Lage, ein `Topic` -Objekt als Cluster zu definieren.

- Eingehende Themendefinitionen oder Proxy-Subskriptionen von anderen Warteschlangenmanagern werden zurückgewiesen, und es wird eine Warnung protokolliert, um den Administrator über eine falsche Konfiguration zu informieren.
- Vollständige Repositorys teilen nicht mehr automatisch Informationen zu jedem Warteschlangenmanager mit allen anderen Teilrepositorys, wenn sie eine Themendefinition empfangen.

Obwohl **PSCLUS** ein Parameter jedes einzelnen Warteschlangenmanagers in einem Cluster ist, ist es nicht beabsichtigt, die Publish/Subscribe-Subskription in einer Untergruppe von Warteschlangenmanagern im Cluster selektiv zu inaktivieren. Wenn Sie auf diese Weise selektiv inaktivieren, werden häufige Fehlernachrichten angezeigt. Dies liegt daran, dass Proxy-Subskriptionen und Themendefinitionen permanent angezeigt und zurückgewiesen werden, wenn ein Thema in einem Warteschlangenmanager in einem Cluster zusammengefasst ist, in dem **PSCLUS** aktiviert ist.

Sie sollten daher versuchen, **PSCLUS** auf jedem WS-Manager im Cluster auf `DISABLED` zu setzen. In der Praxis kann es jedoch schwierig sein, diesen Status zu erreichen und zu verwalten, z. B. Warteschlangenmanager können beitreten und den Cluster zu jedem Zeitpunkt verlassen. Sie müssen mindestens sicherstellen, dass **PSCLUS** für alle vollständigen WS-Manager-Repository-Warteschlangenmanager auf `DISABLED` gesetzt ist. Wenn Sie dies tun und anschließend ein Clusterthema in einem `ENABLED` -Warteschlangenmanager im Cluster definiert wird, führt dies nicht dazu, dass die vollständigen Repositorys alle Warteschlangenmanager jedes anderen Warteschlangenmanagers informieren, und so dass Ihr Cluster vor potenziellen Skalierungsproblemen für alle Warteschlangenmanager geschützt ist. In diesem Szenario wird der Ursprung des Clusterthemas in den Fehlerprotokollen der vollständigen WS-Manager-Repositorys dokumentiert.

Wenn ein Warteschlangenmanager an einem oder mehreren Publish/Subscribe-Clustern und einem oder mehreren Punkt-zu-Punkt-Clustern beteiligt ist, müssen Sie **PSCLUS** auf `ENABLED` in diesem Warteschlangenmanager setzen. Aus diesem Grund sollten Sie bei der Überschneidung eines Punkt-zu-Punkt-Clusters mit einem Publish/Subscribe-Cluster eine separate Gruppe vollständiger Repositorys in jedem Cluster verwenden. Mit dieser Methode können Themendefinitionen und Informationen zu jedem WS-Manager nur im Publish/Subscribe-Cluster fließen.

Um inkonsistente Konfigurationen zu vermeiden, wenn Sie **PSCLUS** von `ENABLED` in `DISABLED` ändern, können keine Clusterthemenobjekte in einem Cluster vorhanden sein, in dem dieser WS-Manager Mitglied ist. Alle solchen Themen, die auch über Remotezugriff definiert sind, müssen gelöscht werden, bevor **PSCLUS** in `DISABLED` geändert wird.

Weitere Informationen zu **PSCLUS** finden Sie in ALTER QMGR (PSCLUS) .

Zugehörige Konzepte
Direkte Publish/Subscribe-Clusterleistung

Publish/Subscribe und mehrere Cluster

Ein einzelner WS-Manager kann Mitglied mehrerer Cluster sein. Diese Anordnung wird manchmal auch als *überlappende Cluster* bezeichnet. Durch eine solche Überlappung können Clusterwarteschlangen von mehreren Clustern aus zugänglich gemacht werden, und der Punkt-zu-Punkt-Datenverkehr kann von Warteschlangenmanagern in einem Cluster an Warteschlangenmanager in einem anderen Cluster weitergeleitet werden. Clusterthemen in Publish/Subscribe-Clustern bieten nicht die gleiche Funktionalität. Daher muss ihr Verhalten bei der Verwendung mehrerer Cluster klar verstanden werden.

Anders als bei einer Warteschlange können Sie eine Themendefinition nicht mehr als einem Cluster zuordnen. Der Geltungsbereich eines Clusterthemas ist auf die WS-Manager im selben Cluster beschränkt, für die das Thema definiert ist. Auf diese Weise können Veröffentlichungen an Subskriptionen nur auf diesen Warteschlangenmanagern in demselben Cluster weitergegeben werden.

Themenstruktur eines Warteschlangenmanagers

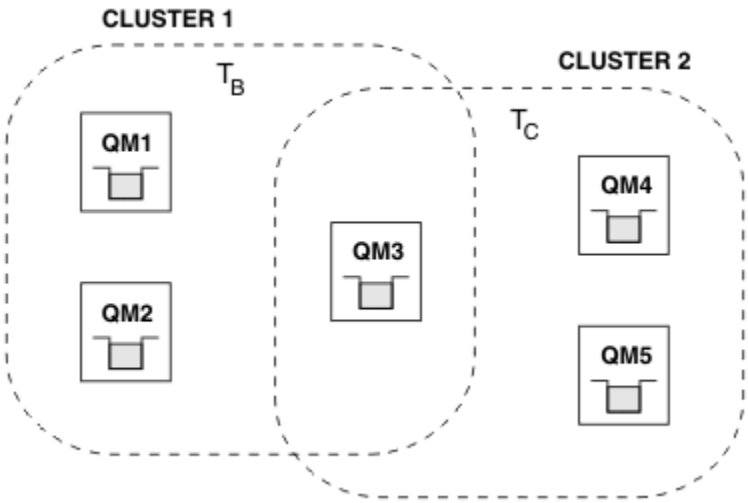


Abbildung 28. Überlappende Cluster: Zwei Cluster, die jeweils verschiedene Themen subskribieren

Wenn ein Warteschlangenmanager Mitglied mehrerer Cluster ist, wird er auf alle in den einzelnen Clustern definierten Clusterthemen aufmerksam gemacht. Zum Beispiel kennt in der vorherigen Abbildung QM3 die verwalteten Clusterthemenobjekte T_B und T_C, während QM1 nur T_B kennt. QM3 wendet beide Themendefinitionen auf sein lokales Thema an und hat daher für bestimmte Themen ein anderes Verhalten gegenüber QM1. Aus diesem Grund ist es wichtig, dass die Clusterthemen aus verschiedenen Clustern sich nicht gegenseitig stören. Kollisionen können auftreten, wenn ein Cluster-Topic oberhalb oder unterhalb eines anderen Clusterthemas in einem anderen Cluster definiert ist (z. B. haben sie Themenzeichenfolgen von `/Sport` und `/Sport/Football`) oder sogar für dieselbe Themenzeichenfolge in beiden. Eine andere Form der Kollision ist, wenn verwaltete Clusterthemenobjekte mit demselben Objektnamen in verschiedenen Clustern definiert werden, aber für unterschiedliche Themenzeichenfolgen.

Wenn eine solche Konfiguration vorgenommen wird, wird die Bereitstellung von Veröffentlichungen zu übereinstimmenden Subskriptionen sehr abhängig von den relativen Positionen der Bereitstellungs-und Subskribenten in Bezug auf den Cluster. Aus diesem Grund können Sie sich nicht auf eine solche Konfiguration verlassen, und Sie sollten es ändern, um die kollierenden Themen zu entfernen.

Wenn Sie eine überlappende Clustertopologie mit Publish/Subscribe-Messaging planen, können Sie jede Kollision vermeiden, indem Sie die Themenstruktur-und Clusterthemenobjektnamen so behandeln, als ob sie sich über alle überlappenden Cluster in der Topologie erstrecken.

Mehrere Publish/Subscribe-Cluster integrieren

Wenn Publish/Subscribe-Messaging in verschiedenen Clustern über Publish/Subscribe-Messaging erforderlich ist, stehen zwei Optionen zur Verfügung:

- Verbinden Sie die Cluster miteinander, indem Sie eine Publish/Subscribe-Hierarchiekonfiguration verwenden. Weitere Informationen hierzu finden Sie im Abschnitt Themenbereiche mehrerer Cluster zusammenfassen.
- Erstellen Sie einen zusätzlichen Cluster, der die vorhandenen Cluster überlagert, und schließt alle Warteschlangenmanager ein, die ein bestimmtes Thema veröffentlichen oder subskribieren müssen.

Bei letzterer Option sollten Sie die Größe des Clusters und den effizientesten Cluster-Routing-Mechanismus sorgfältig prüfen. Siehe„Publish/Subscribe-Cluster entwerfen” auf Seite 81.

Designüberlegungen zu ständigen Veröffentlichungen in Publish/Subscribe-Clustern

Beim Entwurf eines Publish/Subscribe-Clusters für die Arbeit mit ständigen Veröffentlichungen sind einige Einschränkungen zu beachten.

Hinweise

Hinweise 1: Die folgenden Cluster-WS-Manager speichern immer die aktuellste Version einer ständigen Veröffentlichung:

- Der Warteschlangenmanager des Publishers
- In einem Topic-Host-Routing-Cluster der Themenhost (vorausgesetzt, es gibt nur einen Themenhost für das Thema, wie im nächsten Abschnitt dieses Artikels erläutert).
- Alle WS-Manager mit Subskriptionen, die mit der Themenzeichenfolge der ständigen Veröffentlichung übereinstimmen

Hinweise 2: Warteschlangenmanager erhalten keine aktualisierten ständigen Veröffentlichungen, während sie keine Subskriptionen haben. Daher wird jede gespeicherte Publizierung, die auf einem Warteschlangenmanager gespeichert ist und nicht mehr für das Thema subskribiert wird, veraltete Veröffentlichungen.

Hinweise 3: Wenn bei der Erstellung einer Subskription eine lokale Kopie einer ständigen Veröffentlichung für die Themenzeichenfolge vorhanden ist, wird die lokale Kopie an die Subskription übergeben. Wenn Sie der erste Subskribent für eine beliebige Themenzeichenfolge sind, wird eine übereinstimmende ständige Veröffentlichung auch von einem der folgenden Cluster-Member bereitgestellt:

- In einem direkt weitergeleiteten Cluster der Warteschlangenmanager des Publishers
- In einem Topic-Host-Routing-Cluster die Topic-Hosts für das angegebene Thema

Die Zustellung einer ständigen Veröffentlichung von einem Themenhost oder veröffentlichenden Warteschlangenmanager an den subskribierenden Warteschlangenmanager erfolgt asynchron zu den MQSUB-Aufrufen. Wenn Sie daher den Aufruf MQSUBRQ verwenden, wird die letzte ständige Veröffentlichung möglicherweise bis zu einem nachfolgenden Aufruf von MQSUBRQ verpasst.

Implikationen

Bei einem Publish/Subscribe-Cluster speichert der lokale WS-Manager beim Erstellen einer ersten Subskription möglicherweise eine veraltete Kopie einer ständigen Veröffentlichung, und dies ist die Kopie, die an die neue Subskription zugestellt wird. Das Vorhandensein einer Subskription auf dem lokalen WS-Manager bedeutet, dass dies beim nächsten Aktualisieren der ständigen Veröffentlichung behoben werden wird.

Wenn Sie für einen Topic-Host-Publish/Subscribe-Cluster mehr als einen Topic-Host für ein bestimmtes Thema konfigurieren, erhalten neue Subskribenten möglicherweise die neueste ständige Veröffentlichung von einem Themenhost oder sie erhalten möglicherweise eine veraltete Veröffentlichung von einem anderen Themenhost (mit der letzten verloren gegangenen). Für Topic-Host-Routing ist es üblich, mehrere Topic-Hosts für ein bestimmtes Thema zu konfigurieren. Wenn Sie jedoch von Anwendungen erwarten, dass sie ständige Veröffentlichungen verwenden, sollten Sie für jedes Thema nur einen Themahost konfigurieren.

Für alle angegebenen Themenzeichenfolgen sollten Sie nur einen einzigen Publisher verwenden und sicherstellen, dass der Publisher immer denselben Warteschlangenmanager verwendet. Wenn dies nicht der Fall ist, können verschiedene ständige Veröffentlichungen an verschiedenen Warteschlangenmanagern für dasselbe Thema aktiv sein, was zu unerwartetem Verhalten führt. Da mehrere Proxy-Subskriptionen verteilt sind, können mehrere ständige Veröffentlichungen empfangen werden.

Wenn die Subskribenten weiterhin über veraltete Veröffentlichungen besorgt sind, sollten Sie beim Erstellen jeder ständigen Veröffentlichung die Einstellung eines Nachrichtenablaufes in Erwägung ziehen.

Mit dem Befehl **CLEAR TOPICSTR** können Sie eine ständige Veröffentlichung aus einem Publish/Subscribe-Cluster entfernen. Unter bestimmten Umständen müssen Sie den Befehl möglicherweise auf mehreren Membern des Publish/Subscribe-Clusters absetzen, wie in **CLEAR TOPICSTR** beschrieben.

Subskriptionen mit Platzhalterzeichen und ständige Veröffentlichungen

Wenn Sie Platzhaltersubskriptionen verwenden, werden die entsprechenden Proxy-Subskriptionen, die anderen Mitgliedern des Publish/Subscribe-Clusters bereitgestellt werden, vom Topic-Trennzeichen unmittelbar vor dem ersten Platzhalterzeichen aus dem Topic-Trennzeichen entfernt. Siehe Wildcards und Clusterthemen.

Daher kann das verwendete Platzhalterzeichen möglicherweise mehr Themenzeichenfolgen und mehr ständigen Veröffentlichungen entsprechen, als es mit der subskribierenden Anwendung übereinstimmt.

Dadurch wird der für die ständigen Veröffentlichungen benötigte Speicherplatz erhöht, und Sie müssen daher sicherstellen, dass die Speicherkapazität der Host-WS-Manager ausreicht.

Zugehörige Konzepte

Ständige Veröffentlichungen

Individuelle Proxy-Abonnementweiterleitung und Veröffentlichen überall

Hinweise zu REFRESH CLUSTER für Publish/Subscribe-Cluster

Die Ausgabe des Befehls **REFRESH CLUSTER** führt dazu, dass der Warteschlangenmanager vorübergehend lokal gehaltene Informationen zu einem Cluster löscht, einschließlich aller Clusterthemen und der zugehörigen Proxy-Subskriptionen.

Die Zeit, die von der Ausgabe des Befehls **REFRESH CLUSTER** bis zu dem Punkt, an dem der Warteschlangenmanager die erforderlichen Informationen für das Cluster-Publish/Subscribe erhält, benötigt wird, hängt von der Größe des Clusters, der Verfügbarkeit und der Reaktionsfähigkeit der Warteschlangenmanager mit vollständigem Repository ab.

Während der Aktualisierungsverarbeitung erfolgt die Unterbrechung des Publish/Subscribe-Datenverkehrs in einem Publish/Subscribe-Cluster. Bei großen Clustern kann die Verwendung des Befehls **REFRESH CLUSTER** den Cluster unterbrechen, während er in Bearbeitung ist, und danach in 27-Tage-Intervallen, wenn die Clusterobjekte automatisch Statusaktualisierungen an alle interessierten Warteschlangenmanager senden. Nähere Informationen hierzu erhalten Sie im Abschnitt Die Aktualisierung in einem großen Cluster kann sich auf die Leistung und Verfügbarkeit auswirken. Daher darf der Befehl **REFRESH CLUSTER** in einem Publish/Subscribe-Cluster nur unter Anleitung durch das zuständige IBM Support Center verwendet werden.

Die Unterbrechung des Clusters kann extern als die folgenden Symptome auftreten:

- Subskriptionen zu Clusterthemen in diesem WS-Manager erhalten keine Veröffentlichungen von Publishern, die mit anderen WS-Managern im Cluster verbunden sind.
- Nachrichten, die in Clusterthemen in diesem Warteschlangenmanager veröffentlicht werden, werden nicht an Subskriptionen auf anderen Warteschlangenmanagern weitergegeben.
- Subskriptionen für Clusterthemen in diesem Warteschlangenmanager, die in diesem Zeitraum erstellt wurden, senden nicht konsistent Proxy-Subskriptionen an andere Mitglieder des Clusters.
- Subskriptionen für Clusterthemen auf diesem Warteschlangenmanager, die in diesem Zeitraum gelöscht wurden, entfernen nicht konsistent die Proxy-Subskriptionen von anderen Membern des Clusters.
- 10-Sekunden-Pausen oder länger, bei Nachrichtenübermittlung.
- **MQPUT**-Fehler, z. B. MQRC_PUBLICATION_FAILURE.
- Veröffentlichungen, die in der Warteschlange für nicht zustellbare Nachrichten mit dem Grund MQRC_UNKNOWN_REMOTE_Q_MGR platziert wurden

Aus diesen Gründen müssen Publish/Subscribe-Anwendungen in den Quiescemodus versetzt werden, bevor der Befehl **REFRESH CLUSTER** ausgegeben wird.

Nachdem ein **REFRESH CLUSTER** -Befehl auf einem Warteschlangenmanager in einem Publish/Subscribe-Cluster ausgegeben wurde, warten Sie, bis alle Clusterwarteschlangenmanager und Clusterthemen erfolgreich aktualisiert wurden, und synchronisieren Sie dann die Proxy-Subskriptionen wie unter Resynchronisation von Proxy-Subskriptionenbeschrieben. Wenn alle Proxy-Subskriptionen ordnungsgemäß resynchronisiert wurden, starten Sie Ihre Publish/Subscribe-Anwendungen erneut.

Wenn die Ausführung eines **REFRESH CLUSTER** -Befehls viel Zeit in Anspruch nimmt, können Sie ihn überwachen, indem Sie sich die CURDEPTH von SYSTEM.CLUSTER.COMMAND.QUEUEansehen.

Zugehörige Konzepte

„Clustering: Best Practices für REFRESH CLUSTER verwenden" auf Seite 74
Sie verwenden den Befehl **REFRESH CLUSTER** , um alle lokal gespeicherten Informationen zu einem Cluster zu löschen und diese Informationen aus den vollständigen Repositorys im Cluster erneut zu erstellen. Sie sollten diesen Befehl nicht verwenden, außer in außergewöhnlichen Umständen. Wenn Sie es verwenden müssen, gibt es besondere Hinweise darauf, wie Sie es verwenden. Diese Informationen sind ein Leitfaden, der auf Tests und Feedback von Kunden basiert.

Zugehörige Verweise

Anwendungsprobleme bei der Ausführung von REFRESH CLUSTER
MQSC-Befehlsreferenz: REFRESH CLUSTER

Routing in Publish/Subscribe-Hierarchien

Wenn Ihre verteilte WS-Manager-Topologie eine Publish/Subscribe-Hierarchie ist und eine Subskription auf einem WS-Manager erfolgt, wird standardmäßig eine Proxy-Subskription auf jedem Warteschlangenmanager in der Hierarchie erstellt. Veröffentlichungen, die auf einem beliebigen WS-Manager empfangen werden, werden dann über die Hierarchie an jeden Warteschlangenmanager weitergeleitet, der eine übereinstimmende Subskription enthält.

Eine Einführung dazu, wie Nachrichten zwischen Warteschlangenmanagern in Publish/Subscribe-Hierarchien und Clustern weitergeleitet werden, finden Sie unter Verteilte Publish/Subscribe-Netze.

Wenn eine Subskription für ein Thema in einem Warteschlangenmanager in einer verteilten Publish/Subscribe-Hierarchie ausgeführt wird, verwaltet der Warteschlangenmanager den Prozess, mit dem die Subskription an verbundene Warteschlangenmanager weitergegeben wird. *Proxy-Subskriptionen* fließen zu allen Warteschlangenmanagern im Netz. Eine Proxy-Subskription gibt einem WS-Manager die Informationen, die er benötigt, um eine Veröffentlichung an diese Warteschlangenmanager weiterzuleiten, die Subskriptionen für dieses Thema enthalten. Jeder WS-Manager in einer Publish/Subscribe-Hierarchie kennt nur seine direkten Beziehungen. Veröffentlichungen, die an einen Warteschlangenmanager gestellt werden, werden über die direkten Beziehungen zu diesen Warteschlangenmanagern mit Subskriptionen gesendet. Dies wird in der folgenden Abbildung veranschaulicht, in der *Subskribent 1* eine Subskription für ein bestimmtes Thema auf dem Warteschlangenmanager *Asien* registriert (1). Proxy-Subskriptionen für diese Subskription auf dem Warteschlangenmanager *Asien* werden an alle anderen Warteschlangenmanager im Netz (2, 3, 4) weitergeleitet.

