

9.4

Planejando o IBM MQ

IBM

Nota

Antes de usar estas informações e o produto que elas suportam, leia as informações em [“Avisos” na página 211](#).

Esta edição se aplica à versão 9, liberação 4 do IBM® MQ e a todas as liberações e modificações subsequentes, até que seja indicado de outra forma em novas edições

Ao enviar informações para a IBM, você concede à IBM um direito não exclusivo de usar ou distribuir as informações da maneira que julgar apropriada, sem incorrer em qualquer obrigação para com você

© **Copyright International Business Machines Corporation 2007, 2024.**

Índice

Planejando.....	5
IBM MQ tipos de liberação: considerações de planejamento.....	6
Considerações sobre o IBM MQ e o IBM MQ Appliance no local para adequação à GDPR.....	9
Arquiteturas com base em um único gerenciador de filas.....	18
Arquiteturas com base em vários gerenciadores de filas.....	19
Planejando filas distribuídas e clusters.....	20
Planejando sua rede publicar/assinar distribuída.....	73
Planejando seus requisitos de armazenamento e desempenho em multiplataformas.....	113
Requisitos de espaço em disco em multiplataformas.....	114
Planejando o suporte ao sistema de arquivos em multiplataformas.....	116
Planejando o suporte do sistema de arquivos para MFT em Multiplataformas.....	144
Escolhendo a criação de log circular ou linear em multiplataformas.....	145
memória compartilhada no AIX.....	145
Recursos IPC do IBM MQ e UNIX System V.....	146
Prioridade de processo do IBM MQ e do UNIX.....	146
Planning your IBM MQ environment on z/OS.....	146
Planning for your queue manager.....	147
Planning your channel initiator.....	176
Planning your queue sharing group (QSG).....	180
Planning for backup and recovery.....	193
Planning your z/OS UNIX environment.....	202
Planning for Advanced Message Security.....	202
Planning for Managed File Transfer.....	203
Planning to use the IBM MQ Console and REST API on z/OS	209
Avisos.....	211
Informações sobre a Interface de Programação.....	212
Marcas comerciais.....	213

Planejando uma arquitetura do IBM MQ


Ao planejar seu ambiente do IBM MQ, considere o suporte que o IBM MQ fornece para arquiteturas de gerenciador de filas único e múltiplo e para estilos de sistema de mensagem ponto a ponto e publicar/ assinar. Além disso, planeje os requisitos de recursos e sua utilização de recursos de criação de log e backup.

Sobre esta tarefa

Antes de planejar sua arquitetura do IBM MQ, familiarize-se com os conceitos básicos do IBM MQ. Veja [IBM MQ Visão geral técnica](#).

As arquiteturas do IBM MQ vão desde arquiteturas simples que usam um único gerenciador de filas, até redes mais complexas de gerenciadores de filas interconectados. Vários gerenciadores de filas são conectados juntos usando técnicas de enfileiramento distribuído. Para obter informações adicionais sobre como planejar arquiteturas de gerenciador de filas único e de diversos gerenciadores de filas, consulte os tópicos a seguir:

- [“Arquiteturas com base em um único gerenciador de filas”](#) na página 18
- [“Arquiteturas com base em vários gerenciadores de filas”](#) na página 19
 - [“Planejando filas distribuídas e clusters”](#) na página 20
 - [“Planejando sua rede publicar/assinar distribuída”](#) na página 73

 No IBM MQ for z/OS, é possível usar filas compartilhadas e grupos de compartilhamento de fila para possibilitar a implementação de balanceamento de carga de trabalho que e os aplicativos IBM MQ sejam escaláveis e altamente disponíveis. Para obter informações sobre filas compartilhadas e grupos de compartilhamento de filas, consulte [Filas compartilhadas e grupos de compartilhamento de filas](#).

O IBM MQ fornece dois modelos de liberação diferentes:

- A liberação Long Term Support (LTS) é mais adequada para sistemas que requerem uma implementação de longo prazo e estabilidade máxima.
- A liberação Continuous Delivery (CD) é destinada a sistemas que precisam explorar rapidamente os aprimoramentos funcionais mais recentes para o IBM MQ.

Ambos os tipos de liberação são instalados na mesma maneira, mas há considerações relacionadas ao suporte e migração que você precisa entender. Para obter mais informações, consulte [IBM MQ tipos de liberação e versão](#).

Para obter informações sobre planejamento para diversas instalações, requisitos de armazenamento e desempenho e uso de clientes, veja os outros subtópicos.

Conceitos relacionados

[Tipos de liberação e versão do IBM MQ](#)

[“Planning your IBM MQ environment on z/OS”](#) na página 146

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.

[Disponibilidade, recuperação e reinicialização](#)

Tarefas relacionadas

[Verificando Requisitos](#)

[Certificando-se de que Mensagens não Sejam Perdidas \(Criação de Log\)](#)

IBM MQ tipos de liberação: considerações de planejamento

Os dois tipos de liberação principais para IBM MQ são Long Term Support (LTS) e Continuous Delivery (CD). Para cada plataforma suportada, o tipo de liberação escolhido afeta o pedido, a instalação, a manutenção e a migração.

Para obter informações detalhadas sobre os tipos de liberação, consulte [IBM MQ tipos de liberação e versão](#).

Considerações para o IBM MQ for Multiplatforms



Classificação

No Passport Advantage há dois eAssemblies separados para IBM MQ 9.4. Um contém imagens de instalação para a liberação IBM MQ 9.4.0 Long Term Support e o outro contém imagens de instalação para a liberação IBM MQ 9.4.x Continuous Delivery. Faça download das imagens de instalação a partir do eAssembly de acordo com sua opção de liberação.

Todas as versões do IBM MQ e para o IBM MQ 9.4, tanto as liberações LTS quanto as liberações CD pertencem ao mesmo ID do produto.

A autorização para usar o IBM MQ estende-se por todo o produto (PID), sujeito às restrições de componentes licenciados e métricas de precificação. Isso significa que é possível escolher livremente entre LTS release e CD imagens de instalação de liberação para o IBM MQ 9.4.

Instalação

Depois de fazer download de uma imagem de instalação do Passport Advantage, é necessário selecionar para instalação apenas os componentes para os quais você comprou a autorização. Consulte [Informações sobre licença do IBM MQ](#) para obter informações adicionais sobre quais componentes instaláveis são incluídos para cada componente debitável.

É possível instalar a liberação IBM MQ 9.4.0 LTS e a liberação IBM MQ 9.4.x CD na mesma imagem do sistema operacional. Se você fizer isso, os componentes aparecerão como instalações separadas, conforme suportado pelo suporte de múltiplas versões do IBM MQ. Cada versão possui conjuntos distintos de gerenciadores de filas associados a essa versão.

Cada nova liberação CD é fornecida como uma imagem de instalação. A nova liberação do CD pode ser instalada ao lado de uma liberação existente ou uma liberação anterior do CD pode ser atualizada no local pelo instalador para a nova liberação.

As liberações do CD contêm aprimoramentos funcionais, bem como o conjunto mais recente de correções de defeitos e atualizações de segurança. Cada liberação do CD é acumulativa e substitui totalmente todas as anteriores para essa versão do IBM MQ. Portanto, será possível ignorar uma liberação específica do CD se ela não contiver nenhuma função relevante para sua empresa.

Manutenção

A liberação LTS é atendida pelo aplicativo de fix packs, que fornecem correções de defeitos e atualizações de segurança acumulativas (CSUs), que fornecem correções de segurança. Os fix packs e CSUs são disponibilizados periodicamente e acumulativos.

Para o CD, as CSUs são produzidas somente para a liberação mais recente do CD, que pode estar em uma versão subsequente.