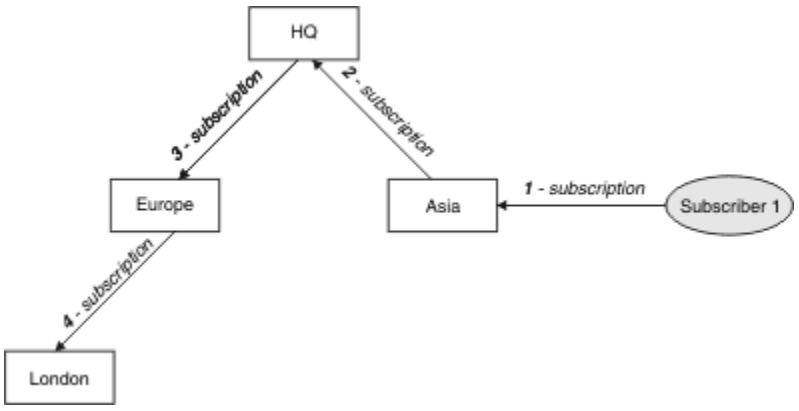


Abbildung 29. Weitergabe von Subskriptionen über ein Warteschlangenmanagernetzwerk

Ein Warteschlangenmanager konsolidiert alle erstellten Subskriptionen, unabhängig davon, ob er von lokalen Anwendungen oder von fernen Warteschlangenmanagern erstellt wird. Es erstellt Proxy-Subskriptionen für die Themen der Subskriptionen mit seinen Nachbarn, es sei denn, es ist bereits eine Proxy-Subskription vorhanden. Dies wird in der folgenden Abbildung veranschaulicht, in der *Subscriber 2* eine

Subskription für dasselbe Thema wie in Abbildung 29 auf Seite 114 auf dem Warteschlangenmanager *HQ* (5) registriert. Die Subskription für dieses Thema wird an den *Asia* -Warteschlangenmanager weitergeleitet, so dass es sich bewusst ist, dass Subskriptionen an anderer Stelle im Netz (6) vorhanden sind. Die Subskription wird nicht an den *Europe* -Warteschlangenmanager weitergeleitet, da bereits eine Subskription für dieses Thema registriert wurde. Weitere Informationen finden Sie in Schritt 3 in Abbildung 29 auf Seite 114.

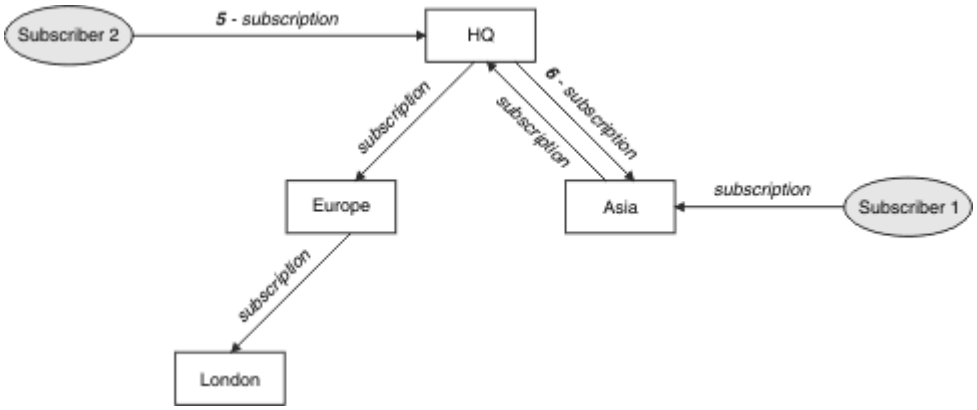


Abbildung 30. Mehrere Subskriptionen

Wenn eine Anwendung Informationen zu einem Thema veröffentlicht, leitet der empfangende WS-Manager sie standardmäßig an alle Warteschlangenmanager weiter, die gültige Subskriptionen für das Thema besitzen. Er kann ihn über einen oder mehrere temporäre Warteschlangenmanager weiterleiten. Dies wird in der folgenden Abbildung veranschaulicht, in der ein Publisher eine Veröffentlichung zu demselben Thema wie in Abbildung 30 auf Seite 115 an den Warteschlangenmanager *Europe* (7) sendet. Es ist eine Subskription für dieses Thema von *HQ* in *Europa* vorhanden, sodass die Veröffentlichung an den Warteschlangenmanager *HQ* (8) weitergeleitet wird. Es ist jedoch keine Subskription von *London* in *Europa* vorhanden (nur von *Europa* nach *London*), daher wird die Veröffentlichung nicht an den *London* -Warteschlangenmanager weitergeleitet. Der Warteschlangenmanager *HQ* sendet die Veröffentlichung direkt an *Subskribent 2* und an den Warteschlangenmanager *Asien* (9). Die Veröffentlichung wird an *Subskribent 1* von *Asien* (10) weitergeleitet.

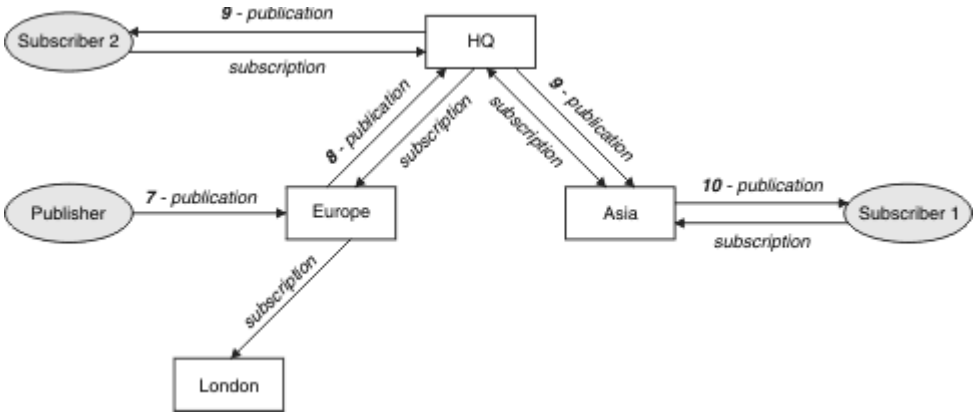


Abbildung 31. Weitergabe von Veröffentlichungen über ein Warteschlangenmanagernetzwerk

Wenn ein Warteschlangenmanager beliebige Veröffentlichungen oder Subskriptionen an einen anderen Warteschlangenmanager sendet, wird seine eigene Benutzer-ID in der Nachricht festgelegt. Wenn Sie eine Publish/Subscribe-Hierarchie verwenden und der eingehende Kanal so konfiguriert ist, dass Nachrichten mit der Berechtigung der Benutzer-ID in der Nachricht angezeigt werden, müssen Sie die Benutzer-ID des sendenden Warteschlangenmanagers berechtigen. Weitere Informationen finden Sie unter Standardbenutzer-IDs mit einer WS-Manager-Hierarchie verwenden .

Anmerkung: Wenn Sie stattdessen Publish/Subscribe-Cluster verwenden, wird die Berechtigung vom Cluster verarbeitet.

Zusammenfassung und weitere Hinweise

Eine Publish/Subscribe-Hierarchie gibt Ihnen präzise Kontrolle über die Beziehung zwischen WS-Managern. Nachdem er erstellt wurde, benötigt er wenig manuellen Eingriff für die Verwaltung. Es gibt jedoch auch bestimmte Einschränkungen auf Ihrem System:

- Die höheren Knoten in der Hierarchie, insbesondere der Stammknoten, müssen auf leistungsfähigen, hoch verfügbaren und leistungsfähigen Geräten gehostet werden. Dies liegt daran, dass mehr Veröffentlichungsverkehr durch diese Knoten fließen soll.
- Die Verfügbarkeit jedes Nicht-Leaf-WS-Managers in der Hierarchie wirkt sich auf die Fähigkeit des Netzes aus, Nachrichten von Publishern an Subskribenten auf anderen Warteschlangenmanagern zu abfließen.
- Standardmäßig werden alle Themenzeichenfolgen, die subskribiert sind, in der gesamten Hierarchie weitergegeben, und die Veröffentlichungen werden nur an ferne Warteschlangenmanager weitergegeben, die über eine Subskription für das zugeordnete Thema verfügen. Daher können schnelle Änderungen an der Gruppe von Subskriptionen zu einem Begrenzungsfaktor werden. Sie können dieses Standardverhalten ändern und stattdessen alle Publizierungsveröffentlichungen an alle Warteschlangenmanager weitergeben, wodurch die Notwendigkeit von Proxy-Subskriptionen entfällt. Weitere Informationen finden Sie im Abschnitt Subskriptionsleistung in Publish/Subscribe-Netzen.

  Anmerkung: Eine ähnliche Einschränkung gilt auch für direkte Routing-Cluster.
- Aufgrund der Vernetzung von Publish/Subscribe-Queue-Managern dauert es, bis Proxy-Subskriptionen sich über alle Knoten im Netz ausbreiten. Ferne Veröffentlichungen beginnen nicht unbedingt sofort, wenn sie sofort subskribiert werden, so dass frühzeitige Veröffentlichungen möglicherweise nicht nach einer Subskription für eine neue Themenzeichenfolge gesendet werden. Sie können die Probleme, die durch die Subskriptionsverzögerung verursacht werden, entfernen, indem alle Veröffentlichungen an alle Warteschlangenmanager weitergegeben werden, wodurch die Notwendigkeit von Proxy-Subskriptionen entfernt wird. Weitere Informationen finden Sie im Abschnitt Subskriptionsleistung in Publish/Subscribe-Netzen.

  Anmerkung: Diese Einschränkung gilt auch für direkte Routing-Cluster.
- Für eine Publish/Subscribe-Hierarchie erfordert das Hinzufügen oder Entfernen von Warteschlangenmanagern eine manuelle Konfiguration der Hierarchie, wobei die Position dieser Warteschlangenmanager und ihre Abhängigkeit von anderen WS-Managern sorgfältig berücksichtigt werden. Wenn Sie keine WS-Manager hinzufügen oder entfernen, die sich am Ende der Hierarchie befinden und deshalb keine weiteren Verzweigungen unterhalb der Hierarchie vorhanden sind, müssen Sie auch andere Warteschlangenmanager in der Hierarchie konfigurieren.

Bevor Sie eine Publish/Subscribe-Hierarchie als Routing-Mechanismus verwenden, untersuchen Sie die alternativen Ansätze, die in „Direktes Routing in Publish/Subscribe-Clustern” auf Seite 83 und „Thema Host-Routing in Publish/Subscribe-Clustern” auf Seite 88 detailliert beschrieben werden.

Verteilte Publish/Subscribe-Systemwarteschlangen

Vier Systemwarteschlangen werden von WS-Managern für Publish/Subscribe-Messaging verwendet. Sie müssen ihr Vorhandensein nur für Fehlerbestimmungs-und Kapazitätsplanungszwecke kennen.

Informationen zum Überwachen dieser Warteschlangen finden Sie im Abschnitt Produzenten und Konsumenten in Publish/Subscribe-Netzen abgleichen .

Tabelle 6. Publish/Subscribe-Systemwarteschlangen auf Multiplatforms-Plattformen

Systemwarteschlange	**Zweck**
SYSTEM.INTER.QMGR.CONTROL	Steuerwarteschlange für verteiltes Publish/Subscribe in IBM MQ

Tabelle 6. Publish/Subscribe-Systemwarteschlangen auf Multiplatforms-Plattformen (Forts.)	
Systemwarteschlange	**Zweck**
SYSTEM.INTER.QMGR.FANREQ	Eingabewarteschlange für den Fan-Out-Prozess der internen Proxy-Subskriptionen beim verteilten Publish/Subscribe in IBM MQ
SYSTEM.INTER.QMGR.PUBS	Veröffentlichungen für verteiltes Publish/Subscribe in IBM MQ
SYSTEM.HIERARCHY.STATE	Status der Hierarchiebeziehungen für verteiltes Publish/Subscribe in IBM MQ

z/OS Unter z/OS konfigurieren Sie die erforderlichen Systemobjekte beim Erstellen des Warteschlangenmanagers, indem Sie die Beispiele "CSQ4INSX", "CSQ4INSR" und "CSQ4INSG" in die Initialisierungseingabedatei "CSQINP2" einfügen. Weitere Informationen finden Sie in Task 13: Eingabedatengruppen für Initialisierung anpassen .

Die Attribute der Publish/Subscribe-Systemwarteschlangen sind in Tabelle 7 auf Seite 117 aufgeführt.

Tabelle 7. Attribute von Publish/Subscribe-Systemwarteschlangen	
Attribut	**Standardwert**
DEFPSIST	Ja
DEFSOPT	SHARED
MAXMSGL	Multi Unter Multiplatforms: Der des Parameters "MAXMSGL" im Befehl "ALTER QMGR" z/OS Unter z/OS: 4194304 (d. h. 4 MB)
MAXDEPTH	999999999
SHARE	nicht zutreffend
z/OS z/OS STGKLASSE	Dieses Attribut wird nur auf z/OS-Plattformen verwendet.

Anmerkung: Die einzige Warteschlange, die von Anwendungen gestellte Nachrichten enthält, ist SYSTEM.INTER.QMGR.PUBS. **MAXDEPTH** wird auf den Maximalwert für diese Warteschlange gesetzt, um eine temporäre Erstellung veröffentlichter Nachrichten während Ausfällen oder Zeiten übermäßiger Auslastung zu ermöglichen. Wenn der WS-Manager auf einem System ausgeführt wird, auf dem die Warteschlangenlänge nicht enthalten sein konnte, sollte dies angepasst werden.

Zugehörige Tasks

Fehlerbehebung bei Problemen mit verteiltem Publish/Subscribe

Fehler in verteilten Publish/Subscribe-Systemwarteschlangen

Fehler können auftreten, wenn verteilte Publish/Subscribe-WS-Manager-Warteschlangen nicht verfügbar sind. Dies wirkt sich auf die Weitergabe von Subskriptionswissen über das Publish/Subscribe-Netz und die Veröffentlichung auf Subskriptionen auf fernen Warteschlangenmanagern aus.

Wenn die Fan-out-Anforderungswarteschlange SYSTEM.INTER.QMGR.FANREQ nicht verfügbar ist, kann die Erstellung einer Subskription einen Fehler generieren. Fehlernachrichten werden in das Fehlerprotokoll des Warteschlangenmanagers geschrieben, wenn Proxy-Subskriptionen direkt verbundenen Warteschlangenmanagers zugestellt werden müssen.

Wenn die Statuswarteschlange SYSTEM.HIERARCHY.STATE für Hierarchiebeziehungen nicht verfügbar ist, wird eine Fehlernachricht in das Fehlerprotokoll des Warteschlangenmanagers geschrieben und die Publish/Subscribe-Engine wird in den Modus COMPAT versetzt. Verwenden Sie den Befehl DISPLAY QMGR PSMODE, um den Publish/Subscribe-Modus anzuzeigen.

Wenn eine andere der SYSTEM.INTER.QMGR-Warteschlangen nicht verfügbar ist, wird eine Fehlernachricht in das Fehlerprotokoll des Warteschlangenmanagers geschrieben. Obwohl die Funktion nicht inaktiviert ist, werden Publish/Subscribe-Nachrichten wahrscheinlich in Warteschlangen auf diesem oder fernen Warteschlangenmanagern erstellt.

Wenn die Publish/Subscribe-Systemwarteschlange oder die erforderliche Übertragungswarteschlange für einen übergeordneten, untergeordneten oder Publish/Subscribe-Cluster-WS-Manager nicht verfügbar ist, treten die folgenden Ergebnisse auf:

- Die Veröffentlichungen werden nicht zugestellt, und eine Veröffentlichungsanwendung kann einen Fehler empfangen. Ausführliche Informationen zu Fehlern, welche die Veröffentlichungsanwendung empfängt, finden Sie in den folgenden Parametern des Befehls **DEFINE TOPIC**: **PMSGDLV**, **NPMSGDLV** und **USEDLQ**.
- Empfangen von Veröffentlichungen zwischen WS-Managern wird in der Eingabewarteschlange zurückgesetzt und anschließend erneut versucht. Wenn der Rücksetzungsschwellenwert erreicht ist, werden die unzustellbare Veröffentlichungen in die Warteschlange für nicht zustellbare Nachrichten gestellt. Das Fehlerprotokoll des Warteschlangenmanagers enthält Details zu dem Problem.
- Eine unzustellbare Proxy-Subskription wird in der Warteschlange für die Fanoutanforderungswarteschlange zurückgesetzt und anschließend erneut versucht. Wenn der Rücksetzungsschwellenwert erreicht ist, wird die unzustellbare Proxy-Subskription nicht an einen verbundenen Warteschlangenmanager geliefert und in die Warteschlange für nicht zustellbare Nachrichten gestellt. Das Fehlerprotokoll des Warteschlangenmanagers enthält Details zu dem Problem, einschließlich der Details der erforderlichen erforderlichen Korrekturmaßnahmen.
- Nachrichten des Hierarchiebeziehungsprotokolls schlagen fehl und der Verbindungsstatus wird als ERROR markiert. Verwenden Sie den Befehl **DISPLAY PUBSUB**, um den Verbindungsstatus anzuzeigen.

Zugehörige Tasks

Fehlerbehebung bei Problemen mit verteiltem Publish/Subscribe

Multi Speicher-und Leistungsanforderungen auf Multiplatforms planen

Sie müssen realistische und erreichbare Speicher-und Leistungsziele für Ihr IBM MQ-System festlegen. Verwenden Sie die Links, um Informationen zu Faktoren zu finden, die sich auf die Speicherung und Leistung auf Ihrer Plattform auswirken.

Die Anforderungen sind unterschiedlich und hängen davon ab, auf welchen Systemen Sie IBM MQ einsetzen und welche Komponenten Sie verwenden möchten.

Aktuelle Informationen zu den unterstützten Hardware- und Softwareumgebungen finden Sie unter Systemvoraussetzungen für IBM MQ.

IBM MQ speichert Warteschlangenmanagerdaten im Dateisystem. Unter den folgenden Links finden Sie Informationen zur Planung und Konfiguration der Verzeichnisstrukturen für die Verwendung mit IBM MQ:

- „Unterstützung von Dateisystemen auf Multiplatforms planen” auf Seite 121
- „Voraussetzungen für gemeinsam genutzte Dateisysteme auf Multiplatforms” auf Seite 122
- „IBM MQ-Dateien in Multiplatforms gemeinsam nutzen” auf Seite 132
- Linux AIX „Verzeichnisstruktur auf Systemen mit AIX and Linux” auf Seite 135
- Windows „Verzeichnisstruktur auf Systemen mit Windows” auf Seite 144
- IBM i „Verzeichnisstruktur unter IBM i” auf Seite 148

Unter folgenden Links erhalten Sie Informationen zu Systemressourcen, gemeinsam genutzten Speicher und Prozesspriorität unter AIX and Linux:

- Linux AIX „IPC-Ressourcen für IBM MQ und UNIX System V” auf Seite 152
- AIX „gemeinsam genutzter Speicher unter AIX” auf Seite 152
- Linux AIX „Prozesspriorität von IBM MQ und UNIX” auf Seite 152

Verwenden Sie die folgenden Links, um Informationen zu Protokolldateien zu erhalten:

- „Kreisförmige oder lineare Protokollierung auf Multiplatforms auswählen” auf Seite 151
- Protokollgröße berechnen

Zugehörige Konzepte

„Planning your IBM MQ environment on z/OS” auf Seite 153
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.

Zugehörige Tasks

„IBM MQ-Architektur planen” auf Seite 5
Beachten Sie bei der Planung einer IBM MQ-Umgebung die Unterstützung, die IBM MQ für Architekturen mit einzelnen oder mehreren Warteschlangenmanagern sowie für Punkt-zu-Punkt- und Publish/Subscribe-Messaging bereitstellt. Planen Sie auch den Ressourcenbedarf und die Nutzung von Protokollierungs- und Sicherheitsfunktionen.

Zugehörige Verweise

Hardware- und Softwarevoraussetzungen unter AIX and Linux
Hardware- und Softwarevoraussetzungen unter Windows

Multi Erforderl. Plattenspeicherplatz auf Multiplatforms-Plattformen

Der Speicherbedarf für IBM MQ ist davon abhängig, welche Komponenten Sie installieren und wie viel Arbeitsspeicher Sie benötigen.

Der Plattenspeicher ist für die optionalen Komponenten, die Sie installieren möchten, erforderlich, einschließlich aller vorausgesetzten Komponenten, die sie benötigen. Der Gesamtspeicherbedarf hängt auch von der Anzahl der verwendeten Warteschlangen, der Anzahl und Größe der Nachrichten in den Warteschlangen und davon ab, ob die Nachrichten permanent sind. Sie benötigen außerdem die Archivierungskapazität auf Platte, Band oder anderen Medien sowie Speicherplatz für Ihre eigenen Anwendungsprogramme.

Die folgenden Tabellen zeigen den ungefähren Plattenspeicherplatz, der erforderlich ist, wenn Sie verschiedene Kombinationen des Produkts auf verschiedenen Plattformen installieren. (Die Werte werden auf die nächsten 5 MB aufgerundet, wobei ein MB 1.048.576 Byte beträgt.)

- LTS „Erforderlicher Plattenspeicherplatz für Long Term Support” auf Seite 119
- CD „Erforderlicher Plattenspeicherplatz für Continuous Delivery” auf Seite 120

Erforderlicher Plattenspeicherplatz für Long Term Support

LTS V 9.4.0

Tabelle 8. IBM MQ-Plattenspeicherbedarf für Multiplatforms für Long Term Support

Plattform	Clientinstallation „1” auf Seite 120	Serverinstallation „2” auf Seite 120	Vollständige Installation „3” auf Seite 120
AIX AIX	335 MB	375 MB	1810 MB

Tabelle 8. IBM MQ-Plattenspeicherbedarf für Multiplatforms für Long Term Support (Forts.)			
Plattform	**Clientinstallation** „1" auf Seite 120	**Serverinstallation** „2" auf Seite 120	**Vollständige Installation** „3" auf Seite 120
IBM i IBM i (siehe Zusätzliche Hinweise für 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 (64-Bit-Installation) „4" auf Seite 120	295 MB	425 MB	2310 MB

Anmerkungen:

1. Eine Clientinstallation umfasst die folgenden Komponenten:
   - Laufzeit
   - Client
2. Eine Serverinstallation umfasst die folgenden Komponenten:
   - Laufzeit
   - Server
3. Eine vollständige Installation enthält alle verfügbaren Komponenten.
4. Windows Nicht alle der hier aufgelisteten Komponenten können auf Windows-Systemen installiert werden; ihre Funktionalität ist gelegentlich in anderen Komponenten enthalten. Siehe IBM MQ-Features für Windows-Systeme.

Zusätzliche Hinweise für IBM i: IBM i

1. Unter IBM i können Sie den nativen Client nicht vom Server trennen. Die Zahlenangaben für den Server in der Tabelle beziehen sich auf 5724H72*BASE ohne Java und mit dem englischen Sprachenlademodul (2924). Es gibt 22 mögliche eindeutige Sprachladevorgänge.
2. Die Zahlenangaben in der Tabelle beziehen sich auf den nativen Client 5725A49 *BASE ohne Java.
3. Java- und JMS-Klassen können Servern und Clients als Bindungen hinzugefügt werden. Wenn Sie diese Features hinzufügen möchten, fügen Sie 110 MB hinzu.
4. Wenn dem Client oder Server eine Beispielquelle hinzugefügt wird, werden zusätzliche 10 MB hinzugefügt.
5. Durch das Hinzufügen von Beispielen zu Java und JMS-Klassen werden zusätzliche 5 MB benötigt.

Erforderlicher Plattenspeicherplatz für Continuous Delivery

V 9.4.0 CD

Tabelle 9. IBM MQ-Plattenspeicherbedarf für Multiplatforms für Continuous Delivery			
Plattform/CD-Release	**Clientinstallation** **„1" auf Seite 121**	**Serverinstallation** **„2" auf Seite 121**	**Vollständige Installation** **„3" auf Seite 121**
AIX **AIX**			
V 9.4.0 IBM MQ 9.4.0	355 MB	390 MB	1440 MB
Linux **Linux für x86-64 (64 Bit)**			
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 for IBM Z**			
V 9.4.0 IBM MQ 9.4.0	260 MB	290 MB	1160 MB
Windows **Windows (64-Bit-Installation)** „4" auf Seite 121			
V 9.4.0 IBM MQ 9.4.0	300 MB	425 MB	1785 MB

Anmerkungen:

1. Eine Clientinstallation umfasst die folgenden Komponenten:
   - Laufzeit
   - Client
2. Eine Serverinstallation umfasst die folgenden Komponenten:
   - Laufzeit
   - Server
3. Eine vollständige Installation enthält alle verfügbaren Komponenten.
4. Windows Nicht alle der hier aufgelisteten Komponenten können auf Windows-Systemen installiert werden; ihre Funktionalität ist gelegentlich in anderen Komponenten enthalten. Siehe IBM MQ-Features für Windows-Systeme.

Zugehörige Konzepte

Komponenten und Funktionen von IBM MQ

Multi Unterstützung von Dateisystemen auf Multiplatforms planen

WS-Manager-Daten werden im Dateisystem gespeichert. Ein Warteschlangenmanager verwendet die Sperrung von Dateisystemen, um zu verhindern, dass mehrere Instanzen eines Warteschlangenmanagers mit mehreren Instanzen gleichzeitig aktiv sind.

Gemeinsam genutzte Dateisysteme

Gemeinsam genutzte Dateisysteme ermöglichen es mehreren Systemen, gleichzeitig auf dieselbe physische Speichereinheit zuzugreifen. Eine Unterbrechung würde auftreten, wenn mehrere Systeme direkt auf dieselbe physische Speichereinheit zugegriffen haben, ohne dass die Steuerung von Sperren und gemeinsamen Zugriff erzwungen werden muss. Betriebssysteme stellen lokale Dateisysteme mit Sperrung und Steuerung des gemeinsamen Zugriffs für lokale Prozesse bereit. Netzdateisysteme stellen die Steuerung von Sperren und die Steuerung des gemeinsamen Zugriffs für verteilte Systeme bereit.

Historische, vernetzte Dateisysteme haben nicht schnell genug ausgeführt oder eine ausreichende Sperrung und Steuerung des gemeinsamen Zugriffs bereitgestellt, um die Anforderungen für die Protokollierung von Nachrichten zu erfüllen. Heute können vernetzte Dateisysteme eine gute Leistung bieten und Implementierungen zuverlässiger Netzdateisystemprotokolle, wie z. B. *RFC 3530, Network File System (NFS) Version 4, Protokoll* , erfüllen die Anforderungen für die zuverlässige Protokollierung von Nachrichten.

Gemeinsam genutzte Dateisysteme und IBM MQ

WS-Manager-Daten für einen WS-Manager mit mehreren Instanzen werden in einem gemeinsam genutzten Netzdateisystem gespeichert. Auf Systemen mit AIX, Linux, and Windows müssen die Datendateien und Protokolldateien des Warteschlangenmanagers in ein gemeinsam genutztes Netzdateisystem gestellt werden. IBM i Unter IBM i werden Journale anstelle von Protokolldateien verwendet, und Journale können nicht gemeinsam genutzt werden. Multi-Instanz-Warteschlangenmanager unter IBM i verwenden die Journalreplikation, oder umschaltbare Journale, um die Journale für mehrere Warteschlangenmanagerinstanzen gleichzeitig verfügbar zu machen.

IBM MQ verwendet Sperren, um zu verhindern, dass mehrere Instanzen desselben Multi-Instanz-Warteschlangenmanagers gleichzeitig aktiv sind. Dieselbe Sperre stellt auch sicher, dass zwei separate Warteschlangenmanager nicht versehentlich die gleiche Gruppe von WS-Manager-Datendateien verwenden können. Es kann immer nur eine Instanz eines Warteschlangenmanagers gleichzeitig gesperrt sein. Aus diesem Grund unterstützt IBM MQ Warteschlangenmanagerdaten, die in einem vernetzten Speicher gespeichert sind, auf den als gemeinsam genutztes Dateisystem zugegriffen wird.

Da nicht alle Sperrprotokolle von Netzdateisystemen stabil sind und ein Dateisystem möglicherweise für die Leistung und nicht für die Datenintegrität konfiguriert ist, müssen Sie den Befehl **amqmfsck** ausführen, um zu testen, ob ein Netzdateisystem den Zugriff auf Warteschlangenmanagerdaten und -Protokolle ordnungsgemäß steuert. Dieser Befehl ist nur auf UNIX, Linux und IBM i Systeme anwendbar. Unter Windows gibt es nur ein unterstütztes Netzdateisystem, sodass der Befehl **amqmfsck** nicht benötigt wird.

Zugehörige Tasks

„Verhalten des gemeinsam genutzten Dateisystems auf mehreren Plattformen überprüfen” auf Seite 124
Führen Sie **amqmfsck** aus, um zu prüfen, ob ein geteiltes Dateisystem auf AIX, Linux , oder IBM i die Anforderungen erfüllt, die Warteschlangenmanagerdaten eines Multiinstanzwarteschlangenmanagers. (Die einzige Voraussetzung für eine Windows-Konfiguration ist, dass SMB 3 für die Bereitstellung gemeinsam genutzten Speichers verwendet wird.)

Multi Voraussetzungen für gemeinsam genutzte Dateisysteme auf Multiplatforms

Gemeinsam genutzte Dateisysteme müssen Folgendes ermöglichen: Schreibintegrität für Daten, garantiert exklusiven Zugriff auf Dateien und die Aufhebung von Sperren für den Fall, dass kein zuverlässiges Arbeiten mit IBM MQ möglich ist.

Anforderungen an ein gemeinsam genutztes Dateisystem

Für ein zuverlässiges Zusammenwirken mit IBM MQ muss ein gemeinsam genutztes Dateisystem drei grundsätzliche Voraussetzungen erfüllen:

1. Datenschreibintegrität

Die Datenschreibintegrität wird manchmal auch als *Write through to disk on disk on flush* bezeichnet. Der Warteschlangenmanager muss in der Lage sein, mit Daten zu synchronisieren, die erfolgreich auf der physischen Einheit festgeschrieben wurden. In einem transaktionsorientierten System müssen Sie sicherstellen, dass einige Schreibvorgänge sicher festgeschrieben wurden, bevor Sie mit der anderen Verarbeitung fortfahren können.

Spezifischer IBM MQ for AIX or LinuxPlattformen verwenden die *O_SYNC*offene Oprion und der `fsync()` Systemaufruf, um explizit Schreiben auf behebbare Medien zu erzwingen und die Schreiboperation hängt davon ab, ob diese Optionen richtig funktionieren.

Achtung: Linux Sie sollten das Dateisystem mit der Option `async` anhängen, die weiterhin die Option synchroner Schreibvorgänge unterstützt und eine bessere Leistung bietet als die Option `sync` .

Es ist jedoch zu beachten, dass das Dateisystem, falls es aus Linux exportiert wurde, weiterhin mit der Option `sync` exportiert werden muss.

2. Garantiert exklusiver Zugriff auf Dateien

   Damit mehrere Warteschlangenmanager synchronisiert werden können, muss ein Mechanismus für einen Warteschlangenmanager vorhanden sein, um eine exklusive Sperre für eine Datei zu erhalten.

3. Release-Sperren bei einem Ausfall

   Wenn ein Warteschlangenmanager ausfällt oder wenn ein Kommunikationsfehler mit dem Dateisystem vorliegt, müssen die vom Warteschlangenmanager gesperrten Dateien entsperrt und anderen Prozessen zur Verfügung gestellt werden, ohne zu warten, dass der Warteschlangenmanager erneut mit dem Dateisystem verbunden wird.

Damit IBM MQ zuverlässig funktioniert, muss ein gemeinsam genutztes Dateisystem diese Anforderungen erfüllen. Ist dies nicht der Fall, werden die Daten und Protokolle des Warteschlangenmanagers beschädigt, wenn das gemeinsam genutzte Dateisystem in einer Multi-Instanz-WS-Manager-Konfiguration verwendet wird.

Bei Warteschlangenmanagern mit mehreren Instanzen unter Microsoft Windowsmuss auf den Netzspeicher über das SMB-Protokoll (Server Message Block) zugegriffen werden, das von Microsoft Windows -Netzen verwendet wird. Der SMB-Client (Server Message Block) erfüllt nicht die IBM MQ -Anforderungen für die Sperrsemantik auf anderen Plattformen als Microsoft Windows. Daher dürfen Warteschlangenmanager mit mehreren Instanzen, die auf anderen Plattformen als Microsoft Windows ausgeführt werden, Server Message Block (SMB) nicht als gemeinsam genutztes Dateisystem verwenden.

Für WS-Manager mit mehreren Instanzen auf anderen unterstützten Plattformen muss auf den Speicher durch ein Netzdateisystemprotokoll zugegriffen werden, das mit der Position "Posix-konform" kompatibel ist, und unterstützt die lease-basierte Sperrung. Network File System 4 erfüllt diese Anforderung. Ältere Dateisysteme, wie z. B. Network File System Version 3, die keinen zuverlässigen Mechanismus zum Freigeben von Sperren nach einem Fehler aufweisen, dürfen nicht mit Warteschlangenmanagern mit mehreren Instanzen verwendet werden.

Überprüfung der Anforderungen an das gemeinsam genutzte Dateisystem

Sie müssen überprüfen, ob das gemeinsam genutzte Dateisystem, das Sie verwenden möchten, diese Anforderungen erfüllt. Außerdem müssen Sie überprüfen, ob das Dateisystem ordnungsgemäß für die Zuverlässigkeit konfiguriert ist. Gemeinsam genutzte Dateisysteme bieten manchmal Konfigurationsoptionen, um die Leistung auf Kosten der Zuverlässigkeit zu verbessern.

Weitere Informationen finden Sie unter Testing statement for IBM MQ multi-instance queue manager file systems(Test- und Unterstützungsangaben für Multi-Instanz-Warteschlangenmanager in IBM MQ).

Unter normalen Umständen funktioniert IBM MQ ordnungsgemäß mit dem Attributcaching, und es ist nicht erforderlich, das Caching zu inaktivieren, z. B. indem Sie NOAC auf einem NFS-Mount festlegen. Das Attributcaching kann Probleme verursachen, wenn mehrere Dateisystemclients für Schreibzugriff auf dieselbe Datei auf dem Dateisystemserver kontendieren, da die zwischengespeicherten Attribute, die von den einzelnen Clients verwendet werden, möglicherweise nicht mit den Attributen auf dem Server

identisch sind. Ein Beispiel für Dateien, auf die auf diese Weise zugegriffen wird, sind WS-Manager-Fehlerprotokolle für einen Multi-Instanz-Warteschlangenmanager. Die WS-Manager-Fehlerprotokolle können sowohl durch eine aktive als auch durch eine Standby-Warteschlangenmanagerinstanz geschrieben werden, und die Attribute der Cachedatei können dazu führen, dass die Fehlerprotokolle größer werden als erwartet, bevor die Rollover der Dateien auftreten.

Um die Überprüfung des Dateisystems zu unterstützen, führen Sie die Task Verhalten des gemeinsam genutzten Dateisystems überprüfen aus. Diese Task prüft, ob das gemeinsam genutzte Dateisystem die Anforderungen 2 und 3erfüllt. Sie müssen die Anforderung 1 in der Dokumentation des gemeinsam genutzten Dateisystems prüfen oder indem Sie mit Protokolldaten auf der Platte experimentieren.

Plattenfehler können beim Schreiben auf Platte zu Fehlern führen, die IBM MQ als Erfassung von Fehlerdaten beim ersten Auftreten (First Failure Data Capture) meldet. Sie können das Dateisystemprüfprogramm für Ihr Betriebssystem ausführen, um das gemeinsam genutzte Dateisystem auf Plattenfehler zu überprüfen. For example:

- Linux AIX Auf AIX and Linux-Plattformen heißt das Dateisystemprüfprogramm "fsck".
- Windows Auf Windows-Plattformen heißt das Dateisystemprüfprogramm "CHKDSK" oder "SCANDISK".

Sicherheit des NFS-Servers

Anmerkungen:

- Sie können die Optionen **nosuid** oder **noexec** nicht für einen Mountpunkt verwenden, der das IBM MQ -Installationsverzeichnis enthält. Dies liegt daran, dass IBM MQ ausführbare setuid/setgid-Programme enthält, die nicht ordnungsgemäß ausgeführt werden dürfen.
- Wenn Sie WS-Manager-Daten nur auf einem Network File System-Server (NFS) einreihen, können Sie die folgenden drei Optionen mit dem Mountbefehl verwenden, um das System sicher zu machen, ohne dass die Ausführung des Warteschlangenmanagers beeinträchtigt wird:

  noexec
  Wenn Sie diese Option verwenden, stoppen Sie die Ausführung von Binärdateien auf dem NFS, wodurch verhindert wird, dass ein ferner Benutzer nicht mehr benötigten Code auf dem System ausführen kann.

  nosuid
  Wenn Sie diese Option verwenden, verhindern Sie die Verwendung der Bits "set-user-identifier" und "set-group-identifier bits", die verhindert, dass ein ferner Benutzer höhere Berechtigungen erhält.

  nodev
  Wenn Sie diese Option verwenden, stoppen Sie die Zeichen-und Blockspezial-Einheiten, die verwendet oder definiert werden, wodurch verhindert wird, dass ein ferner Benutzer aus einem chroot-Gefängnis heraus kommt.

IBM i Linux AIX *Verhalten des gemeinsam genutzten Dateisystems auf mehreren Plattformen überprüfen*

Führen Sie **amqmfsck** aus, um zu prüfen, ob ein geteiltes Dateisystem auf AIX, Linux , oder IBM i die Anforderungen erfüllt, die Warteschlangenmanagerdaten eines Multiinstanzwarteschlangenmanagers. (Die einzige Voraussetzung für eine Windows-Konfiguration ist, dass SMB 3 für die Bereitstellung gemeinsam genutzten Speichers verwendet wird.)

Vorbereitende Schritte

Sie benötigen einen Server mit vernetztem Speicher und zwei weitere Server, die mit ihm verbunden sind, auf denen IBM MQ installiert ist. Sie müssen über die Administratorberechtigung (Root) verfügen, um das Dateisystem konfigurieren zu können, und ein IBM MQ-Administrator sein, um **amqmfsck** ausführen zu können.

Informationen zu diesem Vorgang

Unter „Voraussetzungen für gemeinsam genutzte Dateisysteme auf Multiplatforms" auf Seite 122 sind die Dateisystemanforderung für die Verwendung eines gemeinsam genutzten Dateisystems mit Multi-Instanz-Warteschlangenmanagern beschrieben. In der IBM MQ-Technote Testing statement for IBM MQ multi-instance queue manager file systems werden die gemeinsam genutzten Systeme aufgeführt, mit denen IBM bereits getestet wurde. Die Prozedur in dieser Task beschreibt, wie Sie ein Dateisystem testen, um zu bewerten, ob ein nicht aufgelistete Dateisysteme die Datenintegrität aufrecht erhalten.

Der Failover eines Multi-Instanz-WS-Managers kann durch Hardware-oder Softwarefehler ausgelöst werden, einschließlich Netzproblemen, die verhindern, dass der WS-Manager in seine Daten oder Protokolldateien schreibt. Hauptsächlich sind Sie daran interessiert, Fehler auf dem Dateiserver zu verursachen. Aber Sie müssen auch Fehler auf den IBM MQ-Servern verursachen, um erfolgreich freigegebene Sperren zu testen. Damit Sie in einem gemeinsam genutzten Dateisystem vertrauen können, testen Sie alle folgenden Fehler und alle anderen Fehler, die für Ihre Umgebung spezifisch sind:

1. Das Betriebssystem auf dem Dateiserver herunterfahren, einschließlich der Synchronisierung der Platten.
2. Das Betriebssystem auf dem Dateiserver anhalten, ohne die Platten zu synchronisieren.
3. Drücken Sie die Grundstellungsschaltfläche auf jedem der Server.
4. Ausziehen des Netzkabels aus jedem der Server.
5. Ziehen Sie das Netzkabel aus jedem der Server heraus.
6. Schalten Sie die einzelnen Server aus.

Erstellen Sie das Verzeichnis im Netzspeicher, den Sie für die gemeinsame Nutzung von WS-Manager-Daten und -Protokollen verwenden werden. Der Verzeichniseigner muss ein IBM MQ-Administrator sein, oder anders gesagt, ein Mitglied der `mqm`-Gruppe in AIX and Linux sein. Der Benutzer, der die Tests ausführt, muss über die IBM MQ-Administratorberechtigung verfügen.

Verwenden Sie das Beispiel zum Exportieren und Anhängen eines Dateisystems in Creating a multi-instance queue manager on Linux oder Creating a multi-instance queue manager using journal mirroring and NetServer unter IBM i , um das Dateisystem zu konfigurieren. Unterschiedliche Dateisysteme erfordern unterschiedliche Konfigurationsschritte. Lesen Sie die Dokumentation zum Dateisystem.

Anmerkung: Führen Sie das IBM MQ MQI client -Beispielprogramm **amqsfhac** parallel zu **amqmfsck** aus, um zu demonstrieren, dass ein Warteschlangenmanager die Nachrichtenintegrität während eines Fehlers beibehält.

Vorgehensweise

Bei jedem der Prüfungen führen Sie alle Fehler in der vorherigen Liste durch, während die Dateisystemprüffunktion ausgeführt wird. Wenn Sie **amqsfhac** gleichzeitig mit **amqmfsck** ausführen möchten, müssen Sie die Task „amqsfhac zum Testen der Nachrichtenintegrität ausführen" auf Seite 130 parallel mit dieser Task ausführen.

1. Hängen Sie das exportierte Verzeichnis auf den beiden IBM MQ-Servern an.

   Erstellen Sie auf dem Dateisystemserver ein gemeinsam genutzes Verzeichnis `shared` und ein Unterverzeichnis zum Speichern der Daten für Multi-Instanz-Warteschlangenmanager, `qmdata`. Ein Beispiel für die Einrichtung eines gemeinsam genutzten Verzeichnisses für Warteschlangenmanager mit mehreren Instanzen unter Linuxfinden Sie im Abschnitt Warteschlangenmanager mit mehreren Instanzen unter Linux erstellen .
2. Überprüfen Sie das Verhalten des Basisdateisystems.

   Führen Sie auf einem der IBM MQ-Server das Dateisystemprüfprogramm ohne Parameter aus.

   Auf IBM MQ-Server 1:

   ```
   amqmfsck /shared/qmdata
   ```
3. Prüfen Sie das gleichzeitige Schreiben von beiden IBM MQ-Servern in dasselbe Verzeichnis.

Führen Sie das Dateisystemprüfprogramm auf beiden IBM MQ-Servern gleichzeitig mit der Option `-c` aus.

Auf IBM MQ-Server 1:

```
amqmfsck -c /shared/qmdata
```

Auf IBM MQ-Server 2:

```
amqmfsck -c /shared/qmdata
```

4. Prüfen Sie auf beiden IBM MQ-Servern das Warten auf Sperren und deren Freigabe.

   Führen Sie das Dateisystemprüfprogramm auf beiden IBM MQ-Servern gleichzeitig mit der Option `-w` aus.

   Auf IBM MQ-Server 1:

   ```
   amqmfsck -w /shared/qmdata
   ```

   Auf IBM MQ-Server 2:

   ```
   amqmfsck -w /shared/qmdata
   ```

5. Überprüfen Sie die Datenintegrität.

   a) Formatieren Sie die Testdatei.