Ocasionalmente, você pode ser orientado pela equipe de Suporte do IBM a aplicar uma correção temporária. As correções temporárias também são conhecidas como correções emergenciais ou de teste e são usadas para aplicar atualizações urgentes que não podem esperar pela próxima entrega de manutenção.

Migração entre a liberação do LTS e a liberação do CD

Há restrições e limitações, mas geralmente um único gerenciador de filas pode ser migrado do uso do código da liberação LTS para o código da liberação CD ou do uso do código da liberação CD para o

código da liberação LTS, contanto que a liberação de destino seja maior do que aquela em uso antes da migração.

Duas abordagens são possíveis:

- Instalar a nova liberação do código no lugar para que uma instalação existente do IBM MQ seja atualizada. Quaisquer gerenciadores de filas associados à instalação usam a nova liberação de código quando iniciados.
- Instalar a nova liberação de código como uma nova instalação, em seguida, mover as instâncias do gerenciador de filas individual para a nova instalação usando o comando `setmqm`.

Quando um gerenciador de filas inicia a execução de uma liberação do código do CD, o nível de comando do gerenciador de filas é atualizado para indicar o novo nível de liberação. Isso significa que quaisquer novas funções fornecidas na liberação estão ativadas e não é mais possível reiniciar o gerenciador de fila usando uma liberação de código com um número inferior de VRM.

Considerações para o IBM MQ for z/OS



Classificação

Ao classificar o IBM MQ for z/OS 9.4, dois recursos separados são oferecidos no ShopZ. Os recursos correspondem à liberação LTS e à liberação CD. Ambos os recursos são aplicáveis ao mesmo ID do produto (PID). É o ID do produto que está licenciado, portanto, no local onde um recurso está licenciado há autorização para usar o recurso alternativo se necessário. Ao pedir, você seleciona o recurso correspondente à liberação LTS ou CD.

Se você estiver selecionando produtos para inclusão em um ServerPac, não será possível escolher a liberação LTS e CD no mesmo pedido ServerPac, porque os produtos não podem ser instalados pelo SMP / E na mesma zona de destino.

Instalação

As liberações LTS e CD são fornecidas em conjuntos separados de FMIDs. Observe que esses FMIDs não podem ser instalados na mesma zona de destino SMP/E. Se você precisar das liberações LTS e CD:

- Instale a liberação LTS e a liberação CD em zonas de destino separadas.
- Mantenha bibliotecas de destino e distribuição separadas para as duas liberações.

Se seu gerenciador de filas estiver em um grupo de filas compartilhadas, ao fazer upgrade para a versão mais recente do CD, deve-se fazer upgrade de todos os gerenciadores de fila no grupo.

O nível de comando de um gerenciador de filas é o nível VRM de três dígitos. Um IBM MQ programa pode chamar `MQIA_COMMAND_LEVEL` seletor, para obter o nível de comando do gerenciador de filas ao qual ele está conectado.

Como as liberações usam FMIDs diferentes, não é possível atualizar uma liberação do CD com manutenção para uma liberação do LTS ou o contrário. Da mesma forma, não há como alternar uma versão do código do produto de uma liberação do LTS para uma liberação do CD ou o contrário. Entretanto, é possível alternar um gerenciador de fila entre os modelos de liberação. Consulte [Migração entre liberação do LTS e liberação do CD](#).

Nota:

As liberações do IBM MQ 9.0.x e do IBM MQ 9.1.x CD tinham versão diferente e FMIDs dependentes de liberação. Portanto, mover de 9.0.x CD para 9.1.x CD requer pelo menos uma instalação completa do SMP/E.

A partir do IBM MQ for z/OS 9.2.0, a liberação do CD usa um conjunto de FMIDs que permanecem os mesmos para todas as liberações do IBM MQ for z/OS com um número da versão de 9. Como cada nova versão do IBM MQ está disponível como uma liberação CD e uma LTS, é possível fazer upgrade de liberações CD aplicando PTFs para uma única instalação do SMP/E, mesmo ao cruzar um limite de versão principal. Por exemplo, é possível ir de IBM MQ for z/OS 9.2.0 CD para IBM MQ for z/OS 9.2.2 CD, para IBM MQ for z/OS 9.2.4 CD, para IBM MQ for z/OS 9.3.0 CD, apenas aplicando PTFs.

É possível distinguir entre uma liberação do LTS e CD com o mesmo nível de VRM olhando para a mensagem CSQY000I no log de tarefas do gerenciador de filas.

Manutenção

IBM MQ for z/OS usa PTFs para manutenção.

LTS As PTFs são específicas para um determinado conjunto de bibliotecas correspondentes a um nível de liberação específico. Para recursos do UNIX System Services (ou seja, JMS e WEB UI, Connector Pack e Managed File Transfer) as PTFs do z/OS são alinhadas diretamente com os fix packs do Multiplatforms e as atualizações de segurança acumulativas (CSUs). Essas correções são acumulativas e são disponibilizadas ao mesmo tempo que o fix pack ou CSU equivalente do Multiplatforms..

CD CD CSUs geralmente não são disponibilizados entre liberações de CD, mas são incluídos na próxima liberação do IBM MQ for z/OS CD . Também é possível entrar em contato com o suporte para solicitar um ++ USERMOD

Outras correções em IBM MQ for z/OS são correções distintas em partes específicas. Essas correções resolvem problemas específicos, não são acumulativas, e são disponibilizadas à medida que são produzidas e quando são.

Migração entre a liberação do LTS e a liberação do CD

Há restrições e limitações, mas geralmente um único gerenciador de filas pode ser migrado do uso do código da liberação LTS para o código da liberação CD ou do uso do código da liberação CD para o código da liberação LTS, contanto que a liberação de destino seja maior do que aquela em uso antes da migração.

A partir do IBM MQ for z/OS 9.2.0, é possível migrar para frente e para trás entre as liberações do CD e LTS com o mesmo VRM quantas vezes forem necessárias e sem impacto na capacidade de migrar para trás. Por exemplo, um gerenciador de filas pode ser iniciado em IBM MQ for z/OS 9.3.0 LTS, em seguida, encerrado e iniciado em IBM MQ for z/OS 9.3.0 CD, em seguida, encerrado e iniciado em IBM MQ for z/OS 9.3.0 LTS.

O IBM MQ for z/OS tem tradicionalmente fornecido um recurso de fallback (migração para trás) para que, após um período de execução após uma migração, seja possível efetuar fallback para a liberação anterior. Esse recurso é retido para liberações do LTS e essas liberações do CD com um modificador de 0 , como 9.3.0 CD, mas não é possível quando a origem ou o destino de uma migração é uma liberação do CD com um número de modificador diferente de zero, por exemplo, 9.2.5 ou 9.3.1.

A seguir estão os cenários de migração válidos e ilustram como esse princípio funciona:

Liberação de origem	Liberação de destino	Notas
LTS 9.1.0	9.4.0 LTS ou 9.4.0 CD	Migração para versão anterior não suportada, pois o LTS 9.1.0 está fora do suporte padrão.
9.2.0 LTS	9.4.0 LTS ou 9.4.0 CD	Migração para versão anterior suportada.
9.3.0 LTS	9.4.0 LTS ou 9.4.0 CD	Migração para versão anterior suportada.
9.3.5 CD	9.4.0 LTS ou 9.4.0 CD	A migração para versão anterior não é suportada já que a liberação de origem é um CD com um modificador diferente de zero.