   Erstellen Sie eine große Datei in dem Verzeichnis, das getestet wird. Die Datei wird so formatiert, dass die nachfolgenden Phasen erfolgreich abgeschlossen werden können. Die Datei muss groß genug sein, dass genügend Zeit vorhanden ist, um die zweite Phase zu unterbrechen, um die Funktionsübernahme zu simulieren. Versuchen Sie, den Standardwert von 262144 Seiten (1 GB) zu verwenden. Das Programm reduziert diese Standardeinstellung bei langsamen Dateisystemen automatisch, so dass die Formatierung in ca. 60 Sekunden abgeschlossen wird.

   Auf IBM MQ-Server 1:

   ```
   amqmfsck -f /shared/qmdata
   ```

   Der Server antwortet mit den folgenden Nachrichten:

   ```
   Formatting test file for data integrity test.

   Test file formatted with 262144 pages of data.
   ```

   b) Schreiben Sie Daten mit Hilfe des Dateisystemprüfers in die Testdatei, und verursachen Sie einen Fehler.

   Führen Sie das Testprogramm auf zwei Servern zur gleichen Zeit aus. Starten Sie das Testprogramm auf dem Server, auf dem der Fehler auftreten wird, und starten Sie dann das Testprogramm auf dem Server, das den Fehler überleben wird. Ursache des Fehlers, den Sie untersuchen.

   Das erste Testprogramm stoppt mit einer Fehlernachricht. Das zweite Testprogramm ruft die Sperre für die Testdatei ab und schreibt Daten in die Testdatei, in der das erste Testprogramm abgelassen wurde. Lassen Sie das zweite Testprogramm zum Abschluss führen.

Tabelle 10. Datenintegritätsprüfung auf zwei Servern zur gleichen Zeit ausführen

IBM MQ-Server 1	**IBM MQ-Server 2**
`amqmfsck -a /shared/qmdata`	

`Please start this program on a second ma chine with the same parameters.`
 	
`File lock acquired.`
 	
`Start a second copy of this program with the same parameters on another ser ver.`
 	
`Writing data into test file.`
 	
`To increase the effectiveness of the test, interrupt the writing by ending the pro cess, temporarily breaking the network connecti on to the networked storage, rebooting the server or turning off the power.`
	`amqmfsck -a /shared/qmdata`
	
	`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 in tegrity.`

Der Zeitpunkt des Tests richtet sich nach dem Verhalten des Dateisystems. Beispielsweise dauert es in der Regel 30 bis 90 Sekunden, wenn ein Dateisystem die Dateisperren freigibt, die durch das erste Programm nach einem Stromausfall erhalten wurden. Wenn Sie zu wenig Zeit haben, um den Fehler einzuführen, bevor das erste Testprogramm die Datei gefüllt hat, verwenden Sie die Option `-x` von **amqmfsck** , um die Testdatei zu löschen. Testen Sie den Test ab dem Start mit einer größeren Testdatei.

c) Überprüfen Sie die Integrität der Daten in der Testdatei.

Auf IBM MQ-Server 2:

```
amqmfsck -i /shared/qmdata
```

Der Server antwortet mit den folgenden Nachrichten:

```
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. Löschen Sie die Testdateien.

   Auf IBM MQ-Server 2:

```
amqmfsck -x /shared/qmdata
Test files deleted.
```

   Der Server antwortet mit der Nachricht:

```
Test files deleted.
```

Ergebnisse

Das Programm gibt den Exit-Code 0 zurück, wenn die Tests erfolgreich abgeschlossen wurden, und andernfalls nicht null.

Beispiele

Die erste Gruppe von drei Beispielen zeigt den Befehl, der die minimale Ausgabe erzeugt.

Erfolgreicher Test der Basisdateisperrung auf einem Server

```
> amqmfsck /shared/qmdata
The tests on the directory completed successfully.
```

Fehlgeschlagener Test der Basisdateisperrung auf einem Server

```
> amqmfsck /shared/qmdata
AMQ6245: Error Calling 'write()[2]' on file '/shared/qmdata/amqmfsck.lck' error '2'.
```

Erfolgreicher Test der Sperrung auf zwei Servern

Tabelle 11. Erfolgreiches Sperren auf zwei Servern

IBM MQ-Server 1	IBM MQ-Server 2

`> 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.`	

Tabelle 11. Erfolgreiches Sperren auf zwei Servern (Forts.)	
IBM MQ-Server 1	**IBM MQ-Server 2**
	Lock acquired. The tests on the directory completed successful□ ly

In der zweiten Gruppe von drei Beispielen werden dieselben Befehle im ausführlichen Modus angezeigt.

Erfolgreicher Test der Basisdateisperrung auf einem Server

```
> 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.
```

Fehlgeschlagener Test der Basisdateisperrung auf einem Server

```
> 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).
```

Erfolgreicher Test der Sperrung auf zwei Servern

Tabelle 12. Erfolgreiches Sperren auf zwei Servern-Modus 'verbose'

IBM MQ-Server 1	IBM MQ-Server 2

`> amqmfsck -wv /shared/qmdata`
`Calling 'stat("/shared/qmdata")'`
`Calling 'fd = open("/shared/qmda`
`ta/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/qmda`
	`ta/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...`

`[ Return pressed ]`
`Calling 'close(fd)'`
`Lock released.`	

	`Calling 'fcntl(fd, F_SETLK, F_WRLCK)'`
	`Lock acquired.`
	`The tests on the directory completed successful`
	`ly`

Zugehörige Verweise

Beispielprogramme zur Hochverfügbarkeit

Multi

amqsfhac zum Testen der Nachrichtenintegrität ausführen

Führen Sie das IBM MQ MQI client -Beispielprogramm **amqsfhac** parallel zu **amqmfsck** aus, um zu demonstrieren, dass ein Warteschlangenmanager die Nachrichtenintegrität während eines Fehlers beibehält.

Vorbereitende Schritte

Für diesen Test benötigen Sie vier Server. Zwei Server für den Multi-Instanz-Warteschlangenmanager, einen für das Dateisystem und einen, um **amqsfhac** als IBM MQ MQI client-Anwendung auszuführen.

Führen Sie Schritt „1” auf Seite 125 unter „Verhalten des gemeinsam genutzten Dateisystems auf mehreren Plattformen überprüfen” auf Seite 124 aus, um das Dateisystem für einen Multi-Instanz-Warteschlangenmanager einzurichten.

Informationen zu diesem Vorgang

Das IBM MQ MQI client-Beispielprogramm **amqsfhac** überprüft, ob ein Warteschlangenmanager, der den Netzspeicher verwendet, die Datenintegrität nach einem Fehler aufrechterhält. Führen Sie **amqsfhac** parallel zu **amqmfsck** aus, um zu demonstrieren, dass ein Warteschlangenmanager die Nachrichtenintegrität während eines Fehlers aufrechterhält.

Vorgehensweise

1. Erstellen Sie einen Warteschlangenmanager mit mehreren Instanzen auf einem anderen Server, QM1, und verwenden Sie dabei das Dateisystem, das Sie in Schritt „1” auf Seite 125 in Vorgehensweiseerstellt haben.

   Siehe Multi-Instanz-WS-Manager erstellen .
2. Starten Sie den Warteschlangenmanager auf beiden Servern, die ihn hoch verfügbar machen.

   Auf Server 1:

   ```
   strmqm -x QM1
   ```

   Auf Server 2:

   ```
   strmqm -x QM1
   ```
3. Richten Sie die Clientverbindung für die Ausführung von **amqsfhac**ein.
   a) Führen Sie die Vorgehensweise im Abschnitt *IBM MQ-Installation überprüfen* für die Plattform oder Plattformen aus, die in Ihrem Unternehmen zum einrichten einer Clientverbindung verwendet wird, oder die Beispielscripts im Abschnitt Clientverbindungen konfigurieren.
   b) Ändern Sie den Clientkanal so, dass zwei IP-Adressen vorhanden sind, die den beiden Servern entsprechen, auf der QM1 ausgeführt wird.

      Ändern Sie im Beispielscript Folgendes:

      ```
      DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
      CONNAME('LOCALHOST(2345)') QMNAME(QM1) REPLACE
      ```

      Zu:

      ```
      DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
      CONNAME('server1(2345),server2(2345)') QMNAME(QM1) REPLACE
      ```

      Dabei sind server1 und server2 die Hostnamen der beiden Server, und 2345 ist der Port, an dem der Kanal-Listener empfangsbereit ist. Gewöhnlich ist dies der Standardwert 1414. Sie können 1414 mit der Standard-Listener-Konfiguration verwenden.
4. Erstellen Sie zwei lokale Warteschlangen unter QM1 für den Test.
   Führen Sie das folgende MQSC-Script aus:

   ```
   DEFINE QLOCAL(TARGETQ) REPLACE
   DEFINE QLOCAL(SIDEQ) REPLACE
   ```
5. Testen Sie die Konfiguration mit **amqsfhac**.

   ```
   amqsfhac QM1 TARGETQ SIDEQ 2 2 2
   ```
6. Testen Sie die Nachrichtenintegrität, während Sie die Integrität des Dateisystems testen.

   Führen Sie **amqsfhac** während Schritt „5” auf Seite 126 von „Verhalten des gemeinsam genutzten Dateisystems auf mehreren Plattformen überprüfen” auf Seite 124aus.

   ```
   amqsfhac QM1 TARGETQ SIDEQ 10 20 0
   ```

   Wenn Sie die aktive Warteschlangenmanagerinstanz stoppen, stellt **amqsfhac** die Verbindung zur anderen Warteschlangenmanagerinstanz wieder her, sobald sie aktiv geworden ist. Starten Sie die gestoppte WS-Manager-Instanz erneut, so dass Sie den Fehler beim nächsten Test rückgängig machen können. Sie müssen wahrscheinlich die Anzahl der Iterationen auf der Basis des Experiments mit

Ihrer Umgebung erhöhen, damit das Testprogramm genügend Zeit für die Übernahme von Failover ausgeführt wird.

Ergebnisse

Nachfolgend wird ein Beispiel für die Ausführung von **amqsfhac** in Schritt „6” auf Seite 131 gezeigt. In diesem Beispiel ist der Test ein Erfolg.

```
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
```

Wenn der Test ein Problem festgestellt hat, würde die Ausgabe den Fehler melden. In einigen Testläufen kann `MQRC_CALL_INTERRUPTED` möglicherweise "`Resolving to backed out`" melden. Es macht keinen Unterschied zum Ergebnis. Das Ergebnis hängt davon ab, ob der Schreibzugriff auf die Platte durch den Netzdateispeicher vor oder nach dem Fehlschlagen der Platte festgeschrieben wurde.

Zugehörige Verweise

amqmfsck (Dateisystemprüfung)

Beispielprogramme zur Hochverfügbarkeit

Multi IBM MQ-Dateien in Multiplatforms gemeinsam nutzen

Auf einige IBM MQ-Dateien wird ausschließlich über einen aktiven Warteschlangenmanager zugegriffen, während andere Dateien gemeinsam genutzt werden.

Bei IBM MQ-Dateien wird zwischen Programmdateien und Datendateien unterschieden. Programmdateien werden normalerweise lokal auf jedem Server installiert, auf dem IBM MQ ausgeführt wird. Warteschlangenmanager nutzen den Zugriff auf Datendateien und Verzeichnisse im Standarddatenverzeichnis gemeinsam. sie benötigen exklusiven Zugriff auf die Verzeichnisstrukturen ihres eigenen Warteschlangenamanagers, die sich jeweils in den Verzeichnissen `qmgrs` und `log`, die in Abbildung 32 auf Seite 133 dargestellt sind.

Abbildung 32 auf Seite 133 ist eine Übersicht der IBM MQ-Verzeichnisstruktur. Sie zeigt die Verzeichnisse an, die von den WS-Managern gemeinsam genutzt werden können und die ferne ausgeführt werden können Die Details variieren je nach Plattform. Die gepunkteten Linien geben konfigurierbare Pfade an.

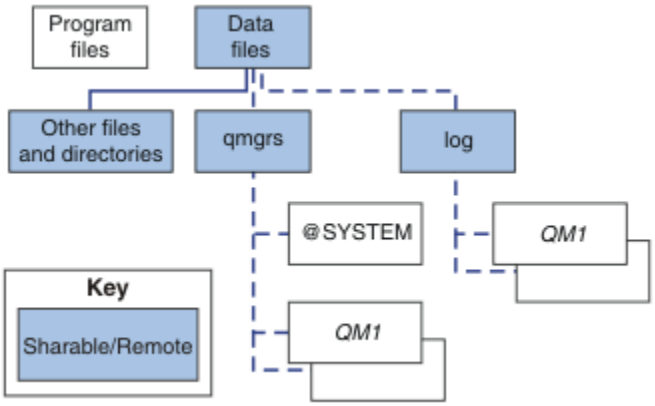


Abbildung 32. Übersicht der IBM MQ-Verzeichnisstruktur

Programm- dateien

Das Verzeichnis der Programmdateien wird in der Regel an der Standardposition belassen, ist lokal und wird von allen WS-Managern auf dem Server gemeinsam genutzt.

Daten- dateien

Das Datenverzeichnis ist in der Regel lokal in der Standardposition, `/var/mqm` auf AIX and Linux-Systemen und kann bei der Installation unter Windows konfiguriert werden. Sie wird von WS-Managern gemeinsam genutzt. Sie können eine ferne Position als Standardposition festlegen, es darf jedoch keine Position sein, die von verschiedenen IBM MQ-Installationen gemeinsam genutzt wird. Das Attribut `DefaultPrefix` in der IBM MQ-Konfiguration verweist auf diesen Pfad.

`qmgrs`

Es gibt zwei alternative Möglichkeiten, die Position der Warteschlangenmanagerdaten anzugeben:

Attribut `Prefix` verwenden

Das Attribut **`Prefix`** gibt die Position des Verzeichnisses `qmgrs` an. IBM MQ erstellt für den Warteschlangenmanager im Verzeichnis `qmgrs` ein Unterverzeichnis mit dem Namen des Warteschlangenmanagers.

Das Attribut **`Prefix`** befindet sich in der ZeilengruppeQueueManager der Datei `mqs.ini` und wird aus dem Wert des Attributs **`DefaultPrefix`** der Zeilengruppe All Queue Managersübernommen. Standardmäßig verwenden die Warteschlangenmanager standardmäßig dasselbe `qmgrs` -Verzeichnis, weil sie sich auf die Verwaltung von Verwaltungsaufgaben nicht teilen.

Wenn Sie die Position des Verzeichnisses `qmgrs` für einen Warteschlangenmanager ändern, müssen Sie den Wert des Attributs **`Prefix`** ändern.

Das Attribut **`Prefix`** für das Verzeichnis `QM1` in Abbildung 32 auf Seite 133 für eine AIX and Linux -Plattform lautet wie folgt:

```
Prefix=/var/mqm
```

Attribut `DataPath` verwenden

Das Attribut **`DataPath`** gibt die Position des Datenverzeichnisses des Warteschlangenmanagers an.

Das Attribut **`DataPath`** gibt den vollständigen Pfad einschließlich des Namens des Datenverzeichnisses des Warteschlangenmanagers an. Das Attribut **`DataPath`** unterscheidet sich vom Attribut **`Prefix`** , das einen unvollständigen Pfad zum Datenverzeichnis des Warteschlangenmanagers angibt.

Das Attribut **DataPath** , falls angegeben, befindet sich in der Zeilengruppe QueueManager der Datei `mqs.ini` . Wenn sie angegeben wurde, hat sie Vorrang vor allen Werten im Attribut **Prefix** .

Wenn Sie die Position des Datenverzeichnisses des Warteschlangenmanagers für einen WS-Manager ändern, müssen Sie den Wert des Attributs `DataPath` ändern.

Das Attribut `DataPath` für das Verzeichnis `QM1` in Abbildung 32 auf Seite 133für eine Linux -oder AIX -Plattform lautet wie folgt:

```
DataPath=/var/mqm/qmgrs/QM1
```

log

Das Protokollverzeichnis wird für jeden Warteschlangenmanager in der Protokollzeilengruppe in der Warteschlangenmanagerkonfiguration separat angegeben. Die Konfiguration des WS-Managers befindet sich in `qm.ini`.

***DataPath*/*QmgrName*/@IPCC-Unterverzeichnisse**

Die Unterverzeichnisse von *DataPath*/*QmgrName*/`@IPCC` befinden sich im Pfad für gemeinsam genutzte Verzeichnisse. Sie werden verwendet, um den Verzeichnispfad für IPC-Dateisystemobjekte zu erstellen. Sie müssen den Namensbereich eines Warteschlangenmanagers unterscheiden, wenn ein Warteschlangenmanager von mehreren Systemen gemeinsam genutzt wird.

Die IPC-Dateisystemobjekte müssen vom System unterschieden werden. Für jedes System, auf dem der Warteschlangenmanager ausgeführt wird, wird dem Verzeichnispfad ein Unterverzeichnis hinzugefügt (siehe Abbildung 33 auf Seite 134).

```
DataPath/QmgrName/@IPCC/esem/myHostName/
```

Abbildung 33. Beispiel für ein IPC-Unterverzeichnis

myHostName ist bis zu den ersten 20 Zeichen des vom Betriebssystem ausgegebenen Hostnamens. Auf einigen Systemen kann der Hostname bis zu 64 Zeichen lang sein, bevor er abgeschnitten wird. Der generierte Wert von *myHostName* kann aus zwei Gründen ein Problem verursachen:

1. Die ersten 20 Zeichen sind nicht eindeutig.
2. Der Hostname wird von einem DHCP-Algorithmus generiert, der nicht immer denselben Hostnamen einem System zuordnet.

Legen Sie in diesen Fällen *myHostName* mithilfe der Umgebungsvariablen **MQS_IPC_HOST**fest (siehe Abbildung 34 auf Seite 134).

```
export MQS_IPC_HOST= myHostName
```

*Abbildung 34. Beispiel für das Festlegen von **MQS_IPC_HOST***

Andere Dateien und Verzeichnisse

Andere Dateien und Verzeichnisse, wie z. B. das Verzeichnis mit den Tracedateien und das allgemeine Fehlerprotokoll, werden normalerweise gemeinsam genutzt und auf dem lokalen Dateisystem gespeichert.

Mit Unterstützung gemeinsam genutzter Dateisysteme verwaltet IBM MQ den exklusiven Zugriff auf diese Dateien mithilfe von Dateisystemsperren. Eine Dateisystemsperre erlaubt es nur einer Instanz eines bestimmten Warteschlangenmanagers, aktiv zu sein.

Wenn Sie die erste Instanz eines bestimmten Warteschlangenmanagers starten, wird das Eigentumsrecht an dem Warteschlangenmanager-Verzeichnis des Warteschlangenmanagers angezeigt. Wenn Sie eine zweite Instanz starten, kann sie nur dann das Eigentumsrecht übernehmen, wenn die erste Instanz gestoppt wurde. Wenn der erste Warteschlangenmanager noch aktiv ist, kann die zweite Instanz nicht gest-

artet werden, und es wird gemeldet, dass der Warteschlangenmanager an anderer Stelle ausgeführt wird. Wenn der erste Warteschlangenmanager gestoppt wurde, übernimmt der zweite Warteschlangenmanager das Eigentumsrecht an den WS-Manager-Dateien und wird zum aktiven Warteschlangenmanager.

Sie können die Prozedur des zweiten Warteschlangenmanagers, der von der ersten übernommen wird, automatisieren. Starten Sie den ersten Warteschlangenmanager mit der Option `strmqm -x` , der es einem anderen WS-Manager ermöglicht, von diesem Warteschlangenmanager zu übernehmen. Der zweite WS-Manager wartet dann, bis die WS-Manager-Dateien entsperrt sind, bevor er versucht, das Eigentumsrecht an den WS-Manager-Dateien zu übernehmen, und startet.

Linux AIX Verzeichnisstruktur auf Systemen mit AIX and Linux

Die IBM MQ-Verzeichnisstruktur auf Systemen mit AIX and Linux kann unterschiedlichen Dateisystemen zugeordnet werden, um die Verwaltung zu vereinfachen, die Leistung zu erhöhen oder die Zuverlässigkeit zu verbessern.

Nutzen Sie die flexible Verzeichnisstruktur von IBM MQ, um gemeinsam genutzte Dateisysteme für die Ausführung von Multi-Instanz-Warteschlangenmanagern zu verwenden.

Verwenden Sie den Befehl **crtmqm** *QM1* , um die in Abbildung 35 auf Seite 135 gezeigte Verzeichnisstruktur zu erstellen, wobei R das Release des Produkts ist. Dies ist eine typische Verzeichnisstruktur für einen Warteschlangenmanager, der auf einem IBM MQ -System erstellt wurde. Einige Verzeichnisse, Dateien und .ini-Attributeinstellungen werden aus Gründen der Übersichtlichkeit weggelassen, und ein anderer Name des WS-Managers kann durch das Mangeln geändert werden. Die Namen der Dateisysteme hängen von unterschiedlichen Systemen ab.

In einer Standardinstallation zeigen alle Warteschlangenmanager, die Sie erstellen, auf allgemeine `log`- und `qmgrs`-Verzeichnisse auf dem lokalen Dateisystem. In einer Konfiguration mit mehreren Instanzen befinden sich die Verzeichnisse `log` und `qmgrs` in einem Netzdateisystem, das gemeinsam mit einer anderen Installation von IBM MQ gemeinsam genutzt wird.

Abbildung 35 auf Seite 135 zeigt die Standardkonfiguration für IBM MQ V7.R unter AIX, wobei R die Releasenummer des Produkts ist. Beispiele für andere Konfigurationen mit mehreren Instanzen finden Sie unter „Beispiele für Verzeichniskonfigurationen auf Systemen mit AIX and Linux” auf Seite 140.

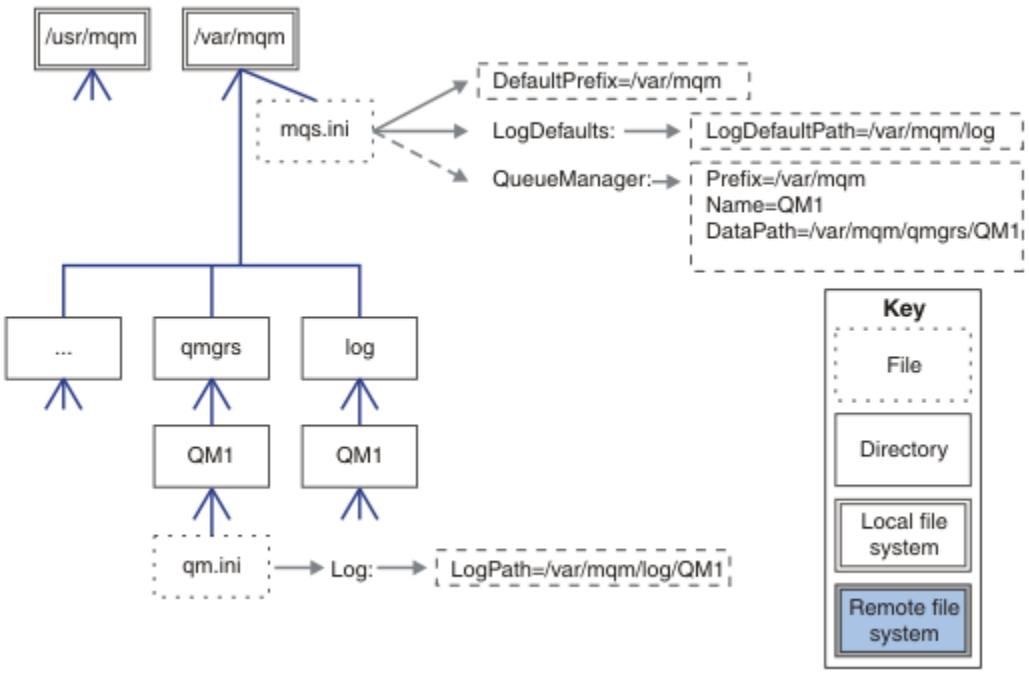


Abbildung 35. Beispiel für eine standardmäßige IBM MQ-Dateistruktur für Systeme mit AIX and Linux

Das Produkt wird standardmäßig in /usr/mqm unter AIX installiert, bei anderen Systemen unter /opt/mqm. Die Arbeitsverzeichnisse werden in das Verzeichnis /var/mqm installiert.

Anmerkung: Wenn Sie das /var/mqm-Dateisystem vor der Installation von IBM MQ erstellt haben, sollten Sie sicherstellen, dass der Benutzer 'mqm' über vollständige Verzeichnisberechtigungen verfügt, z. B. im Dateimodus '755'.

Anmerkung: Das /var/mqm/errors-Verzeichnis sollte ein separates Dateisystem sein, um zu verhindern, dass FFDCs, die vom Warteschlangenmanager erstellt werden, das Dateisystem füllen, das /var/mqm enthält.

Weitere Informationen finden Sie unter Dateisysteme auf Systemen mit AIX and Linux erstellen.

Die Verzeichnisse log und qmgrs werden an ihren Standardpositionen angezeigt, die durch die Standardwerte der Attribute LogDefaultPath und Standardpräfix in der mqs.ini-Datei definiert sind. Wenn ein Warteschlangenmanager erstellt wird, wird standardmäßig das Datenverzeichnis des Warteschlangenmanagers in *DefaultPrefix*/qmgrs und das Verzeichnis für die Protokolldatei in *LogDefaultPath*/log erstellt. LogDefaultPath und DefaultPrefix wirken sich nur auf die Erstellung von Warteschlangenmanagern und Protokolldateien aus. Die tatsächliche Position eines WS-Manager-Verzeichnisses wird in der Datei mqs.ini gespeichert, die Position des Protokolldateiverzeichnisses wird in der Datei qm.ini gespeichert.

Das Protokolldateiverzeichnis für einen Warteschlangenmanager ist in der Datei qm.ini im Attribut Protokollpfad definiert. Verwenden Sie die Option -ld im Befehl **crtmqm** , um das Attribut LogPath für einen Warteschlangenmanager festzulegen, z. B. **crtmqm** -ld *LogPath* QM1 . Wenn Sie den Parameter ld nicht angeben, wird stattdessen der Wert von LogDefaultPath verwendet.

Das Datenverzeichnis des Warteschlangenmanagers wird im Attribut Datenpfad in der Zeilengruppe QueueManager in der Datei mqs.ini definiert. Verwenden Sie die Option -md im Befehl **crtmqm** , um DataPath für einen Warteschlangenmanager festzulegen, z. B. **crtmqm** - md *DataPath* QM1 . Wenn Sie den Parameter md nicht angeben, wird stattdessen der Wert des Attributs DefaultPrefix oder Prefix verwendet. Präfix hat Vorrang vor DefaultPrefix .

In der Regel erstellen Sie QM1 , indem Sie sowohl die Protokoll-als auch die Datenverzeichnisse in einem einzigen Befehl angeben.

```
 crtmqm
-md DataPath -ld
LogPath QM1
```

Sie können die Position eines WS-Manager-Protokolls und der Datenverzeichnisse eines vorhandenen Warteschlangenmanagers ändern, indem Sie die Attribute Datenpfad und Protokollpfad in der Datei qm.ini bearbeiten, wenn der Warteschlangenmanager angehalten ist.

Der Pfad zum Verzeichnis errors ist wie die Pfade zu allen anderen Verzeichnissen in /var/mqm nicht änderbar. Die Verzeichnisse können jedoch auf verschiedenen Dateisystemen angehängt werden oder symbolisch mit verschiedenen Verzeichnissen verknüpft sein.

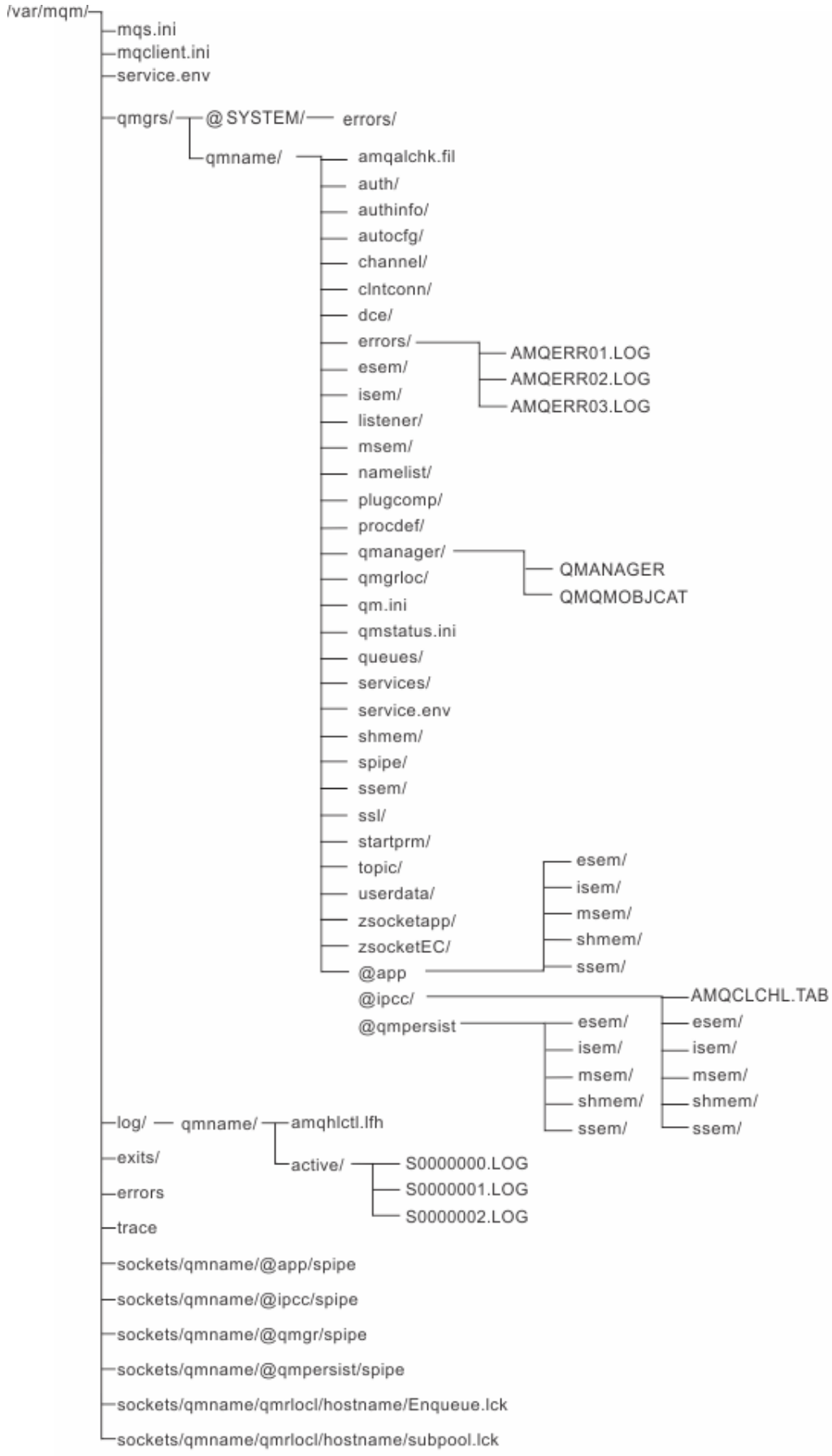
Linux AIX *Verzeichnisinhalt auf Systemen mit AIX and Linux*

Inhalt der Verzeichnisse, die einem WS-Manager zugeordnet sind.

Informationen zur Position der Produktdateien finden Sie unter Installationsposition auswählen.

Informationen zu alternativen Verzeichniskonfigurationen finden Sie unter „Unterstützung von Dateisystemen auf Multiplatforms planen” auf Seite 121.

Die folgende Verzeichnisstruktur ist repräsentativ für IBM MQ , wenn ein Warteschlangenmanager seit einiger Zeit verwendet wird. Die tatsächliche Struktur, von der Sie abhängig sind, hängt davon ab, welche Operationen auf dem Warteschlangenmanager ausgeführt wurden.


/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/

Das Verzeichnis /var/mqm enthält Konfigurationsdateien und Ausgabeverzeichnisse, die für eine IBM MQ-Installation als Ganzes gelten, und nicht für einen einzelnen Warteschlangenmanager.

Tabelle 13. Dokumentierter Inhalt des /var/mqm-Verzeichnisses unter AIX and Linux

Verzeichnis-oder Dateiname	Inhalt
mqs.ini	Für die ganze IBM MQ-Installation geltende Konfigurationsdatei, die gelesen wird, wenn ein Warteschlangenmanager gestartet wird. Der Dateipfad kann mit der Umgebungsvariablen **AMQ_MQS_INI_LOCATION** geändert werden. Stellen Sie sicher, dass dies in der Shell festgelegt und exportiert wird, in der der **strmqm** -Befehl ausgeführt wird.
mqclient.ini	Standardmäßige Clientkonfigurationsdatei, die von IBM MQ MQI client-Programmen gelesen wird. Der Dateipfad kann mit der Umgebungsvariablen **MQCLNTCF** geändert werden.
service.env	Enthält Umgebungsvariablen des Maschinenbereichs für einen Serviceprozess. Dateipfad wurde korrigiert.
Fehler/	Systemweit geltende Fehlerprotokolle und FFST-Dateien. Verzeichnispfad wurde korrigiert. Siehe auch FFST: IBM MQ for UNIX and Linux.
Sockets/	Enthält nur Informationen zu jedem Warteschlangenmanager für die Systemverwendung.
Trace/	Tracedateien. Verzeichnispfad wurde korrigiert.
web/	Verzeichnis 'mqweb server'.
exits/	Standardverzeichnis, das Benutzerkanalexitprogramme enthält. Die Position kann in ApiExit-Zeilengruppen in der Datei 'mqs.ini' geändert werden.
exits64/	

/var/mqm/qmgrs/*qmname*/

/var/mqm/qmgrs/*qmname*/ enthält Verzeichnisse und Dateien für einen Warteschlangenmanager. Das Verzeichnis ist für exklusiven Zugriff durch die aktive WS-Manager-Instanz gesperrt. Der Verzeichnispfad kann direkt in der Datei mqs.ini oder mithilfe der Option **md** des Befehls **crtmqm** geändert werden.

Tabelle 14. Dokumentierter Inhalt des /var/mqm/qmgrs/qmname-Verzeichnisses unter AIX and Linux

Verzeichnis-oder Dateiname	Inhalt
qm.ini	Warteschlangenmanagerkonfigurationsdatei, lesen, wenn ein Warteschlangenmanager gestartet wird.
Fehler/	Fehlerprotokolle des Warteschlangenmanagers. *qmname* = @system enthält kanalbezogene Nachrichten für einen unbekannten oder nicht verfügbaren WS-Manager.

Tabelle 14. Dokumentierter Inhalt des /var/mqm/qmgrs/qmname-Verzeichnisses unter AIX and Linux (Forts.)	
Verzeichnis-oder Dateiname	**Inhalt**
@ipcc/ AMQCLCHL.TAB	Standardmäßige Steuertabelle für den Clientkanal, die vom IBM MQ-Server erstellt und von IBM MQ MQI client-Programmen gelesen wird. Der Dateipfad kann mit den Umgebungsvariablen **MQCHLLIB** und **MQCHLTAB** geändert werden.
qmanager	WS-Manager-Objektdatei: QMANAGER Objektkatalog des WS-Managers: QMQMOBJCAT
authinfo/	Jedem Objekt, das im Warteschlangenmanager definiert ist, wird eine Datei in diesen Verzeichnissen zugeordnet. Der Dateiname stimmt ungefähr mit dem Definitionsnamen überein; siehe IBM MQ-Dateinamen verstehen.
Kanal/	
clntconn/	
Empfangsprogramm/	
namelist/	
procdef/	
Warteschlangen/	
Dienstleistungen/	
Themen/	
...	Andere Verzeichnisse, die von IBM MQ verwendet werden, z. B. @ipcc, und nur von IBM MQ geändert werden sollen.
Benutzerdaten/	Kann verwendet werden, um den persistenten Status von Anwendungen zu speichern (kann vom RDQM beim Verschieben von Warteschlangenmanager an verschiedene Knoten verwendet werden - siehe Persistenten Anwendungsstatus speichern.)
***DataPath*\autocfg**	Wird für die automatische Konfiguration verwendet

/var/mqm/log/*qmname*/

/var/mqm/log/*qmname*/ enthält die WS-Manager-Protokolldateien. Das Verzeichnis ist für exklusiven Zugriff durch die aktive WS-Manager-Instanz gesperrt. Der Verzeichnispfad kann in der Datei qm.ini oder mithilfe der Option **ld** des Befehls **crtmqm** geändert werden.

Tabelle 15. Dokumentierter Inhalt des /var/mqm/log/qmname-Verzeichnisses unter AIX and Linux	
Verzeichnis-oder Dateiname	**Inhalt**
amqhlctl.lfh	Protokollsteuerdatei.
Aktiv/	Dieses Verzeichnis enthält die Protokolldateien S0000000.LOG, S0000001.LOG, S0000002.LOG und so weiter.

/opt/mqm

/opt/mqm ist standardmäßig das Installationsverzeichnis auf den meisten Plattformen. Weitere Informationen dazu, wie viel Speicherplatz für das Installationsverzeichnis auf der Plattform oder den Plattformen benötigt wird, die Ihr Unternehmen verwendet, finden Sie unter „Erforderl. Plattenspeicherplatz auf Multiplatforms-Plattformen” auf Seite 119.

Linux AIX *Beispiele für Verzeichniskonfigurationen auf Systemen mit AIX and Linux*

Beispiele für alternative Dateisystemkonfigurationen auf Systemen mit AIX and Linux.

Sie können die Verzeichnisstruktur von IBM MQ auf verschiedene Arten anpassen, um eine Reihe von unterschiedlichen Zielsetzungen zu erreichen.

- Platzieren Sie die Verzeichnisse qmgrs und log auf gemeinsam genutzten Remote-Dateisystemen, um einen Multi-Instanz-Warteschlangenmanager zu konfigurieren.
- Verwenden Sie separate Dateisysteme für die Daten-und Protokollverzeichnisse und ordnen Sie die Verzeichnisse verschiedenen Platten zu, um die Leistung zu verbessern, indem Sie die E/A-Konkurrenzsituationen verringern.
- Verwenden Sie schnellere Speichereinheiten für Verzeichnisse, die sich stärker auf die Leistung auswirken. Die Latenzzeit der physischen Einheit ist häufig ein wichtiger Faktor bei der Leistung des persistenten Messaging, als ob eine Einheit lokal oder über Remotezugriff angehängt ist. Die folgende Liste zeigt, welche Verzeichnisse die meisten und die leistungsfähigsten Verzeichnisse sind.
  1. log
  2. qmgrs
  3. Andere Verzeichnisse, einschließlich /usr/mqm
- Erstellen Sie die Verzeichnisse qmgrs und log in Dateisystemen, die auf einem Speicher mit einer guten Ausfallsicherheit liegen, z. B. ein redundantes Plattenarray.
- Es ist besser, die allgemeinen Fehlerprotokolle in var/mqm/errors lokal und nicht in einem Netzdateisystem zu speichern, so dass der Fehler im Zusammenhang mit dem Netzdateisystem protokolliert werden kann.

Abbildung 36 auf Seite 141 ist eine Vorlage, aus der alternative IBM MQ-Verzeichnisstrukturen abgeleitet werden können. In der Schablone stellen gepunktete Linien Pfade dar, die konfiguriert werden können. In den Beispielen werden die gepunkteten Linien durch durchgezogene Linien ersetzt, die den Konfigurationsdaten entsprechen, die in der Umgebungsvariablen AMQ_MQS_INI_LOCATION und in den Dateien mqs.ini und qm.ini gespeichert sind.

Anmerkung: Die Pfadinformationen werden angezeigt, wie sie in den mqs.ini- oder qm.ini-Dateien angezeigt werden. Wenn Sie Pfadparameter im Befehl **crtmqm** angeben, lassen Sie den Namen des Warteschlangenmanagerverzeichnisses weg: Der Warteschlangenmanagername wird dem Pfad von IBM MQhinzugefügt.

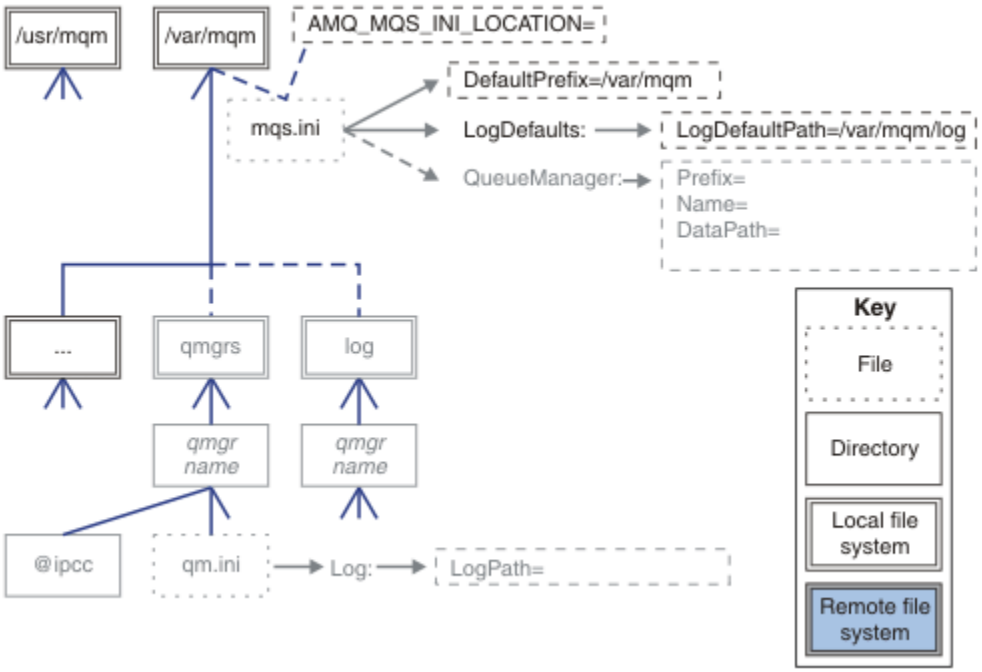


Abbildung 36. Vorlage für Verzeichnisstrukturmuster

Typische Verzeichnisstruktur für IBM MQ

Abbildung 37 auf Seite 142 zeigt die standardmäßige Verzeichnisstruktur, die in IBM MQ mit dem Befehl **crtmqm***QM1* erstellt wird.

Die Datei `mqs.ini` verfügt über eine Zeilengruppe für den QM1-Warteschlangenmanager, die unter Bezugnahme auf den Wert von `Standardpräfix` erstellt wird. Die Zeilengruppe `Protokoll` in der `qm.ini`-Datei hat einen Wert für `Protokollpfad`, der durch Verweis auf `LogDefaultPath` in `mqs.ini` festgelegt wird.

Verwenden Sie die optionalen Parameter **crtmqm** , um die Standardwerte von `DataPath` und `LogPath`zu überschreiben.

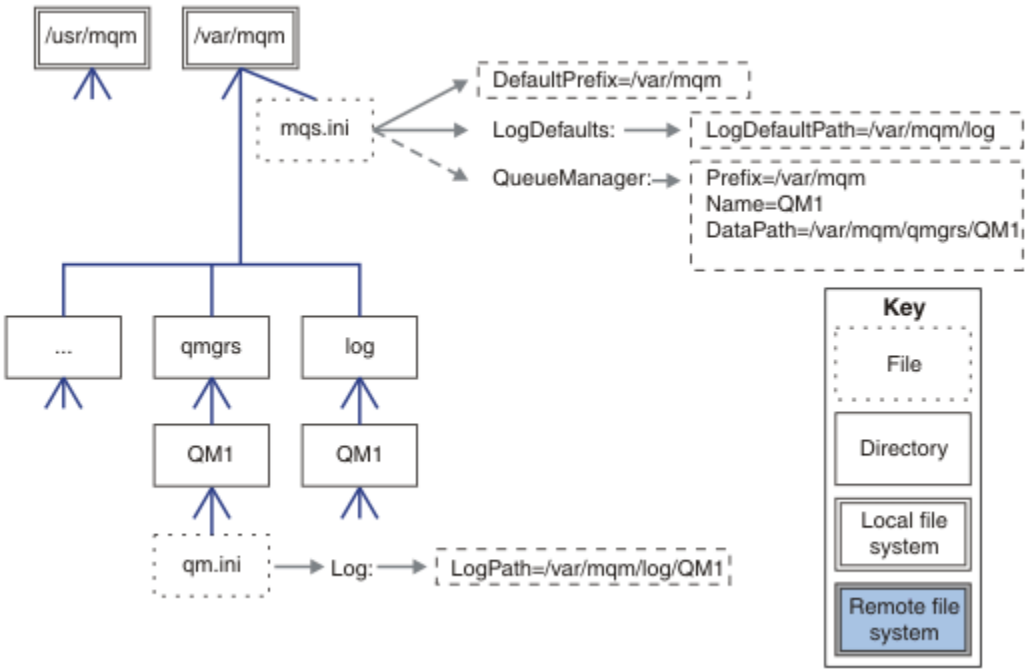


Abbildung 37. Beispiel für eine standardmäßige IBM MQ-Dateistruktur für Systeme mit AIX and Linux

`qmgrs` -und `log` -Standardverzeichnisse gemeinsam nutzen

Eine Alternative zu „Alles teilen” auf Seite 143ist die separate gemeinsame Nutzung der Verzeichnisse `qmgrs` und `log` (Abbildung 38 auf Seite 142). In dieser Konfiguration muss `AMQ_MQS_INI_LOCATION` nicht festgelegt werden, da die Standarddatei mqs.ini im lokalen `/var/mqm` -Dateisystem gespeichert wird. Die Dateien und Verzeichnisse, wie z. B. `mqclient.ini` und `mqserver.ini`, werden ebenfalls nicht gemeinsam genutzt.

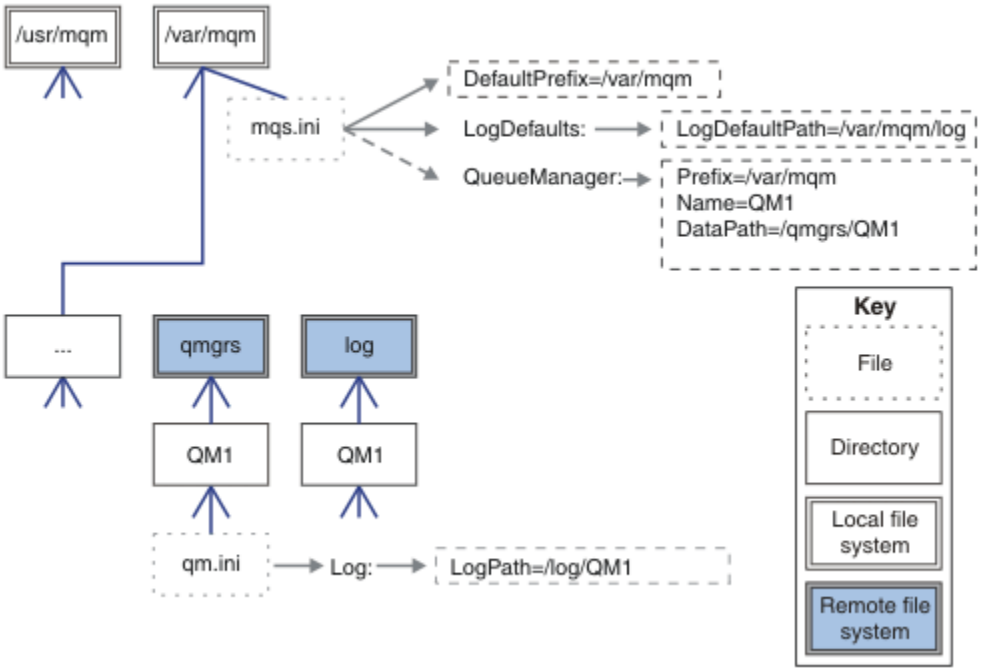


Abbildung 38. `qmgrs`- und `log`-Verzeichnisse gemeinsam nutzen

Benannte Verzeichnisse `qmgrs` und `log` gemeinsam nutzen

Bei der Konfiguration in Abbildung 39 auf Seite 143 werden `log` und `qmgrs` in einem gemeinsam genutzten Remote-Dateisystem mit dem Namen `/ha` platziert. Dieselbe physische Konfiguration kann auf zwei verschiedene Arten erstellt werden.

1. Legen Sie `LogDefaultPath=/ha` fest und führen Sie dann den Befehl **`crtmqm`** - `md` *`/ha/qmgrs`* `QM1`aus. Das Ergebnis entspricht exakt der Darstellung in Abbildung 39 auf Seite 143.
2. Lassen Sie die Standardpfade unverändert und führen Sie dann den Befehl **`crtmqm`** - `ld` *`/ha/log`* - `md` *`/ha/qmgrs`* `QM1`aus.

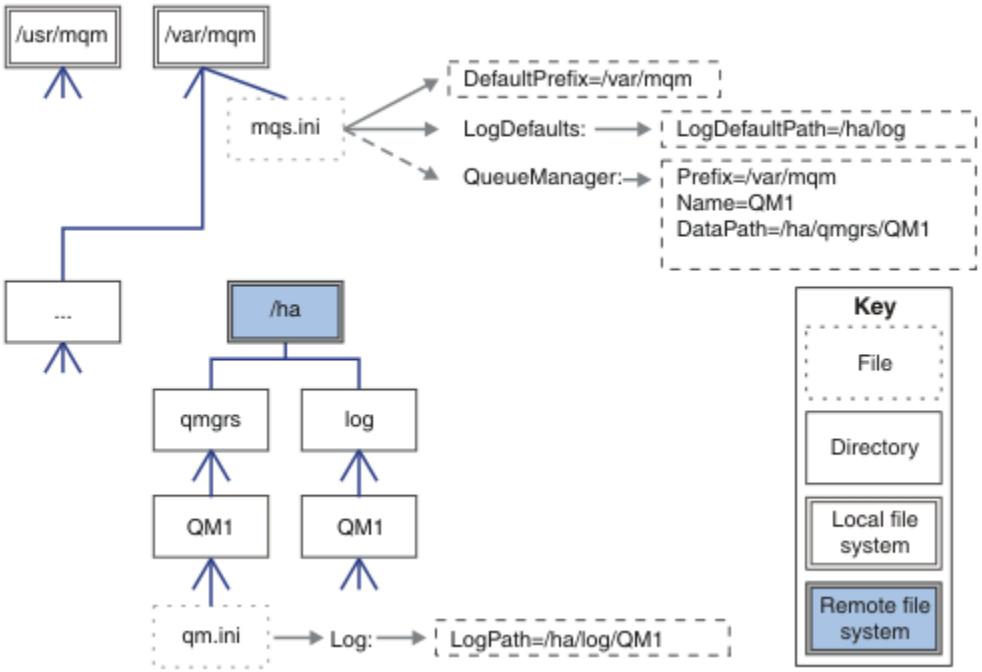


Abbildung 39. Benannte Verzeichnisse `qmgrs` und `log` gemeinsam nutzen

Alles teilen

In Abbildung 40 auf Seite 144 ist eine einfache Konfiguration für ein System mit schnellem vernetztem Dateispeicher dargestellt.

Hängen Sie `/var/mqm` als ein gemeinsam genutztes Remote-Dateisystem ein. Wenn Sie QM1 starten, sucht er standardmäßig nach `/var/mqm`, findet es auf dem gemeinsam genutzten Dateisystem und liest die `mqs.ini`-Datei in `/var/mqm`. Statt die einzige `/var/mqm/mqs.ini` -Datei für Warteschlangenmanager auf allen Ihren Servern zu verwenden, können Sie die Umgebungsvariable `AMQ_MQS_INI_LOCATION` auf jedem Server so festlegen, dass sie auf verschiedene `mqs.ini` -Dateien verweist.

Anmerkung: Der Inhalt der generischen Fehlerdatei in `/var/mqm/errors/` wird von Warteschlangenmanagern auf verschiedenen Servern gemeinsam genutzt.

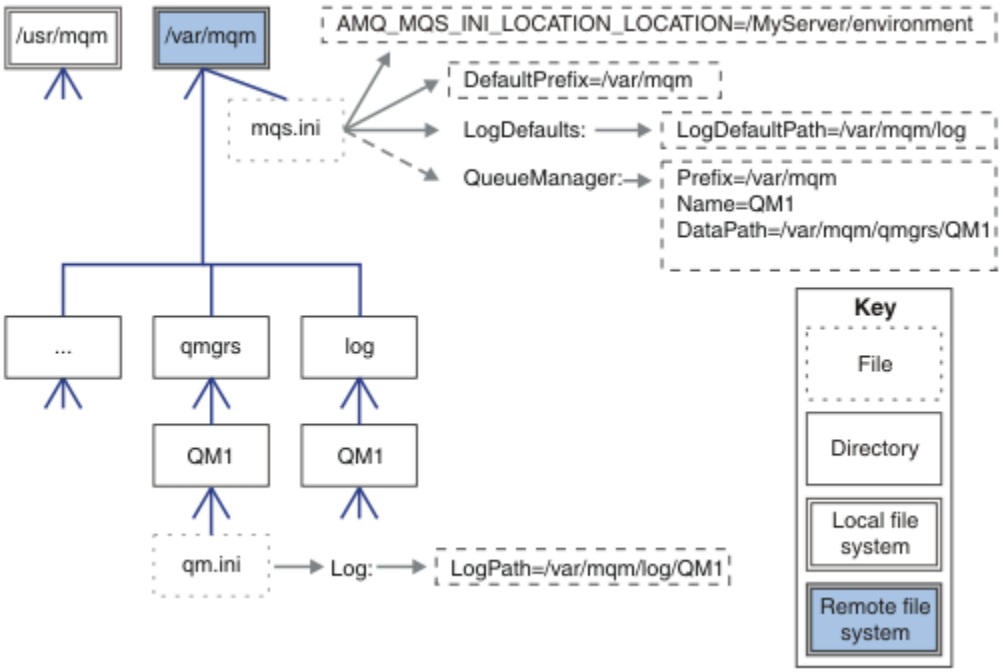


Abbildung 40. Alles teilen

Beachten Sie, dass Sie dies nicht für WS-Manager mit mehreren Instanzen verwenden können. Der Grund dafür ist, dass es für jeden Host in einem Multi-Instanz-Warteschlangenmanager erforderlich ist, über eine eigene lokale Kopie von `/var/mqm` zu verfügen, um die lokalen Daten, wie z. B. Semaphoren und gemeinsam genutzten Speicher, zu verfolgen. Diese Entitäten können nicht über Hosts hinweg gemeinsam genutzt werden.

Windows Verzeichnisstruktur auf Systemen mit Windows

Vorgehensweise zum Auffinden von Warteschlangenmanager-Konfigurationsinformationen und von Verzeichnissen unter Windows.

Folgende Standardverzeichnisse werden bei der Installation von IBM MQ for Windows erstellt:

Programmverzeichnis
`C:\Programme\IBM\MQ`

Datenverzeichnis
`C:\ProgramData \IBM \MQ`

Wichtig: Windows Für Windows-Installationen gelten die genannten Verzeichnisse, es sei denn, das Produkt wurde schon einmal installiert und die Registrierungseinträge und/oder Warteschlangenmanager dieser früheren Version sind noch vorhanden. In diesem Fall wird für die neue Installation das bereits vorhandene Datenverzeichnis verwendet. Weitere Informationen finden Sie unter Positionen von Programm- und Datenverzeichnis.

Wenn Sie wissen möchten, welches Installationsverzeichnis und welches Datenverzeichnis verwendet wird, führen Sie den Befehl dspmqver aus.

Das Installationsverzeichnis wird im Feld **InstPath** aufgelistet, und das Datenverzeichnis wird im Feld **DataPath** aufgelistet.

Wenn Sie den Befehl **dspmqver** ausführen, werden beispielsweise die folgenden Informationen angezeigt:

```
>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
```

Warteschlangenmanager mit mehreren Instanzen

Zum Konfigurieren eines Warteschlangenmanagers mit mehreren Instanzen müssen die Protokoll-und Datenverzeichnisse in den Netzspeicher gestellt werden, vorzugsweise auf einem anderen Server auf einem der Server, auf denen Instanzen des Warteschlangenmanagers ausgeführt werden.

Im Befehl **crtmqm** werden zwei Parameter bereitgestellt, **-md** und **-ld**, um die Angabe der Position der Warteschlangenmanagerdaten und Protokollverzeichnisse zu vereinfachen. Die Angabe des **-md** -Parameters hat eine vierfache Auswirkung:

1. Die `mqs.ini` Zeilengruppe `QueueManager\`*`QmgrName`* enthält eine neue Variable *Datenpfad*, die auf das Datenverzeichnis des Warteschlangenmanagers verweist. Im Gegensatz zur Variablen *Prefix* enthält der Pfad den Namen des WS-Manager-Verzeichnisses.
2. Die Konfigurationsdaten des Warteschlangenmanagers, die in der Datei mqs.ini gespeichert sind, werden auf *Name*, *Prefix*, *Directory* und *DataPath* reduziert.

Windows *Verzeichnisinhalt*

Listet die Position und den Inhalt von IBM MQ-Verzeichnissen auf.

Zu einer IBM MQ-Konfiguration gehören drei Hauptgruppen von Dateien und Verzeichnissen:

1. Ausführbare Dateien und andere schreibgeschützte Dateien, die nur aktualisiert werden, wenn die Wartung angewendet wird. For example:
   - Die Readme-Datei
   - Dateien für das IBM MQ Explorer-Plug-in und Hilfedateien
   - Lizenzdateien

   Diese Dateien werden in Tabelle 16 auf Seite 145 beschrieben.
2. Potenziell änderbare Dateien und Verzeichnisse, die für einen bestimmten WS-Manager nicht spezifisch sind. Diese Dateien und Verzeichnisse werden in Tabelle 17 auf Seite 146 beschrieben.
3. Dateien und Verzeichnisse, die für die einzelnen WS-Manager auf einem Server spezifisch sind. Diese Dateien und Verzeichnisse werden in Tabelle 18 auf Seite 147 beschrieben.

Ressourcenverzeichnisse und -dateien

Die Ressourcenverzeichnisse und -dateien enthalten den gesamten ausführbaren Code und die Ressourcen für die Ausführung eines Warteschlangenmanagers. Die Variable *FilePath* im installationsspezifischen Registrierungsschlüssel für die IBM MQ-Konfiguration enthält den Pfad zu den Ressourcenverzeichnissen.

Tabelle 16. Verzeichnisse und Dateien im Verzeichnis `FilePath`

Dateipfad	**Inhalt**
FilePath\bin	Befehle und DLLs
FilePath\bin64	Befehle und DLLs (64 Bit)
FilePath\conv	Datenkonvertierungstabellen
FilePath\doc	Hilfedateien für

Tabelle 16. Verzeichnisse und Dateien im Verzeichnis `FilePath` (Forts.)	
Dateipfad	**Inhalt**
FilePath\MQExplorer	Eclipse-Plug-ins für Explorer und Explorer
FilePath\gskit8	Globaler Sicherheitssatz
FilePath\java	Java-Ressourcen, einschließlich JRE
FilePath\licenses	Lizenzinformationen
FilePath\Non_IBM_License	Lizenzinformationen
FilePath\properties	Wird intern verwendet
FilePath\Tivoli	
FilePath\tools	Entwicklungsressourcen und -beispiele
FilePath\web	Beschrieben in Dateistruktur der Installationskomponente von IBM MQ Console und REST API für nicht bearbeitbare Dateien.
FilePath\Uninst	Wird intern verwendet
FilePath\README.TXT	Readme-Datei

Verzeichnisse, die nicht für einen Warteschlangenmanager spezifisch sind

Einige Verzeichnisse enthalten Dateien, wie z. B. Tracedateien und Fehlerprotokolle, die nicht spezifisch für einen bestimmten Warteschlangenmanager sind. Die Variable *DefaultPrefix* enthält den Pfad zu diesen Verzeichnissen. *Standardpräfix* ist Teil der Zeilengruppe `AllQueueManagers`.

Tabelle 17. Verzeichnisse und Dateien im Verzeichnis `DefaultPrefix`	
Dateipfad	**Inhalt**
DefaultPrefix\config	Wird intern verwendet
DefaultPrefix\conv	Konvertierungssteuerdatei für `ccsid_part2.tbl` und `ccsid.tbl data`, die in Datenkonvertierung beschrieben wird
DefaultPrefix\errors	Fehlerprotokolle des Nicht-WS-Managers, `AMQERR` *nn*`.LOG`
DefaultPrefix\exits	Kanalexitprogramme
DefaultPrefix\exits64	Kanalexitprogramme (64 Bit)
DefaultPrefix\ipc	Nicht verwendet
DefaultPrefix\qmgrs	Beschrieben in Tabelle 18 auf Seite 147
DefaultPrefix\trace	Tracedateien
DefaultPrefix\web	Beschrieben in Dateistruktur der Installationskomponente von IBM MQ Console und REST API für vom Benutzer bearbeitbare Dateien
DefaultPrefix\amqmjpse.txt	Wird intern verwendet

WS-Manager-Verzeichnisse

Wenn Sie einen WS-Manager erstellen, wird eine neue Gruppe von Verzeichnissen erstellt, die für den Warteschlangenmanager spezifisch sind.

Wenn Sie einen Warteschlangenmanager mit dem Parameter **-md** *filepath* erstellen, wird der Pfad in der Variable *DataPath* in der Zeilengruppe des Warteschlangenmanagers der Datei mqs.ini gespeichert. Wenn Sie einen Warteschlangenmanager erstellen, ohne den Parameter **-md** *filepath* festzulegen, werden die Warteschlangenmanagerverzeichnisse in dem Pfad erstellt, der in *DefaultPrefix*gespeichert ist, und der Pfad wird in die Variable *Prefix* in der Zeilengruppe des Warteschlangenmanagers in der Datei mqs.ini kopiert.

Tabelle 18. Verzeichnisse und Dateien in DataPath- und Prefix\qmgrs\QmgrName-Verzeichnissen

Dateipfad	**Inhalt**
DataPath\@ipcc	Standardposition für AMQCLCHL.TAB, die Clientverbindungstabelle.
DataPath\authinfo	Wird intern verwendet.
DataPath\channel	
DataPath\clntconn	
DataPath\errors	Fehlerprotokolle, AMQERR *nn*.LOG
DataPath\listener	Wird intern verwendet.
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	WS-Manager-Konfiguration
DataPath\qmstatus.ini	Status des Warteschlangenmanagers
DataPath\userdata	Kann verwendet werden, um den persistenten Status von Anwendungen zu speichern.
Prefix\qmgrs*QmgrName*	Wird intern verwendet
Prefix\qmgrs\@SYSTEM	Nicht verwendet
Prefix\qmgrs\@SYSTEM\errors	
*DataPath**autocfg*	Wird für die automatische Konfiguration verwendet

IBM i Verzeichnisstruktur unter IBM i

Dieser Abschnitt enthält eine Beschreibung des integrierten Dateisystems (IFS) sowie eine Beschreibung der IFS-Verzeichnisstruktur in IBM MQ für Server, Client und Java.

Das integrierte Dateisystem (Integrated File System, IFS) als Bestandteil von IBM i unterstützt, ähnlich wie Personal Computer und Betriebssysteme wie AIX and Linux, die Datenstromeingabe/-ausgabe und Speicherverwaltung und stellt gleichzeitig eine Integrationsstruktur für alle auf dem Server gespeicherten Informationen bereit.

Bei IBM i beginnen Verzeichnisnamen mit dem Zeichen `& (ampersand)` anstelle des Zeichens `@ (at)`. Zum Beispiel: `@system` bei IBM i ist `&system`.

IFS-Stammdateisystem für einen IBM MQ-Server

Wenn Sie einen IBM MQ-Server unter IBM i installieren, werden im IFS-Stammdateisystem die folgenden Verzeichnisse erstellt.

ProdData:

Übersicht