Liberação de origem	Liberação de destino	Notas
9.4.0 LTS ou 9.4.0 CD	9.4.1 CD	<p>A migração para versão anterior não é suportada já que a liberação de destino é um CD com um modificador diferente de zero.</p> <p>O Write to operator with reply <code>CSQY041D</code> é emitido para confirmar a migração</p>

Tarefas relacionadas

 Aplicando e removendo a manutenção no z/OS

Informações relacionadas

[Baixando IBM MQ 9.4](#)

Considerações sobre o IBM MQ e o IBM MQ Appliance no local para adequação à GDPR

Para PID (s):

Distributed

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

z/OS

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

IBM MQ Appliance

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

Aviso:

Este documento é destinado a ajudá-lo em suas preparações para prontidão de GDPR. Ele fornece informações sobre recursos do IBM MQ que você pode configurar e os aspectos do uso do produto que você deve considerar para ajudar a sua organização com a prontidão do GDPR. Essas informações não são uma lista exaustiva, pois há muitas maneiras de escolher e configurar recursos e de usar o produto sozinho e com aplicativos e sistemas de terceiro.

Os clientes são responsáveis por assegurar a sua própria conformidade com as várias leis e regulamentações, incluindo a General Data Protection Regulation da União Europeia. Os clientes são os únicos responsáveis por obter conselho de consultoria jurídica competente quanto à identificação e interpretação de quaisquer leis e regulamentações relevantes que possam afetar os negócios do cliente e quaisquer ações que os clientes possam precisar tomar para obedecer tais leis e regulamentações.

Os produtos, serviços e outros recursos descritos neste documento não são adequados para todas as situações do cliente e podem ter disponibilidade restringida. A IBM não fornece aviso legal, contábil ou de auditoria nem representa ou garante que seus serviços ou produtos assegurarão que os clientes estejam em conformidade com qualquer lei ou regulamentação.

Tabela de Conteúdo

1. [GDPR](#)
2. [Configuração do Produto para GDPR](#)
3. [Ciclo de Vida de Dados](#)
4. [Coleta de dados](#)
5. [Armazenamento de dados](#)
6. [Acesso a Dados](#)
7. [Processamento de dados](#)
8. [Exclusão de dados](#)
9. [Monitoramento de dados](#)
10. [Capacidade para restringir o uso de dados pessoais](#)
11. [Manipulação de arquivo](#)

GDPR

O General Data Protection Regulation (GDPR) foi adotado pela União Europeia ("UE") e é aplicável a partir de 25 de maio de 2018.

Por que GDPR é importante?

O GDPR estabelece uma estrutura regulamentar de proteção de dados mais forte para o processamento de dados pessoais de indivíduos. O GDPR traz:

- Direitos Novos e Aprimorados para Indivíduos
- Definição de dados pessoais mais ampla
- Novas obrigações para os transformadores
- Potencial para multas financeiras significativas por não conformidade
- Notificação de violação de dados obrigatória

Leia mais sobre o GDPR:

- [Portal de Informações GDPR da UE](#)
- [website do ibm.com/GDPR](http://www.ibm.com/GDPR)

Configuração do produto - considerações sobre adequação à GDPR

As seções a seguir fornecem considerações para configurar o IBM MQ para ajudar a sua organização com a prontidão do GDPR.

Ciclo de Vida de Dados

O IBM MQ é um produto de middleware orientado a mensagens transacionais que permite que os aplicativos troquem dados fornecidos pelos aplicativos assincronamente. O IBM MQ suporta um intervalo de APIs, protocolos e pontes do sistema de mensagens para o propósito de conectar aplicativos. Como tal, o IBM MQ pode ser usado para trocar muitos formulários de dados, alguns dos quais podem estar potencialmente sujeitos à GDPR. Há vários produtos de terceiros com os quais o IBM MQ pode trocar dados. Alguns deles são de propriedade da IBM, mas muitos outros são fornecidos por outros fornecedores de tecnologia. O [website de Relatórios de compatibilidade do produto de software](#) fornece listas do software associado. Para considerações sobre a adequação à GDPR de um produto de terceiro, é necessário consultar a documentação desse produto. Os administradores do IBM MQ controlam a maneira na qual o IBM MQ interage com os dados que passam por ele, pela definição de filas, tópicos e assinaturas.

Quais tipos de fluxo de dados por meio de IBM MQ?

Como o IBM MQ fornece o serviço de sistema de mensagens assíncrono para dados do aplicativo, não há nenhuma resposta definitiva para essa pergunta porque os casos de uso variam na implementação do aplicativo. Os dados da mensagem do aplicativo são persistidos em arquivos de fila (conjuntos de páginas ou o Recurso de acoplamento no z/OS), logs e archives e a própria mensagem podem conter dados que são controlados pelo GDPR. Os dados da mensagem fornecidos pelo aplicativo também podem ser incluídos em arquivos coletados para propósitos de determinação de problema, como logs de erro, arquivos de rastreamento e FFSTs. Os dados da mensagem fornecidos pelo aplicativo do z/OS também podem ser incluídos nos dumps de espaço de endereço ou de Recurso de Acoplamento.

A seguir estão alguns exemplos típicos de dados pessoais que podem ser trocados usando o IBM MQ:

- Funcionários do cliente (por exemplo, o IBM MQ pode ser usado para conectar os sistemas de folha de pagamento ou de RH do cliente)
- Os dados pessoais dos próprios clientes do cliente (por exemplo, o IBM MQ pode ser usado por um cliente para trocar dados entre aplicativos que estão relacionados a seus clientes, como obter oportunidades de vendas e armazenar dados dentro de seu sistema CRM).
- Os dados pessoais sensíveis dos próprios clientes do cliente (por exemplo, o IBM MQ pode ser usado dentro de contextos do segmento de mercado que requerem que dados pessoais sejam trocados, como registros de assistência médica baseados em HL7 ao integrar aplicativos clínicos).

Além dos dados da mensagem fornecidos pelo aplicativo, o IBM MQ processa os tipos de dados a seguir:

- Credenciais de autenticação (como nome do usuário e senhas, chaves API, etc.)
- informações pessoais tecnicamente identificáveis (como IDs de dispositivo, identificadores baseados em uso, endereço IP, etc. - quando vinculadas a um indivíduo)

Dados pessoais usados para contato on-line com a IBM

Os clientes do IBM MQ podem enviar comentários/feedback/solicitações on-line para entrar em contato com a IBM sobre assuntos do IBM MQ de várias maneiras, principalmente:

- Área de comentários públicos em páginas na [área do IBM MQ no IBM Developer](#)
- Área de comentários públicos em páginas de informações do produto [IBM MQ em IBM Documentation](#)
- Comentários públicos nos [Fóruns de Suporte IBM](#)
- Comentários públicos no [IBM Integration Ideas](#)

Geralmente, apenas o nome e o endereço de e-mail do cliente são usados para ativar respostas pessoais para o assunto do contato e o uso de dados pessoais em conformidade com o [IBM IBM Online Privacy Statement](#).

Coleta de Dados

O IBM MQ pode ser usado para coletar dados pessoais. Ao avaliar seu uso do IBM MQ e suas necessidades para atender às demandas de GDPR, é necessário considerar os tipos de dados pessoais que, em suas circunstâncias, estão passando pelo IBM MQ. Você pode desejar considerar aspectos como:

- Como os dados chegam aos seus gerenciadores de filas? (Ao longo de quais protocolos?) Os dados são criptografados? Os dados são assinados?)
- Como os dados são enviados de seus gerenciadores de filas? (Ao longo de quais protocolos?) Os dados são criptografados? Os dados são assinados?)
- Como os dados são armazenados à medida que passam por um gerenciador de filas? (Qualquer aplicativo de sistema de mensagens tem o potencial de gravar dados da mensagem em mídia stateful, mesmo se uma mensagem não for persistente. Você está ciente de como os recursos do sistema de mensagens podem potencialmente expor aspectos dos dados de mensagem do aplicativo que passam pelo produto?)
- Como as credenciais são coletadas e armazenadas quando necessário pelo IBM MQ para acessar aplicativos de terceiros?

O IBM MQ pode precisar se comunicar com outros sistemas e serviços que requerem autenticação, por exemplo, LDAP. Quando necessário, os dados de autenticação (IDs de usuário, senhas) são configurados e

armazenados pelo IBM MQ para seu uso em tais comunicações. Sempre que possível, é necessário evitar usar credenciais pessoais para autenticação do IBM MQ. Considere a proteção do armazenamento usado para dados de autenticação. (Consulte o Armazenamento de Dados a seguir.)

Armazenamento de Dados

Quando os dados da mensagem atravessarem os gerenciadores de filas, o IBM MQ persistirá (talvez múltiplas cópias) esses dados diretamente na mídia stateful. Os usuários do IBM MQ podem desejar considerar a proteção dos dados da mensagem enquanto eles estão em repouso.

Os itens a seguir destacam áreas em que o IBM MQ persiste os dados fornecidos pelo aplicativo, que os usuários podem desejar considerar ao assegurar a conformidade com a GDPR.

- **Filas de Mensagens do Aplicativo:**

O IBM MQ fornece filas de mensagens para permitir a troca de dados assíncrona entre os aplicativos. As mensagens persistentes e não persistentes armazenadas em uma fila são gravadas em mídia stateful.
- **Filas do Agente de Transferência de Arquivos:**

O IBM MQ Managed File Transfer utiliza filas de mensagens para coordenar a transferência confiável de dados do arquivo; os arquivos que contêm dados pessoais e registros de transferências são armazenados nessas filas.
- **Filas de Transmissão:**

Para transferir mensagens confiavelmente entre os gerenciadores de filas, as mensagens são armazenadas temporariamente nas filas de transmissão.
- **Filas de Devoluções:**

Há algumas circunstâncias em que as mensagens não podem ser colocadas em uma fila de destino e são armazenadas em uma fila de mensagens não entregues, caso tal fila esteja configurada no gerenciador de filas.
- **Filas de Backout:**

As interfaces de sistema de mensagens JMS e XMS fornecem um recurso que permite que mensagens suspeitas sejam movidas para uma fila de restauração após um número de restaurações ter ocorrido para permitir que outras mensagens válidas sejam processadas.
- **Fila de erros do AMS:**

O IBM MQ Advanced Message Security moverá as mensagens que não estão em conformidade com uma política de segurança para o SYSTEM.PROTECTION.ERROR.QUEUE é semelhante ao enfileiramento de mensagens não entregues.
- **Publicações Retidas:**

O IBM MQ fornece um recurso de publicação retida para permitir que os aplicativos de assinatura façam a chamada de uma publicação anterior.
- **Entrega adiada:**

O IBM MQ suporta o recurso de atraso de entrega JMS 2.0 e Jakarta Messaging 3.0 que permite que as mensagens sejam entregues para seu destino em um momento futuro. As mensagens que ainda não foram entregues estão armazenadas na fila SYSTEM.DDELAY.LOCAL.QUEUE.

Leia mais:

- [Criação de log: certificando-se de que as mensagens não sejam perdidas](#)
- [Configurações da Fila do Agente MFT](#)
- [Usando a fila de devoluções](#)
- [Manipulando mensagens suspeitas em classes do IBM MQ para JMS](#)
- [Manipulação de Erros AMS](#)
- [Publicações retidas](#)

- [JMS 2.0 atraso de entrega](#)

Os itens a seguir destacam áreas em que o IBM MQ pode persistir indiretamente os dados fornecidos pelo aplicativo que os usuários também podem desejar considerar ao assegurar a conformidade com a GDPR.

- Mensagem de Rastreamento de Rota:

O IBM MQ fornece recursos de rota de rastreamento, que registram a rota que uma mensagem usa entre os aplicativos. As mensagens de eventos geradas podem incluir informações pessoais tecnicamente identificáveis, como endereços IP.

- Rastreamento de atividade do aplicativo:

O IBM MQ fornece rastreamento de atividade do aplicativo, que registra as atividades da API do sistema de mensagens de aplicativos e canais; o rastreamento de atividade do aplicativo pode registrar os conteúdos dos dados da mensagem fornecidos pelo aplicativo para mensagens de eventos.

- Rastreamento de serviço:

O IBM MQ fornece recursos de rastreamento de serviço, que registram os caminhos de código interno por meio dos quais os dados da mensagem fluem. Como parte desses recursos, o IBM MQ pode registrar os conteúdos dos dados da mensagem fornecidos pelo aplicativo para rastrear arquivos armazenados em disco.

- Eventos do gerenciador de filas:

O IBM MQ pode gerar mensagens de eventos que podem incluir dados pessoais, como eventos de autoridade, comando e configuração.

Leia mais:

- [Sistema de Mensagens de Rastreamento de Rotas](#)
- [Utilizando Rastreamento](#)
- [Monitoramento de Eventos](#)
- [Eventos do Gerenciador de Filas](#)

Para proteger o acesso a cópias dos dados da mensagem fornecidos pelo aplicativo, considere as ações a seguir:

- Restrinja o acesso de usuário privilegiado aos dados do IBM MQ no sistema de arquivos, por exemplo, restringindo a participação do usuário do grupo 'mqm' em plataformas UNIX and Linux®.
- Restrinja o acesso de aplicativo aos dados do IBM MQ por meio de filas dedicadas e controle de acesso. Quando apropriado, evite o compartilhamento desnecessário de recursos, como filas entre aplicativos, e forneça controle de acesso granular para recursos de fila e tópico.
- Restrinja o acesso a cópias replicadas de dados do IBM MQ em configurações de alta disponibilidade (HA) ou de recuperação de desastre (DR) e proteja as conexões usadas para replicação.
- Use o IBM MQ Advanced Message Security para fornecer assinatura e/ou criptografia de ponta a ponta de dados da mensagem.
- Use a criptografia de nível de arquivo ou de volume para proteger diretórios ou sistemas de arquivos que podem conter dados, rastreamento ou logs do IBM MQ
- Depois de fazer upload do rastreamento de serviço para a IBM, será possível excluir os arquivos de rastreamento de serviço e os dados FFST se você estiver preocupado com os conteúdos que possuem potencialmente dados pessoais.

Leia mais:

- [Usuários Privilegiados](#)
- [Planejando o suporte ao sistema de arquivos em multiplataformas](#)
- [Criptografia do sistema de arquivos no IBM MQ Appliance](#)

Um administrador do IBM MQ pode configurar um gerenciador de filas com credenciais (nome do usuário e senha, chaves API, etc.) para serviços de terceiros como LDAP. Esses dados geralmente são

armazenados no diretório de dados do gerenciador de filas protegido por meio de permissões do sistema de arquivos.

Quando um gerenciador de filas do IBM MQ é criado, o diretório de dados é configurado com o controle de acesso baseado em grupo, de forma que o IBM MQ possa ler os arquivos de configuração e usar as credenciais para se conectar a esses sistemas. Os administradores do IBM MQ são considerados usuários privilegiados e são membros desse grupo, portanto, têm acesso de leitura aos arquivos. Alguns arquivos são ofuscados, mas eles não são criptografados. Por essa razão, para proteger totalmente o acesso às credenciais, é necessário considerar as ações a seguir:

- Restrinja o acesso de usuário privilegiado aos dados do IBM MQ, por exemplo, restringindo a participação do grupo 'mqm' em plataformas UNIX and Linux.
- Use a criptografia de nível de arquivo ou de volume para proteger os conteúdos do diretório de dados do gerenciador de filas.
- Criptografe backups do diretório de configuração de produção e armazene-os com controles de acesso apropriados.
- Considere fornecer trilhas de auditoria para falha de autenticação, controle de acesso e mudanças na configuração com eventos de segurança, comando e configuração.

Leia mais:

- [Protegendo o IBM MQ](#)

Acesso a Dados

Os dados do gerenciador de filas do IBM MQ podem ser acessados por meio das interfaces do produto a seguir, algumas das quais são projetadas para acesso por meio de uma conexão remota e outras para acesso por meio de uma conexão local.

- IBM MQ Console [Somente Remoto]
- API de REST administrativa do IBM MQ [somente remota]
- API de REST do sistema de mensagens do IBM MQ [somente remota]
- MQI [Local e Remoto]
- JMS [Local e Remoto]
- XMS [Local e Remoto]
- IBM MQ Telemetry (MQTT) [Apenas Remoto]
- IBM MQ Light (AMQP) [apenas remoto]
- Ponte IMS do IBM MQ [Apenas Local]
- IBM MQ Ponte CICS [Apenas Local]
- IBM MQ Pontes de Protocolo MFT [Somente Remoto]
- IBM MQ Connect: Pontes Diretas [Apenas Remoto]
- IBM MQ MQAI [Local e Remoto]
- IBM MQ Comandos PCF [Local e Remoto]
- IBM MQ Comandos MQSC [Local e Remoto]
- IBM MQ Explorer [Local e Remoto]
- Saídas de usuário do IBM MQ [somente local]
- IBM MQ Internet Pass-Thru [somente remoto]
- Métricas de monitoramento Red Hat® OpenShift® (Prometheus) (as métricas são dados numéricos sobre estatísticas do gerenciador de filas)
- Console serial do IBM MQ Appliance [somente local]
- SSH do IBM MQ Appliance [somente remoto]
- IBM MQ Appliance API de REST [Somente Remoto]

- IU da web do IBM MQ Appliance [somente remota]
- **V 9.4.0** IBM MQ Kafka Conectores (Kafka Connect) [Local e Remoto]

As interfaces são projetadas para permitir que os usuários façam mudanças em um gerenciador de filas do IBM MQ e em mensagens armazenadas nele. As operações de administração e de sistema de mensagens são protegidas de modo que haja três estágios envolvidos quando uma solicitação for feita;

- Autenticação
- Mapeamento de função
- Autorização

Autenticação:

Se a mensagem ou operação administrativa foi solicitada por meio de uma conexão local, a origem dessa conexão será um processo em execução no mesmo sistema. O usuário que está executando o processo deve ter passado por quaisquer etapas de autenticação fornecidas pelo sistema operacional. O nome do usuário do proprietário do processo por meio do qual a conexão foi feita é declarado como a identidade. Por exemplo, esse pode ser o nome do usuário que está executando o shell por meio do qual um aplicativo foi iniciado. As formas possíveis de autenticação para conexões locais são:

1. Nome do Usuário Asserido (S.O. local)
2. Nome do usuário e senha opcionais (S.O., LDAP ou repositórios de terceiros customizados)
3. Token de Segurança (JWT) IBM MQ apenas

Se a ação administrativa foi solicitada por meio de uma conexão remota, as comunicações com o IBM MQ serão feitas por meio de uma interface de rede. As formas de identidade a seguir podem ser apresentadas para autenticação por meio de conexões de rede;

1. Nome do usuário asserido (do S.O. remoto)
2. Nome do usuário e senha (S.O., LDAP ou repositórios de terceiros customizados)
3. Endereço de rede de origem (como endereço IP)
4. Certificado X.509 Digital (autenticação mútua de SSL/TLS)
5. Tokens de segurança (como token LTPA2 ou token JWT)
6. Outra segurança customizada (recurso fornecido por saídas de terceiros)
7. Chaves SSH

A integração do IBM MQ com o IBM Cloud Pak for Integration inclui um novo tipo de autenticação para o IBM MQ Console: Conexão única com o Cloud Pak. (apenas CP4I)

Mapeamento de função:

No estágio de mapeamento de função, as credenciais que foram fornecidas no estágio de autenticação podem ser mapeadas para um identificador de usuário alternativo. Contanto que o identificador de usuário mapeado tenha permissão para continuar (por exemplo, usuários administrativos podem ser bloqueados por regras de autenticação de canal), o ID do usuário mapeado é transportado para o estágio final ao autorizar atividades com relação aos recursos do IBM MQ.

Autorização:

O IBM MQ fornece a capacidade para diferentes usuários terem diferentes autoridades com relação a diferentes recursos do sistema de mensagens, como filas, tópicos e outros objetos do gerenciador de filas.

Atividade de criação de log:

Alguns usuários do IBM MQ podem precisar criar um registro de auditoria de acesso aos recursos do MQ. Exemplos de logs de auditoria desejados podem incluir mudanças na configuração que contêm informações sobre a mudança além de quem a solicitou.

As origens de informações a seguir estão disponíveis para implementar esse requisito:

1. Um gerenciador de filas do IBM MQ pode ser configurado para produzir eventos de comando quando um comando de administrador tiver sido executado com êxito.
2. Um gerenciador de filas do IBM MQ pode ser configurado para produzir eventos de configuração quando um recurso do gerenciador de filas é criado, alterado ou excluído.
3. Um gerenciador de filas do IBM MQ pode ser configurado para produzir um evento de autoridade quando uma verificação de autorização falha para um recurso.
4. As mensagens de erro indicando verificações de autorização com falha são gravadas nos logs de erro do gerenciador de filas.
5. O Console IBM MQ gravará mensagens de auditoria em seus logs quando a autenticação, as verificações de autorização falharem ou quando os gerenciadores de filas forem criados, iniciados, parados ou excluídos.
6. O IBM MQ Appliance gravará mensagens de auditoria em seus logs para registrar os logins do usuário e as mudanças no sistema.

Ao considerar esse tipo de solução, os usuários do IBM MQ podem desejar considerar os pontos a seguir:

- As mensagens de eventos são não persistentes, portanto, quando um gerenciador de filas é reiniciado, as informações são perdidas. Quaisquer monitores de eventos devem ser configurados para consumir constantemente quaisquer mensagens disponíveis e transferir o conteúdo para a mídia persistente.
- Os usuários privilegiados do IBM MQ têm privilégios suficientes para os eventos desativados, limpar logs ou excluir gerenciadores de filas.

Para obter mais informações sobre como proteger o acesso aos dados do IBM MQ e fornecer uma trilha de auditoria, consulte os tópicos a seguir:

- [Mecanismos de Segurança do IBM MQ](#)
- [Eventos de configuração](#)
- [Eventos de comando](#)
- [Usando logs de erro](#)

Processamento de Dados

Criptografia usando uma Infra-estrutura de Chave Pública:

É possível proteger conexões de rede com o IBM MQ especificando que as conexões usam TLS, que também pode fornecer autenticação mútua do lado iniciador da conexão.

O uso dos recursos de segurança PKI que são fornecidos pelos mecanismos de transporte é a primeira etapa para proteger o processamento de dados com o IBM MQ. No entanto, sem ativar os recursos de segurança adicionais, o comportamento de um aplicativo consumidor é processar todas as mensagens entregues a ele sem validar a origem da mensagem ou se foi alterada enquanto em trânsito.

Os usuários do IBM MQ que são licenciados para usar os recursos do Advanced Message Security (AMS) podem controlar a maneira na qual os aplicativos processam os dados pessoais retidos em mensagens, por meio da definição e configuração de políticas de segurança. As políticas de segurança permitem que a assinatura e/ou criptografia digital sejam aplicadas aos dados da mensagem entre aplicativos.

É possível usar políticas de segurança para requerer e validar uma assinatura digital ao consumir mensagens para assegurar que as mensagens sejam autênticas. A criptografia do AMS fornece um método pelo qual os dados da mensagem são convertidos de um formato legível para uma versão codificada que pode ser decodificada por outro aplicativo somente se ele for o destinatário-alvo ou a mensagem e tiver acesso à chave de descriptografia correta.

Para obter mais informações sobre o uso de SSL e de certificados para proteger suas conexões de rede, consulte os tópicos a seguir na documentação do produto IBM MQ:

- [Configurando a Segurança do TLS para o IBM MQ](#)
- [Visão Geral do AMS](#)

Exclusão de Dados

O IBM MQ fornece comandos e ações da interface com o usuário para excluir dados que foram fornecidos para o produto. Isso significa que os usuários do IBM MQ podem excluir dados que se relacionam a indivíduos específicos, caso isso seja necessário.

- Áreas de comportamento do IBM MQ a serem consideradas para obedecer à exclusão de dados de cliente da GDPR
 - Exclua dados da mensagem armazenados em uma fila do aplicativo da forma a seguir:
 - Removendo mensagens individuais, usando a API do sistema de mensagens ou o conjunto de ferramentas ou usando a validação de mensagem.
 - Especificando que as mensagens não são persistentes, retidas em uma fila na qual a classe de mensagem não persistente é normal e reiniciando o gerenciador de filas.
 - Limpando administrativamente a fila.
 - Excluindo a fila.
 - Exclua os dados da publicação retida armazenados em um tópico da forma a seguir:
 - Especificando que as mensagens não são persistentes e reiniciando o gerenciador de filas.
 - Substituindo os dados retidos por novos dados ou usando a validação de mensagem.
 - Limpando administrativamente a sequência de tópicos.
 - Exclua dados armazenados em um gerenciador de filas ao excluindo o gerenciador de filas inteiro e quaisquer cópias replicadas para alta disponibilidade ou recuperação de desastre.
 - Exclua os dados armazenados pelos comandos de rastreamento de Serviço, excluindo os arquivos no diretório de rastreamento.
 - Exclua os dados FFST armazenados, excluindo os arquivos no diretório de erros.
 - Exclua os dumps de espaço de endereço e de Recurso de Acoplamento (no z/OS).
 - Exclua o archive, o backup ou outras cópias de tais dados.
- Áreas de comportamento do IBM MQ a serem consideradas para obedecer à exclusão de dados da conta da GDPR
 - É possível excluir dados da conta e preferências armazenadas pelo IBM MQ para se conectar a gerenciadores de filas e serviços de terceiros, excluindo (inclusive archive, backup ou cópias replicadas disso):
 - Objetos de informações sobre autenticação do gerenciador de filas que armazenam credenciais.
 - Registros de autoridade do gerenciador de filas que referenciam identificadores de usuários.
 - Regras de autenticação de canal do gerenciador de filas que mapeiam ou bloqueiam endereços IP específicos, DNS do certificado ou identificadores de usuários.
 - Arquivos de credenciais usados pelo Agente do IBM MQ Managed File Transfer, Criador de Logs e Plug-in do MQ Explorer MFT para autenticação com gerenciador de filas e servidores de arquivos.
 - Certificados digitais X.509 que representam ou contêm informações sobre um indivíduo de keystores que podem ser usados por conexões SSL/TLS ou IBM MQ Advanced Message Security (AMS).
 - Contas de usuários individuais do IBM MQ Appliance, incluindo referência a essas contas em arquivos de log do sistema.
 - Metadados da área de trabalho do IBM MQ Explorer e configurações do Eclipse.
 - Armazém de senhas do IBM MQ Explorer conforme especificado nas [Preferências de senha](#).
 - Arquivos de configuração do IBM MQ Console e do servidor mqweb.
 - Arquivos de configuração e keystores do IBM MQ Internet Pass-Thru.

Leia mais:

- [Autenticação de conexão do MFT e do IBM MQ](#)

- [Mapeando Credenciais para um Servidor de Arquivos Usando o Arquivo ProtocolBridgeCredentials.xml](#)
- [Configurando IBM MQ Console usuários e funções](#)

Monitoramento de Dados

O IBM MQ fornece uma faixa de recursos de monitoramento que os usuários podem explorar para ter um entendimento melhor de como os aplicativos e gerenciadores de filas estão sendo executados.

O IBM MQ também fornece vários recursos que ajudam a gerenciar logs de erros do gerenciador de filas.

Leia mais:

- [Monitorando sua rede do IBM MQ](#)
- [Serviços de mensagem de diagnóstico](#)
- [Serviço QMErrorLog](#)
- [Monitoramento e relatório do IBM MQ Appliance](#)

Capacidade para restringir o uso de dados pessoais

Usando os recursos resumidos neste documento, o IBM MQ permite que um usuário final restrinja o uso de seus dados pessoais.

As filas de mensagens do IBM MQ não devem ser usadas como um armazenamento de dados permanentes da mesma maneira que um banco de dados, que é particularmente verdadeiro ao manipular dados do aplicativo que estão sujeitos à GDPR.

Diferentemente de um banco de dados no qual os dados podem ser localizados por meio de uma consulta de procura, pode ser difícil localizar dados da mensagem, a menos que você saiba os identificadores de fila, mensagem e correlação de uma mensagem.

Contanto que as mensagens que contêm dados de um indivíduo possam ser prontamente identificadas e localizadas, é possível usar recursos de sistema de mensagens padrão do IBM MQ para acessar ou modificar dados da mensagem.

Manipulação de arquivo

1. O IBM MQ Managed File Transfer não executa a varredura de malware em arquivos transferidos. Os arquivos são transferidos no estado em que se encontram, e uma verificação de integridade é executada para garantir que os dados do arquivo não sejam modificados durante a transferência. As somas de verificação de origem e de destino são publicadas como parte da publicação de status de transferência. É recomendado que os usuários finais implementem a varredura de malware conforme apropriado para seu ambiente antes que o MFT transfira o arquivo e após o MFT entregar um arquivo para um terminal remoto.
2. O IBM MQ Managed File Transfer não executa ações baseadas no tipo MIME nem na extensão do arquivo. O MFT lê os arquivos e transfere os bytes exatamente como lidos do arquivo de entrada.

Arquiteturas com base em um único gerenciador de filas

A arquitetura mais simples do IBM MQ envolve a configurar e usar de um único gerenciador de filas.

Antes de planejar sua arquitetura do IBM MQ, familiarize-se com os conceitos básicos do IBM MQ. Veja [IBM MQ Visão geral técnica](#).

Várias arquiteturas possíveis usando um único gerenciador de filas são descritas nas seções a seguir:

- [“Gerenciador de filas único com aplicativos locais acessando um serviço” na página 19](#)
- [“Gerenciador de filas simples com aplicativos remotos acessando um serviço como clientes” na página 19](#)
- [“Gerenciador de filas simples com uma configuração de publicação/assinatura” na página 19](#)

Gerenciador de filas único com aplicativos locais acessando um serviço

A primeira arquitetura com base em um único gerenciador de filas é quando os aplicativos que acessam um serviço estão em execução no mesmo sistema que o aplicativo que fornece o serviço. Um gerenciador de filas do IBM MQ fornece intercomunicação assíncrona entre os aplicativos que solicitam o serviço e os aplicativos que fornecem o serviço. Isso significa que a comunicação entre os aplicativos pode continuar mesmo que um dos aplicativos esteja off-line por um período de tempo prolongado.

Gerenciador de filas simples com aplicativos remotos acessando um serviço como clientes

A segunda arquitetura com base em um gerenciador de filas único tem os aplicativos em execução remotamente a partir do aplicativo que fornece o serviço. Os aplicativos remotos estão sendo executados em sistemas diferentes daqueles dos serviços. Os aplicativos conectam-se como clientes ao gerenciador de filas único. Isso significa que o acesso a um serviço pode ser fornecido para vários sistemas através de um único gerenciador de filas.

Uma limitação dessa arquitetura é que uma conexão de rede deve estar disponível para um aplicativo operar. A interação entre o aplicativo e o gerenciador de filas sobre a conexão de rede é síncrona.

Gerenciador de filas simples com uma configuração de publicação/assinatura

Uma arquitetura alternativa usando um único gerenciador de filas é usar uma configuração de publicação/assinatura. No sistema de mensagens de publicação/assinatura, você pode separar o provedor de informações dos consumidores dessas informações. Isso difere dos estilos de sistema de mensagens ponto a ponto nas arquiteturas descritas anteriormente, em que os aplicativos devem saber informações sobre o aplicativo de destino, por exemplo, o nome da fila na qual colocar mensagens. Usando a publicação/assinatura de IBM MQ, o aplicativo de envio publica uma mensagem com um tópico especificado com base no assunto das informações. O IBM MQ manipula a distribuição da mensagem para aplicativos que registraram interesse nesse assunto por meio de uma assinatura. Os aplicativos de recebimento também não precisam saber nada sobre a origem das mensagens para recebê-las. Para obter mais informações, consulte [Publicar/assinar mensagens](#) e [Exemplo de uma configuração de publicação/assinatura de gerenciador de filas única](#).

Conceitos relacionados

[Introdução ao IBM MQ](#)

Tarefas relacionadas

[“Planejando uma arquitetura do IBM MQ” na página 5](#)

Ao planejar seu ambiente do IBM MQ, considere o suporte que o IBM MQ fornece para arquiteturas de gerenciador de filas único e múltiplo e para estilos de sistema de mensagem ponto a ponto e publicar/assinar. Além disso, planeje os requisitos de recursos e sua utilização de recursos de criação de log e backup.

[Criando e gerenciando gerenciadores de filas em Multiplataformas](#)

Arquiteturas com base em vários gerenciadores de filas

É possível usar mensagens técnicas de enfileiramento distribuído para criar uma arquitetura do IBM MQ que envolva a configuração e o uso de vários gerenciadores de filas.

Antes de planejar sua arquitetura do IBM MQ, familiarize-se com os conceitos básicos do IBM MQ. Veja [IBM MQ Visão geral técnica](#).

Uma arquitetura do IBM MQ pode ser alterada, sem mudança para aplicativos que fornecem serviços, incluindo gerenciadores de filas adicionais.

Os aplicativos podem ser hospedados na mesma máquina que um gerenciador de filas e, em seguida, obter comunicação assíncrona com um serviço hospedado em outro gerenciador de filas em outro sistema. Como alternativa, os aplicativos que estão acessando um serviço podem se conectar como

clientes a um gerenciador de filas que então fornece acesso assíncrono para o serviço em outro gerenciador de filas.

Roteamentos que conectam gerenciadores de filas diferentes e suas filas são definidos usando técnicas de enfileiramento distribuído. Os gerenciadores de filas dentro da arquitetura são conectados usando canais. Canais são usados para mover as mensagens automaticamente de um gerenciador de filas para outro em uma direção, dependendo da configuração dos gerenciadores de filas.

Para obter uma visão geral de alto nível de planejamento de uma rede do IBM MQ, veja [“Projetando redes do gerenciador de filas distribuído”](#) na página 21.

Para obter informações sobre como planejar canais para a sua arquitetura do IBM MQ, consulte [Técnicas de enfileiramento distribuído do IBM MQ](#).

O gerenciamento de filas distribuído permite criar e monitorar a comunicação entre gerenciadores de filas. Para obter mais informações sobre gerenciamento de filas distribuídas, veja [Introdução ao gerenciamento de filas distribuídas](#).

Tarefas relacionadas

[“Planejando uma arquitetura do IBM MQ”](#) na página 5

Ao planejar seu ambiente do IBM MQ, considere o suporte que o IBM MQ fornece para arquiteturas de gerenciador de filas único e múltiplo e para estilos de sistema de mensagem ponto a ponto e publicar/ assinar. Além disso, planeje os requisitos de recursos e sua utilização de recursos de criação de log e backup.

[Criando e gerenciando gerenciadores de filas em Multiplataformas](#)

Planejando filas distribuídas e clusters

É possível conectar manualmente filas hospedadas em gerenciadores de filas distribuídas ou você criar um cluster de gerenciador de filas e permitir que o produto conecte os gerenciadores de filas para você. Para escolher uma topologia adequada para sua rede do sistema de mensagens distribuído, considere seus requisitos de controle manual, tamanho da rede, frequência de mudança, disponibilidade e escalabilidade.

Antes de começar

Esta tarefa supõe que você entenda o que são redes de mensagens distribuídas e como elas funcionam. Para uma visão geral técnica, consulte [Enfileiramento e clusters distribuídos](#).

Sobre esta tarefa

Para criar uma rede de mensagens distribuída, configure manualmente os canais para conectar filas hospedadas em diferentes gerenciadores de filas ou crie um cluster de gerenciador de filas. Armazenamento em cluster permite que os gerenciadores de filas no cluster se comuniquem entre si sem necessidade de configurar definições extras de canal ou de fila remota, simplificando a configuração e o gerenciamento.

Para escolher uma topologia apropriada para sua rede distribuída de publicação/assinatura, é preciso considerar as seguintes questões gerais:

- Quanto controle manual você precisa sobre as conexões em sua rede?
- Qual será o tamanho da sua rede?
- Quão dinâmica ela será?
- Quais são seus requisitos de disponibilidade e escalabilidade?

Procedimento

- Considere quanto controle manual você precisa sobre as conexões em sua rede.

Se precisar apenas de algumas conexões, ou se as conexões individuais precisarem ser muito precisamente definidas, você provavelmente deve criar a rede manualmente.

Se precisar vários gerenciadores de filas relacionados logicamente e que precisem compartilhar dados e aplicativos, considere o agrupá-los em um cluster de gerenciador de filas.

- Estime o tamanho que sua rede precisa ter.
 - a) Estime de quantos gerenciadores de filas você precisa. Tenha em mente que as filas podem ser hospedadas em mais de um gerenciador de filas.
 - b) Se você estiver considerando o uso de um cluster, inclua dois gerenciadores de filas extras para atuarem como repositórios completos.

Para redes maiores, a configuração manual e a manutenção de conexões podem ser muito demoradas, e você deve considerar o uso de um cluster.

- Considere o quão dinâmica a atividade de rede será.

Planeje hospedar filas ocupadas em gerenciadores de fila de alto desempenho.

Se esperar que filas sejam criadas e excluídas com frequência, considere usar um cluster.

- Considere os requisitos de disponibilidade e escalabilidade.
 - a) Decida se precisa garantir a alta disponibilidade de gerenciadores de filas. Se precisar, estime a quantos gerenciadores de filas esse requisito se aplica.
 - b) Considere se alguns de seus gerenciadores de filas são menos capazes do que outros.
 - c) Considere se os links de comunicação para alguns de seus gerenciadores de filas são mais frágeis do que para outros.
 - d) Considere hospedar filas em vários gerenciadores de fila.

Redes e clusters configurados manualmente podem ser configurados para serem altamente disponíveis e escaláveis. Se você usar um cluster, precisa definir dois gerenciadores de filas adicionais como repositórios completos. Ter dois repositórios completos assegura que o cluster continuará a operar se um dos repositórios completos ficar indisponível. Certifique-se de que os gerenciadores de filas de repositório completo sejam robustos, de bom desempenho e com boa conectividade de rede. Não planeje usar os gerenciadores de filas de repositório completo para nenhum outro trabalho.

- Com base nesses cálculos, use os links fornecidos para ajudá-lo a decidir se deve configurar manualmente as conexões entre os gerenciadores de filas ou usar um cluster.

Como proceder a seguir

Você está pronto para configurar seu provedor de rede distribuída.

Tarefas relacionadas

[Configurando enfileiramento distribuído](#)

[Configurando um cluster do gerenciador de filas](#)

Projetando redes do gerenciador de filas distribuído

O IBM MQ envia e recebe dados entre aplicativos e através de redes usando Gerenciadores de Filas e Canais. O planejamento de rede envolve definir os requisitos para criar uma estrutura para conectar esses sistemas em uma rede.

Os canais podem ser criados entre o seu sistema e qualquer outro sistema com o qual você precisa ter comunicações. Os canais multi-hop podem ser criados para se conectarem aos sistemas em que você não tenha conexões diretas. As conexões do canal de mensagens descritas nos cenários são mostrados como um diagrama de rede em [Figura 1 na página 22](#).

Se for necessário criar canais entre sistemas em diferentes redes físicas ou canais que se comunicam por meio de um firewall, usar o IBM MQ Internet Pass-Thru pode simplificar a configuração. Para obter mais informações, consulte [IBM MQ Internet Pass-Thru](#).

Nomes da Fila de Transmissão e Canal

As filas de transmissão podem receber qualquer nome. Mas para evitar confusão, você pode fornecer a elas os mesmos nomes que os nomes do gerenciador de filas de destino ou os nomes do alias

do gerenciador de filas, conforme apropriado. Isso associa a fila de transmissão à rota que é usada, fornecendo uma visão geral clara das rotas paralelas criadas por meio dos gerenciadores de fila intermediários (com multi-hop).

Isso não é tão bem-definido para nomes de canal. Os nomes de canal em [Figura 1 na página 22](#) para QM2, por exemplo, devem ser diferentes para canais recebidos e de saída. Todos os nomes de canal ainda podem conter seus nomes da fila de transmissão, mas eles devem ser qualificados para que se tornem exclusivos.

Por exemplo, no QM2, existe um canal QM3 proveniente de QM1, e um canal QM3 indo para QM3. Para tornar os nomes exclusivos, o primeiro pode se chamar QM3_from_QM1 e o segundo pode se chamar QM3_from_QM2. Desta maneira, os nomes de canal mostram o nome da fila de transmissão na primeira parte do nome. O nome do gerenciador de filas adjacente e a direção são mostrados na segunda parte do nome.

Uma tabela de nomes de canal sugeridos para [Figura 1 na página 22](#) é fornecida em [Tabela 1 na página 22](#).

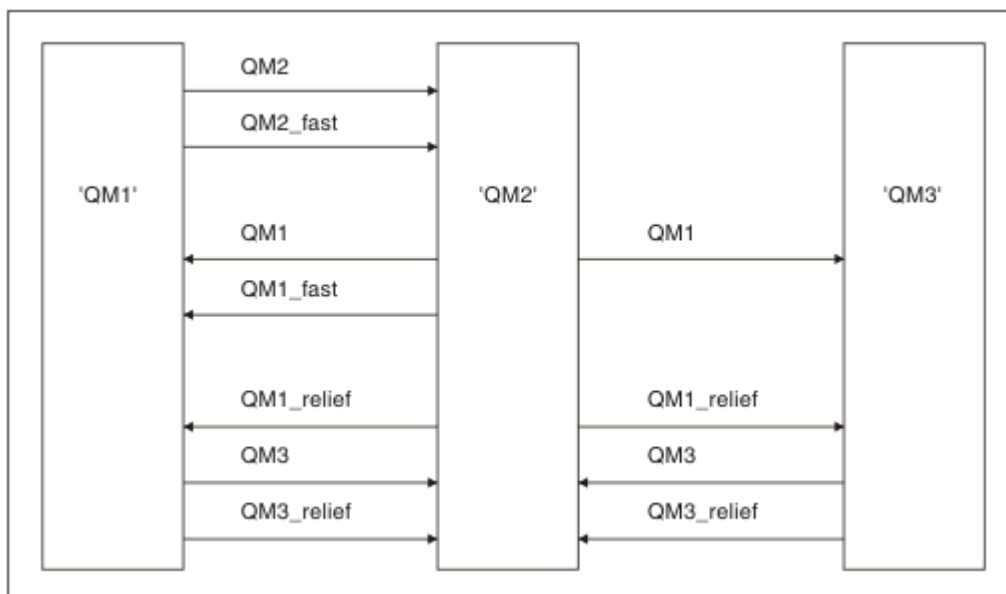


Figura 1. Diagrama de Rede que Mostra Todos os Canais


Tabela 1. Exemplo de Nomes de Canal

Nome da Rota	Gerenciadores de fila que hospedam o canal	Nome da fila de transmissão	Nome do canal sugerido
QM1	QM1 & QM2	QM1 (em QM2)	QM1.from.QM2
QM1	QM2 & QM3	QM1 (em QM3)	QM1.from.QM3
QM1_fast	QM1 & QM2	QM1_fast (em QM2)	QM1_fast.from.QM2
QM1_relief	QM1 & QM2	QM1_relief (em QM2)	QM1_relief.from.QM2
QM1_relief	QM2 & QM3	QM1_relief (em QM3)	QM1_relief.from.QM3
QM2	QM1 & QM2	QM2 (em QM1)	QM2.from.QM1
QM2_fast	QM1 & QM2	QM2_fast (em QM1)	QM2_fast.from.QM1
QM3	QM1 & QM2	QM3 (em QM1)	QM3.from.QM1

Tabela 1. Exemplo de Nomes de Canal (continuação)

Nome da Rota	Gerenciadores de fila que hospedam o canal	Nome da fila de transmissão	Nome do canal sugerido
QM3	QM2 & QM3	QM3 (em QM2)	QM3.from.QM2
QM3_relief	QM1 & QM2	QM3_relief (em QM1)	QM3_relief.from.QM1
QM3_relief	QM2 & QM3	QM3_relief (em QM2)	QM3_relief.from.QM2

Nota:

1.  No IBM MQ for z/OS, os nomes de gerenciadores de filas são limitados a quatro caracteres.
2. Nomeie todos os canais em sua rede exclusivamente. Conforme mostrado em Tabela 1 na página 22, incluir os nomes do gerenciador de filas de origem e destino no nome do canal é uma boa maneira de se fazer isso.

Planejador de Rede

Criar uma rede assume que existe outra função de nível superior de *planejador de rede* cujos planos são implementados por outros membros da equipe.

Para os aplicativos amplamente usados, é mais econômico pensar em termos de sites de acesso local para a concentração de tráfego de mensagem, usando links de banda ampla entre os sites de acesso local, conforme mostrado em Figura 2 na página 24.

Neste exemplo existem dois sistemas principais e inúmeros sistemas de satélite. A configuração real dependeria de considerações de negócios. Existem dois gerenciadores de fila de concentrador localizados nos centros convenientes. Cada concentrador QM tem canais de mensagem para os gerenciadores de fila local:

- O concentrador QM 1 tem canais de mensagens para cada um dos três gerenciadores de fila local, QM1, QM2 e QM3. Os aplicativos que usam esses gerenciadores de filas podem se comunicar entre si por meio dos concentradores QM.
- O concentrador QM 2 tem canais de mensagens para cada um dos três gerenciadores de fila local, QM4, QM5 e QM6. Os aplicativos que usam esses gerenciadores de filas podem se comunicar entre si por meio dos concentradores QM.
- Os concentradores QM possuem canais de mensagens entre si permitindo assim que qualquer aplicativo em um gerenciador de filas trocasse mensagens com qualquer outro aplicativo em outro gerenciador de filas.