```
QIBM
   '-- ProdData
      '-- mqm
      '-- doc
      '-- inc
      '-- lib
      '-- samp
      '-- licenses
      '-- LicenseDoc
      '-- 5724H72_V8R0M0
```

/QIBM/ProdData/mqm
Unterverzeichnisse unterhalb dieses Verzeichnisses enthalten alle Produktdaten, z. B. C + + -Klassen, Trace-Formatdateien und Lizenzdateien. Die Daten in diesem Verzeichnis werden bei jeder Installation des Produkts gelöscht und ersetzt.

/QIBM/ProdData/mqm/doc
Eine Befehlsreferenz für die CL-Befehle wird im HTML-Format bereitgestellt und hier installiert.

/QIBM/ProdData/mqm/inc
Die Headerdateien zum Kompilieren Ihrer C-oder C + + -Programme.

/QIBM/ProdData/mqm/lib
Hilfsdateien, die von MQ verwendet werden.

/QIBM/ProdData/mqm/samp
Weitere Muster.

/QIBM/ProdData/mqm/licenses
Lizenzdateien. Die beiden Dateien für jede Sprache haben die Namen LA_ *xx* und LI_ *xx* , wobei *xx* die zweistellige Sprachkennung für jede gelieferte Sprache ist.

Außerdem werden in dem folgenden Verzeichnis Lizenzvereinbarungen gespeichert:

/QIBM/ProdData/LicenseDoc/5724H72_V8R0M0
Lizenzdateien. Die Dateien tragen den Namen 5724H72_V8R0M0_ *xx* , wobei *xx* die aus 2 oder 5 Zeichen bestehende Sprachenkennung für jede bereitgestellte Sprache ist.

UserData:

Übersicht

```
QIBM
   '-- UserData
      '-- mqm
      '-- errors
      '-- trace
      '-- qmgrs
      '-- &system
      '-- qmgrname1
      '-- qmgrname2
      '-- and so on
```

/QIBM/UserData/mqm
Unterverzeichnisse unterhalb dieses Verzeichnisses enthalten alle Benutzerdaten, die sich auf Warteschlangenmanager beziehen.

Wenn Sie das Produkt installieren, wird eine Datei mqs.ini im Verzeichnis/QIBM/UserData/mqm/ erstellt (sofern sie nicht bereits von einer früheren Installation vorhanden ist).

Wenn Sie einen Warteschlangenmanager erstellen, wird eine Datei qm.ini im Verzeichnis/QIBM/UserData/mqm/qmgrs/ *QMGRNAME* /erstellt (wobei *QMGRNAME* für den Namen des Warteschlangenmanagers steht).

Die Daten in den Verzeichnissen werden beibehalten, wenn das Produkt gelöscht wird.

IFS-Stammdateisystem für einen IBM MQ MQI client

Wenn Sie einen IBM MQ MQI client for IBM i installieren, werden im IFS-Stammdateisystem die folgenden Verzeichnisse erstellt.

ProdData:

Übersicht

```
QIBM
   '-- ProdData
      '-- mqm
      '-- lib
```

/QIBM/ProdData/mqm
Unterverzeichnisse unterhalb dieses Verzeichnisses enthalten alle Produktdaten. Die Daten in diesem Verzeichnis werden gelöscht und jedes Mal ersetzt, wenn das Produkt ersetzt wird.

UserData:

Übersicht

```
QIBM
   '-- UserData
      '-- mqm
      '-- errors
      '-- trace
```

/QIBM/UserData/mqm
Unterverzeichnisse unterhalb dieses Verzeichnisses enthalten alle Benutzerdaten.

IFS-Stammdateisystem für einen IBM MQ Java

Wenn Sie IBM MQ Java unter IBM i installieren, werden im IFS-Stammdateisystem die folgenden Verzeichnisse erstellt.

ProdData:

Übersicht

```
QIBM
   '-- ProdData
      '-- mqm
      '-- java
      '--samples
      '-- bin
      '-- lib
```

/QIBM/ProdData/mqm/java
Unterverzeichnisse unterhalb dieses Verzeichnisses enthalten alle Produktdaten, einschließlich Java-Klassen. Die Daten in diesem Verzeichnis werden gelöscht und jedes Mal ersetzt, wenn das Produkt ersetzt wird.

/QIBM/ProdData/mqm/java/samples
Unterverzeichnisse unterhalb dieses Verzeichnisses enthalten alle Java-Beispielklassen und -daten.

Bibliotheken, die von Server-und Clientinstallationen erstellt

Bei der Installation des IBM MQ-Servers oder -Clients werden die folgenden Bibliotheken erstellt.

- QMQM

  Die Produktbibliothek.

- QMQMSAMP

  Die Beispielbibliothek (wenn Sie die Beispiele installieren möchten).

- QMxxxx

  Nur Server.

  Jedes Mal, wenn Sie einen Warteschlangenmanager erstellen, erstellt IBM MQ automatisch eine zugehörige Bibliothek mit einem Namen wie QMxxxx, wobei xxxx vom Namen des Warteschlangenmanagers abgeleitet ist. Diese Bibliothek enthält Objekte, die für den Warteschlangenmanager spezifisch sind, einschließlich der Journale und der zugeordneten Empfänger. Standardmäßig wird der Name dieser Bibliothek aus dem Namen des Warteschlangenmanagers abgeleitet, der mit den Zeichen QM vorangestellt ist. Für einen WS-Manager mit dem Namen TEST würde die Bibliothek beispielsweise QMTEST genannt.

  Anmerkung: Wenn Sie einen WS-Manager erstellen, können Sie den Namen seiner Bibliothek angeben, wenn Sie möchten. For example:

  ```
  CRTMQM MQMNAME(TEST) MQMLIB(TESTLIB)
  ```

  Mit dem Befehl "WRKLIB" können Sie alle Bibliotheken auflisten, die von IBM MQ for IBM i erstellt wurden. Für die Bibliotheken der Warteschlangenmanager wird der Text QMGR: QMGRNAME angezeigt. Das Format des Befehls lautet wie folgt:

  ```
  WRKLIB LIB(QM*)
  ```

  Diese WS-Manager-zugeordneten Bibliotheken werden beibehalten, wenn das Produkt gelöscht wird.

Multi Dateisystemunterstützung für MFT auf Multiplatforms planen

IBM MQ Managed File Transfer MFT -Agenten können zum Übertragen von Daten an Dateien und aus Dateien in einem Dateisystem verwendet werden. Darüber hinaus können Ressourcenüberwachungen, die in einem Agenten ausgeführt werden, für die Überwachung von Dateien in einem Dateisystem konfiguriert werden.

MFT erfordert, dass diese Dateien in einem Dateisystem gespeichert werden, das Sperren unterstützt. Hierfür gibt es zwei Gründe:

- Ein Agent sperrt eine Datei, um sicherzustellen, dass sie sich nicht ändert, sobald er begonnen hat, Daten aus ihr zu lesen oder in sie zu schreiben.
- Ressourcenmonitore sperren Dateien, um sicherzustellen, dass sie von keinen anderen Prozessen verwendet werden.

Agenten und Ressourcenüberwachungen verwenden die Java Methode **`FileChannel.tryLock()`** , um Sperren auszuführen, und das Dateisystem muss in der Lage sein, Dateien zu sperren, wenn es mit diesem Aufruf dazu aufgefordert wird.

Wichtig: Die folgenden Dateisysteme werden nicht unterstützt, da sie die technischen Anforderungen von MFTnicht erfüllen:

- GlusterFS
- NFS Version 3

Multi Kreisförmige oder lineare Protokollierung auf Multiplatforms auswählen

In IBM MQ können Sie zwischen Umlaufprotokollierung und linearer Protokollierung wählen. Die folgenden Informationen geben Ihnen einen Überblick über beide Typen.

Vorteile der Umlaufprotokollierung

Die Hauptvorteile der Umlaufprotokollierung sind, dass die Umlaufprotokollierung wie folgt lautet:

- Easier zu verwalten.

  Wenn Sie die Umlaufprotokollierung ordnungsgemäß für Ihre Workload konfiguriert haben, ist keine weitere Verwaltung erforderlich. Für die lineare Protokollierung müssen Datenträgerimages aufgezeichnet werden, und die Protokollspeicherbereiche, die nicht mehr benötigt werden, müssen archiviert oder gelöscht werden.
- Bessere Leistung

  Die Umlaufprotokollierung führt eine bessere Leistung als die lineare Protokollierung aus, da die Umlaufprotokollierung Protokollspeicherbereiche wiederverwenden kann, die bereits formatiert wurden. Während die lineare Protokollierung neue Protokollerweiterungen zuordnen und diese formatieren muss.

Weitere Informationen finden Sie im Abschnitt Protokolle verwalten .

Vorteile der linearen Protokollierung

Der Hauptvorteil der linearen Protokollierung besteht darin, dass die lineare Protokollierung einen Schutz vor mehr Fehlern bietet.

Weder die kreisförmige noch die lineare Protokollierung schützen vor einem beschädigten oder gelöschten Protokoll, oder Nachrichten oder Warteschlangen, die von Anwendungen oder vom Administrator gelöscht wurden.

Lineare Protokollierung (aber nicht kreisförmig) ermöglicht die Wiederherstellung beschädigter Objekte. Die lineare Protokollierung bietet also Schutz vor beschädigten oder gelöschten Warteschlangendateien, da diese beschädigten Warteschlangen aus einem linearen Protokoll wiederhergestellt werden können.

Sowohl kreisförmiger als auch linearer Schutz vor Stromausfall und Kommunikationsfehler wie in Wiederherstellung nach Stromausfall oder Kommunikationsfehlern beschrieben beschrieben.

Weitere Überlegungen

Ob linear oder kreisförmig gewählt wird, hängt davon ab, wie viel Redundanz Sie benötigen.

Es entstehen Kosten für die Auswahl von mehr Redundanz, d. a. der linearen Protokollierung, die durch die Leistungskosten und die Verwaltungskosten verursacht werden.

Weitere Informationen finden Sie unter Protokolltypen .

AIX gemeinsam genutzter Speicher unter AIX

Wenn bestimmte Anwendungstypen wegen einer Speicherbegrenzung unter AIX keine Verbindung herstellen können, kann dies in den meisten Fällen durch eine angepasste Einstellung der Variablen "EXTSHM=ON" behoben werden.

Bei einigen 32-Bit-Prozessen unter AIX kann eine Betriebssystembegrenzung dazu führen, dass die Prozesse keine Verbindung zu IBM MQ-Warteschlangenmanagern herstellen können. Jede Standardverbindung zu IBM MQ verwendet gemeinsam genutzten Speicher, aber im Gegensatz zu anderen UNIX-Plattformen ermöglicht AIX 32-Bit-Prozessen nur 11 gemeinsam genutzte Speichergruppen zuzuordnen.

Bei den meisten 32-Bit-Prozessen tritt dieser Grenzwert nicht auf, aber Anwendungen mit hohem Speicherbedarf können möglicherweise mit Ursachencode 2102: MQRC_RESOURCE_PROBLEM keine Verbindung zu IBM MQ herstellen. In den folgenden Anwendungstypen wird möglicherweise dieser Fehler angezeigt:

- Programme, die auf einer Java Virtual Machine mit 32-Bit-Konfiguration ausgeführt werden.
- Programme, die die großen oder sehr großen Speichermodelle verwenden
- Programme, die Verbindungen zu vielen Warteschlangenmanagern oder Datenbanken herstellen
- Programme, die an gemeinsam genutzte Speichergruppen angehängt sind

AIX bietet eine Erweiterungsfunktion an, mit der 32-Bit-Prozesse mehr gemeinsam genutzten Speicher anhängen können. Wenn Sie eine Anwendung mit dieser Funktion ausführen möchten, exportieren Sie die Umgebungsvariable EXTSHM=ON, bevor Sie Ihre Warteschlangenmanager und Ihr Programm starten. Die Funktion EXTSHM=ON verhindert in den meisten Fällen diesen Fehler, ist aber mit Programmen, die die Option SHM_SIZE der Funktion shmctl verwenden, nicht kompatibel.

IBM MQ MQI client-Anwendungen und alle 64-Bit-Prozesse sind von dieser Begrenzung nicht betroffen. Sie können unabhängig von der Einstellung der Variablen "EXTSHM" eine Verbindung zu IBM MQ-Warteschlangenmanagern herstellen.

Linux AIX IPC-Ressourcen für IBM MQ und UNIX System V

Ein WS-Manager verwendet einige IPC-Ressourcen. Verwenden Sie **ipcs -a** , um herauszufinden, welche Ressourcen verwendet werden.

Diese Informationen gelten nur für IBM MQ auf Systemen mit AIX and Linux.

IBM MQ nutzt Ressourcen für Interprozesskommunikation (Interprocess Communication, IPC) von System V, also *Semaphore* und *gemeinsam genutzte Speichersegmente*, um Daten zu speichern und an Systemkomponenten zu übergeben. Diese Ressourcen werden von WS-Managerprozessen und -Anwendungen verwendet, die eine Verbindung zum Warteschlangenmanager herstellen. IBM MQ MQI clients verwenden keine IPC-Ressourcen, mit Ausnahme der IBM MQ-Tracesteuerung. Verwenden Sie den UNIX -Befehl **ipcs -a** , um vollständige Informationen zur Anzahl und Größe der momentan auf der Maschine verwendeten IPC-Ressourcen abzurufen.

Linux AIX Prozesspriorität von IBM MQ und UNIX

Good Practices beim Festlegen der Werte für die Prozesspriorität *nice* .

Diese Informationen gelten nur für IBM MQ auf Systemen mit AIX and Linux.

Wenn Sie einen Prozess im Hintergrund ausführen, kann diesem Prozess durch die aufrufende Shell ein höherer *nice* -Wert (und damit eine niedrigere Priorität) erteilt werden. Dies kann allgemeine Auswirkung auf die Leistung von IBM MQ haben. Wenn es in stark beanspruchten Situationen viele gebrauchsfertige Threads mit einer höheren Priorität und einigen mit einer niedrigeren Priorität gibt, können die Merkmale der Betriebssystem-Zeitplanung die Threads mit der niedrigeren Priorität der Prozessorzeit vorenthalten.

Es ist ein bewährtes Verfahren, dass unabhängig gestartete Prozesse, die Warteschlangenmanagern wie **runmqlsr**zugeordnet sind, dieselben *nice* -Werte haben wie der Warteschlangenmanager, dem sie zugeordnet sind. Stellen Sie sicher, dass die Shell diesen Hintergrundprozessen keinen höheren *nice* -Wert zuordnet. Verwenden Sie beispielsweise in 'ksh' die Einstellung "`set +o bgnice`" , um zu verhindern, dass 'ksh ' den Wert *nice* für Hintergrundprozesse erhöht. Sie können die *nice* -Werte von aktiven Prozessen überprüfen, indem Sie die Spalte *NI* einer Liste " ps -efl " prüfen.

Starten Sie außerdem IBM MQ-Anwendungsprozesse mit derselben Prioritätszahl (*nice*) wie beim Warteschlangenmanager. Wenn sie mit unterschiedlichen *nice* -Werten ausgeführt werden, blockiert ein Anwendungsthread möglicherweise einen WS-Manager-Thread, oder umgekehrt, wodurch sich die Leistung absetzt.

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

“IBM MQ-Architektur planen” on page 5

Beachten Sie bei der Planung einer IBM MQ-Umgebung die Unterstützung, die IBM MQ für Architekturen mit einzelnen oder mehreren Warteschlangenmanagern sowie für Punkt-zu-Punkt- und Publish/Subscribe-Messaging bereitstellt. Planen Sie auch den Ressourcenbedarf und die Nutzung von Protokollierungs- und Sicherheitsfunktionen.

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 156
- “System LX usage” on page 156
- “Storage configuration” on page 157
- “Disk storage” on page 161

See, “Where to find more information about storage and performance requirements” on page 162 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 156 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. Download-Links für die Programmverzeichnisse finden Sie unterIBM MQ for z/OS Programmverzeichnis PDF-Dateien .

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 157
- “Channel initiator storage configuration from IBM MQ 9.4.0” on page 160

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.

z/OS V 9.4.0 *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 180 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 197 describes how to determine how much storage Db2 requires for the IBM MQ data.

- **coupling facility storage**

  “Defining coupling facility resources” on page 187 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 162 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. Download-Links für die Programmverzeichnisse finden Sie unterIBM MQ for z/OS Programmverzeichnis PDF-Dateien .
IEALIMIT and IEFUSI exits	See IEALIMIT and IEFUSI in the *z/OS:MVS Installation Exits* documentation.
Latest information	IBM MQ SupportPac website SupportPacs für IBM MQ und andere Projektbereiche.
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

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 167 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 165. 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 199.

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.

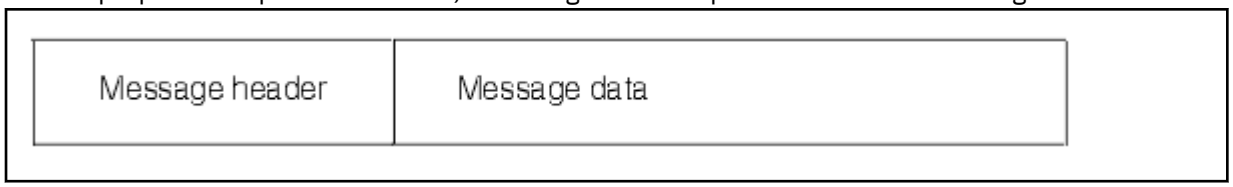
Note: The sizes of the structures and control information given in this section are liable to change between major releases. For details specific to your release of IBM MQ, refer to SupportPac MP16 - IBM MQ for z/OS -Kapazitätsplanung und -optimierung and IBM MQ -Produktfamilie-Leistungsberichte

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 165:

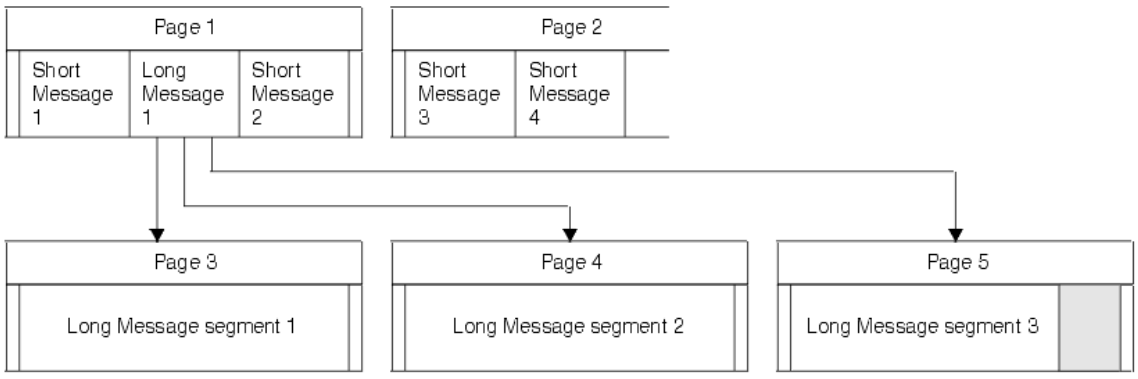


Figure 41. How IBM MQ stores long messages on page sets

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 167
2. “Decide on the settings for each buffer pool” on page 168
3. “Monitor the performance of buffer pools under expected load” on page 168
4. “Adjust buffer pool characteristics” on page 168

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 169 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 167, 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 168, if required.

Use the performance statistics from “Monitor the performance of buffer pools under expected load” on page 168 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 -Kapazitätsplanung und -optimierung 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 171
- Active log placement
- “Active log encryption with z/OS data set encryption” on page 172

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 181 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:

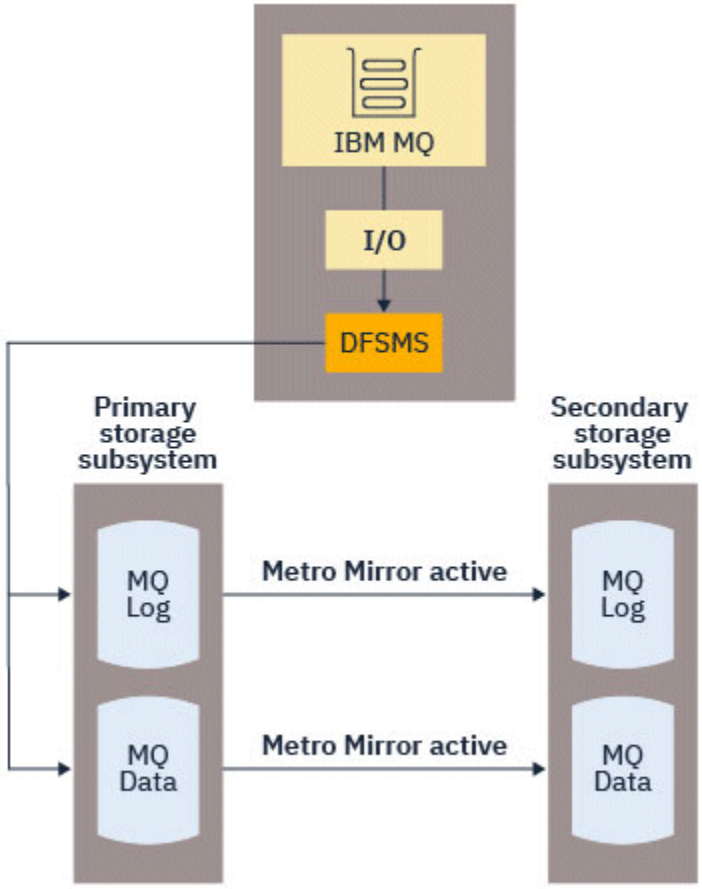
- Active logs
- Archive logs
- Bootstrap data set (BSDS)
- Page sets
- Shared message data set (SMDS)

- 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.

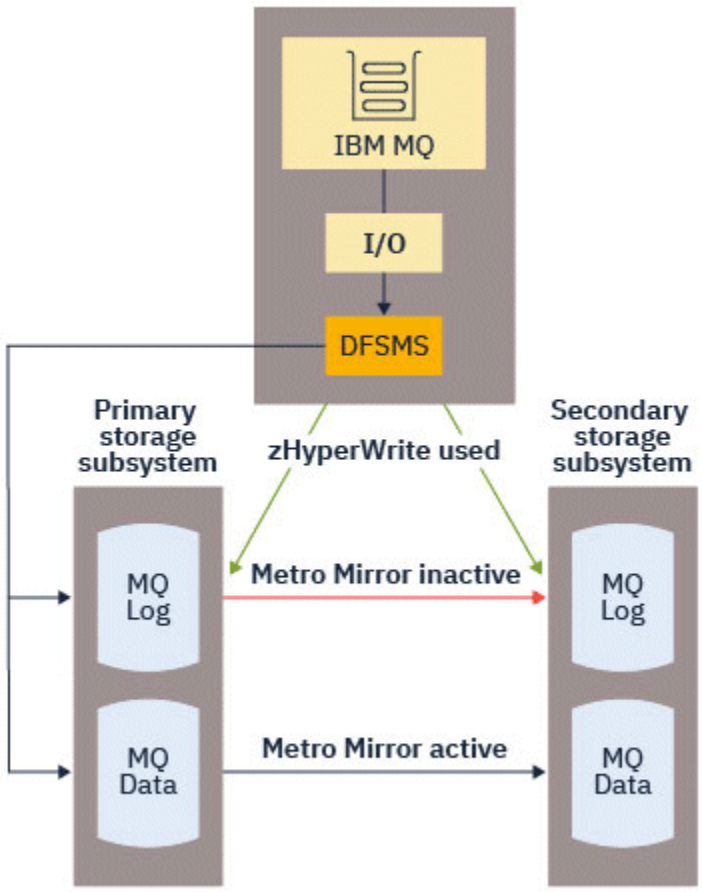


Because both writes to the primary and secondary storage subsystems need to complete before the write returns to IBM MQ, use of Metro Mirror can have a performance impact. You need to balance this performance impact against the availability benefits of using Metro Mirror.

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 ***Schnellerer Protokolldurchsatz mit zHyperLink***

zHyper Die Link-Technologie soll die Eingabe-/Ausgabelatenz (E/A) reduzieren, indem sie einen schnellen, zuverlässigen und direkten Kommunikationspfad zwischen der CPU und dem E/A-Gerät bereitstellt.

ÜberblickzHyper Verknüpfung

zHyperLink kann den Durchsatz der aktiven Protokolldatei verbessern und die IBM MQ -Transaktionszeit um bis zu 3.5 Mal reduzieren. Dieses Ziel wird erreicht durch die InstallationzHyper Link-Adapter auf derz/OS Host, wählen SieIBM Speicherhardware und deren Verbindung überzHyper Verbindungskabel. Dadurch wird eine Punkt-zu-Punkt-Verbindung zwischen der CPU und dem E/A-Gerät hergestellt, die die E/A-Antwortzeit um bis zu das Zehnfache reduziert, im Vergleich zuIBM z Leistungsstarkes FICON® (zHPF). Diese niedrige Antwortzeit wird durch die Verwendung synchroner E/A-Anforderungen erreicht.

Die Vorteile von synchronem I/O gegenüber asynchronem I/O

DerIBM MQ Die Logger-Aufgabe besteht aus einer Schleife, die auf das nächste Datenelement wartet, das in das Protokoll geschrieben werden muss. Wenn diese Daten verfügbar sind, plant der Logger den Schreibvorgang, wartet, bis er abgeschlossen ist, und fährt dann mit dem nächsten Datenelement fort.

Herkömmliche E/A sind langsamer als die CPU. Daher ist es am effizientesten, die E/A asynchron auszuführen, um die CPU für andere Aufgaben freizugeben. Daher erfordert der herkömmliche asynchrone E/A-Vorgang, dass die Logger-Aufgabe angehalten wird, bis der Schreibvorgang abgeschlossen ist. Wenn der Schreibvorgang abgeschlossen ist, muss die Logger-Aufgabe warten, bis eine CPU verfügbar ist. Dadurch kommt es zu einer kurzen erneuten Verteilungsverzögerung sowie zu Verzögerungen durch die Neubefüllung des CPU-Cache.

zHyper Link bietet viel schnellere I/O-Zeiten, die näher an der CPU-Geschwindigkeit liegen, daher mitzHyper Link, I/O kann synchron ausgeführt werden, was bedeutet, dass die Logger-Aufgabe während des Schreibvorgangs nicht angehalten wird, wodurch erneute Verteilung und Cache-bedingte Verzögerungen vermieden werden.

Während der Schreibvorgang stattfindet, nutzt die Logger-Aufgabe weiterhin aktiv die CPU, was die CPU-Auslastung im Vergleich zum herkömmlichen E/A erhöht.

Wenn der Warteschlangenmanager versucht,zHyper Link und diezHyper Das Schreiben des Links schlägt fehl, beispielsweise aufgrund von Konfigurationsproblemen. Anschließend greift der Warteschlangenmanager transparent auf herkömmliche E/A zurück.

Hardwaremindestvoraussetzungen

- IBM z14 oder höher
- DS8880 oder höher

Softwarevoraussetzungen

- zHyperLink Express wird unter z/OS 2.3 oder höher unterstützt.
- Das z/OS -Image muss in einer LPAR ausgeführt werden, nicht als Gast unter IBM z/VM®.
- Für zHyperLink muss IBM z High-Performance FICON (zHPF) aktiviert sein.

zHyperLink mit aktiven IBM MQ -Protokollen verwenden

Zur VerwendungzHyper Um eine Verknüpfung mit den aktiven Protokollen eines Warteschlangenmanagers herzustellen, müssen Sie Folgendes tun:

- KonfigurierenIBM MQ benutzenzHyper Link und
- Stellen Sie sicher, dass die aktiven Protokolle aktiviert sindzHyper Linkfähige Volumes.

  Sehen Anfangen mitIBMzHyper Link fürz/OS für mehr Informationen.

Sie können konfigurierenIBM MQ benutzenzHyper Stellen Sie mithilfe einer der folgenden Methoden eine Verknüpfung her:

- Geben Sie ZHYLINK(YES) in den Protokollparametern an.
- Geben Sie den Befehl SET LOG ZHYLINK(YES) aus.

Anmerkungen:

- zHyperLink erfordert, dass zHyperWrite eingeschaltet ist. Damit ZHYLINK verwendet werden kann, muss auch ZHYWRITE in den Protokollparametern eingeschaltet sein. Wenn ZHYLINK (YES) nur angegeben wird, wenn ZHYWRITE (NO) im Warteschlangenmanager festgelegt ist, wird der Parameter ZHYWRITE automatisch auf YES gesetzt.
- Der explizite Versuch, ZHYLINK (YES) mit ZHYWRITE (NO) festzulegen, führt zu einer abnormalen Beendigung des Befehls SET LOG.
- Die Einstellung ZHYLINK=YES in den ZPRMs überschreibt ZHYWRITE auf YES.

Wenn Probleme auftreten, finden Sie weitere Informationen unter Fehlerbehebung zHyperLink .

IBM MQ erfordert nicht, dass sich alle aktiven Protokolldateien auf zHyperLink-fähigen Datenträgern befinden. Es wird jedoch empfohlen, dies zu tun. Wenn IBM MQ erkennt, dass sich einige aktive Protokolldateien auf zHyperLink-fähigen Datenträgern befinden und andere nicht, gibt es die Nachricht CSQJ601E aus und setzt die Verarbeitung fort.

IBM MQ prüft, ob aktive Protokolldateien zHyperLink-fähig sind, wenn die Dateien zum ersten Mal geöffnet werden. Protokolldateien werden entweder beim Start des Warteschlangenmanagers oder beim dynamischen Hinzufügen mit dem Befehl DEFINE LOG geöffnet. Wenn die Protokolldateien zHyperLink-fähig gemacht werden, während ein Warteschlangenmanager geöffnet ist, erkennt der Warteschlangenmanager dies erst, wenn er erneut gestartet wurde.

Wenn ZHYLINK (YES) angegeben wird, versucht IBM MQ stets, zHyperLink beim Schreiben in die aktiven Protokolle zu verwenden, unabhängig davon, ob sich die Protokolle auf zHyperLink-fähigen Datenträgern befinden. Der Versuch, zHyperLink zu verwenden, hat keine negativen Auswirkungen, wenn sich die Protokolle nicht auf zHyperLink-fähigen Datenträgern befinden.

Sie können die Ausgabe des Befehls DISPLAY LOG verwenden, um den Status von zHyperLink für die momentan aktiven Protokolldateien anzugeben:

```
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
```

Der zHyperLink-Status ist einer der folgenden:

JA
zHyperLink ist auf dem WS-Manager aktiviert und wird bei allen Schreibvorgängen versucht.

NEIN
zHyperLink ist auf dem Warteschlangenmanager nicht aktiviert und die Datei **ist auf zHyperLink-fähigen Datenträgern nicht** .

CAPABLE
zHyperLink ist auf dem WS-Manager nicht aktiviert und die Datei **befindet sich** auf einem zHyperLink-fähigen Datenträger.

Es gibt mehrere zusätzliche SMF-Statistiken zur Überwachung und zum VerständniszHyper Link-Performance; siehezHyper Link-Statistiken für Details.

Schreibsitzungen

Beim BenutzenzHyper Link, ein oder mehrerezHyper Es werden Link-Write-Sitzungen mit dem DASD hergestellt. Aktuelle DASDs unterstützen maximal 64 gleichzeitige Schreibsitzungen. Sie sollten daher sorgfältig überlegen, welche Warteschlangenmanager Sie aktivieren.zHyper Link aktiviert ist und ob andere Subsysteme, wie z. B.Db2 verwenden auchzHyper Link zum Schreiben auf dasselbe DASD. Wenn keine Schreibsitzungen mehr verfügbar sind, wechselt der Warteschlangenmanager automatisch zurück zur Verwendung des herkömmlichen asynchronen E/A.

Sie können die Anzahl derzHyper Verknüpfen Sie Schreibsitzungen wie folgt:

```
Number of log copies (either 1 or 2) * number of stripes per log copy * 2
if Metro Mirror (PPRC) is used.
```

Daher kann ein Queue-Manager im Single-Logging-Modus mit einem Stripe und ohneMetro Mirror verwendet eine einzelne Schreibsitzung. Ein Warteschlangenmanager im dualen Protokollierungsmodus mit zwei Stripes und PPRC verwendet 8 Schreibsitzungen.

Anmerkung: WährendMetro Mirror zur Folge hat, dass doppelt so viele Schreibsitzungen verwendet werden, diese Schreibsitzungen werden gleichmäßig zwischen den beiden gespiegelten DASDs aufgeteilt.

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 179 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 178 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 178, 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 184 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 “Verteilte Warteschlangen und Cluster planen” on page 21 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 153
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 153
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

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 189 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 189. 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 191 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 191 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

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 IBM MQ for z/OS verwalten 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 207 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.

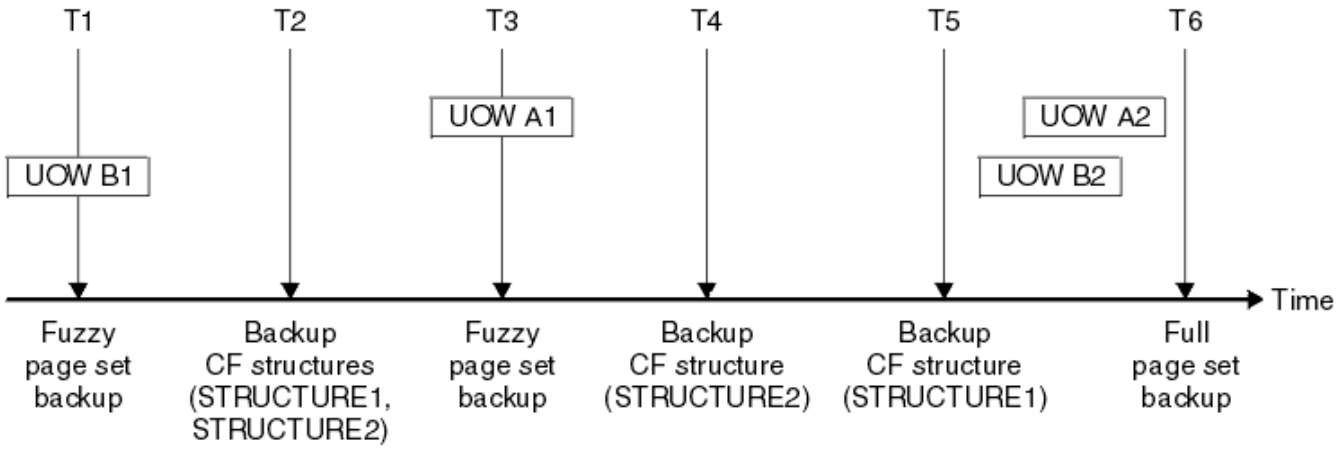


Figure 43. Example of queue manager backup activity

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 153
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.

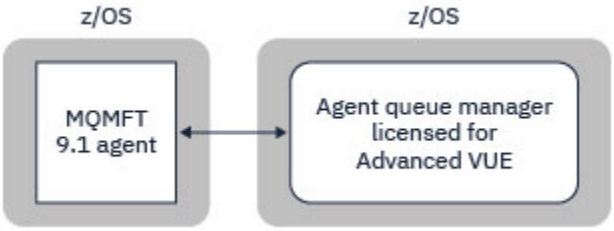


Figure 44. MFT 9.1 agents on z/OS can connect to a queue manager using the CLIENT transport, assuming the queue manager is licensed for Advanced VUE.

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.

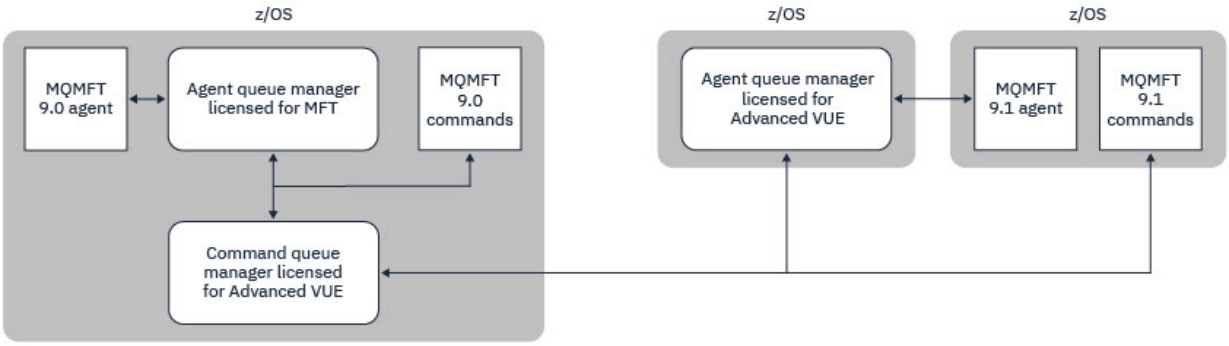


Figure 46. Commands connect to the command queue manager for an MFT topology. When running these commands on z/OS, the command queue manager must also be on z/OS

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.

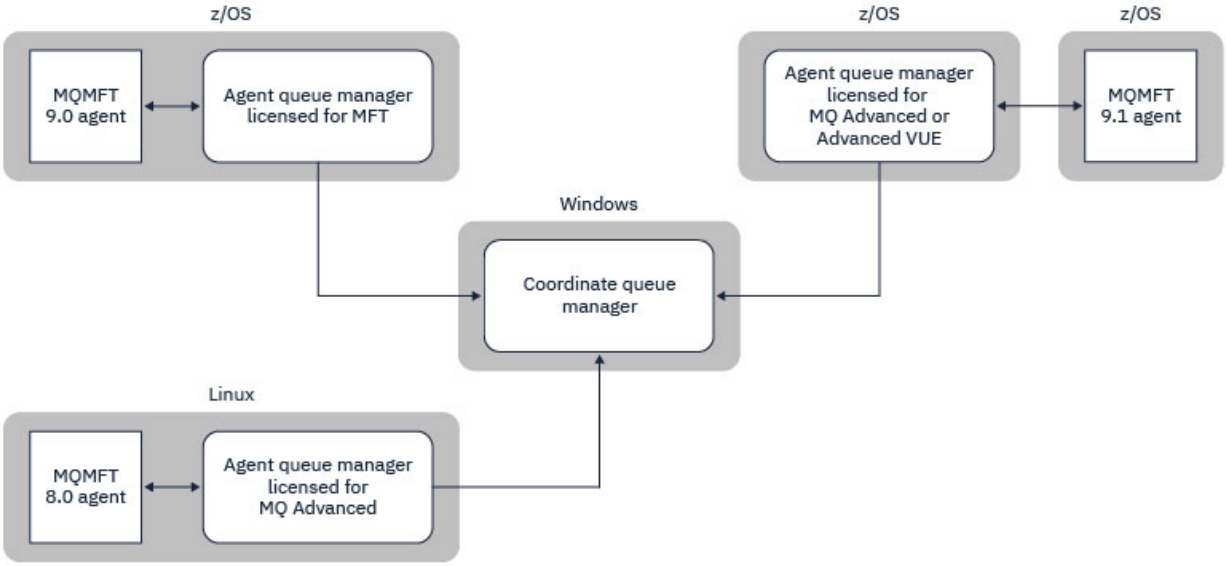


Figure 47. MFT agents running on z/OS can be part of an MFT topology where the coordination queue manager is running on an IBM MQ multiplatform.

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.

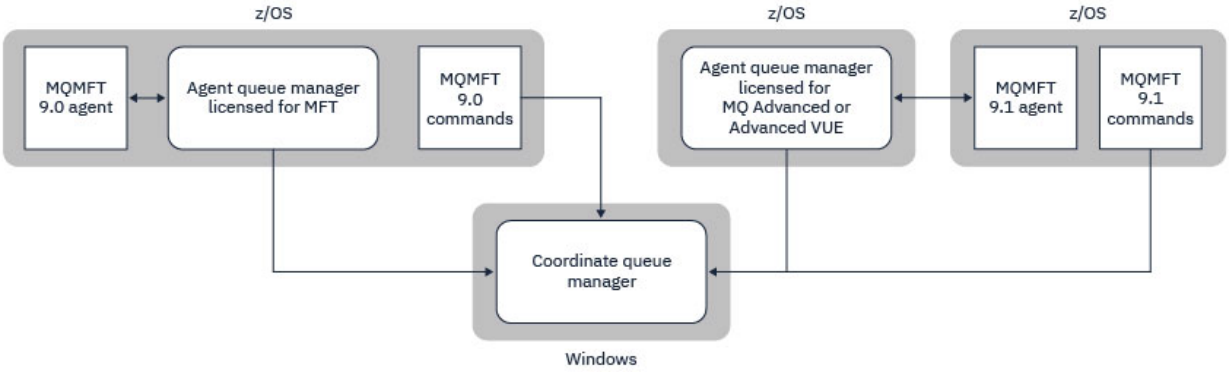


*Figure 48. Certain commands, such as **fteListAgents**, connect directly to the coordination queue manager for an MFT topology.*

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.

Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in diesem Dokument beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf Produkte, Programme oder Services von IBM bedeuten nicht, dass nur Produkte, Programme oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder andere Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Fremdprodukten, Fremdprogrammen und Fremdservices liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing
IBM Europe, Middle East & Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Défense
U.S.A.

Bei Lizenzanforderungen zu Double-Byte-Information (DBCS) wenden Sie sich bitte an die IBM Abteilung für geistiges Eigentum in Ihrem Land oder senden Sie Anfragen schriftlich an folgende Adresse:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuausgabe veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Europe, Middle East & Africa
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in diesen Informationen beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren und können Namen von Personen, Firmen, Marken oder Produkten enthalten. Sämtliche dieser Namen sind fiktiv. Ähnlichkeiten mit Namen und Adressen tatsächlicher Unternehmen oder Personen sind zufällig.

COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellensprache geschrieben sind und Programmiertechniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Beispielprogramme kostenlos ohne Zahlung an IBM in jeder Form kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Beispielprogramme geschrieben sind. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten.

Wird dieses Buch als Softcopy (Book) angezeigt, erscheinen keine Fotografien oder Farbabbildungen.

Informationen zu Programmierschnittstellen

Die bereitgestellten Informationen zur Programmierschnittstelle sollen Sie bei der Erstellung von Anwendungssoftware für dieses Programm unterstützen.

Dieses Handbuch enthält Informationen zu geplanten Programmierschnittstellen, die es dem Kunden ermöglichen, Programme zum Abrufen der Services von IBM MQzu schreiben.

Diese Informationen können jedoch auch Angaben über Diagnose, Bearbeitung und Optimierung enthalten. Die Informationen zu Diagnose, Bearbeitung und Optimierung sollten Ihnen bei der Fehlerbehebung für die Anwendungssoftware helfen.

Wichtig: Verwenden Sie diese Diagnose-, Änderungs-und Optimierungsinformationen nicht als Programmierschnittstelle, da sie Änderungen unterliegen.

Marken

IBM, das IBM Logo, ibm.com, sind Marken der IBM Corporation in den USA und/oder anderen Ländern. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite "Copyright and trademark information"www.ibm.com/legal/copytrade.shtml. Weitere Produkt- und Servicenamen können Marken von IBM oder anderen Unternehmen sein.

Microsoft und Windows sind eingetragene Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Dieses Produkt enthält Software, die von Eclipse Project (https://www.eclipse.org/) entwickelt wurde.

Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen.

Teilenummer:

(1P) P/N: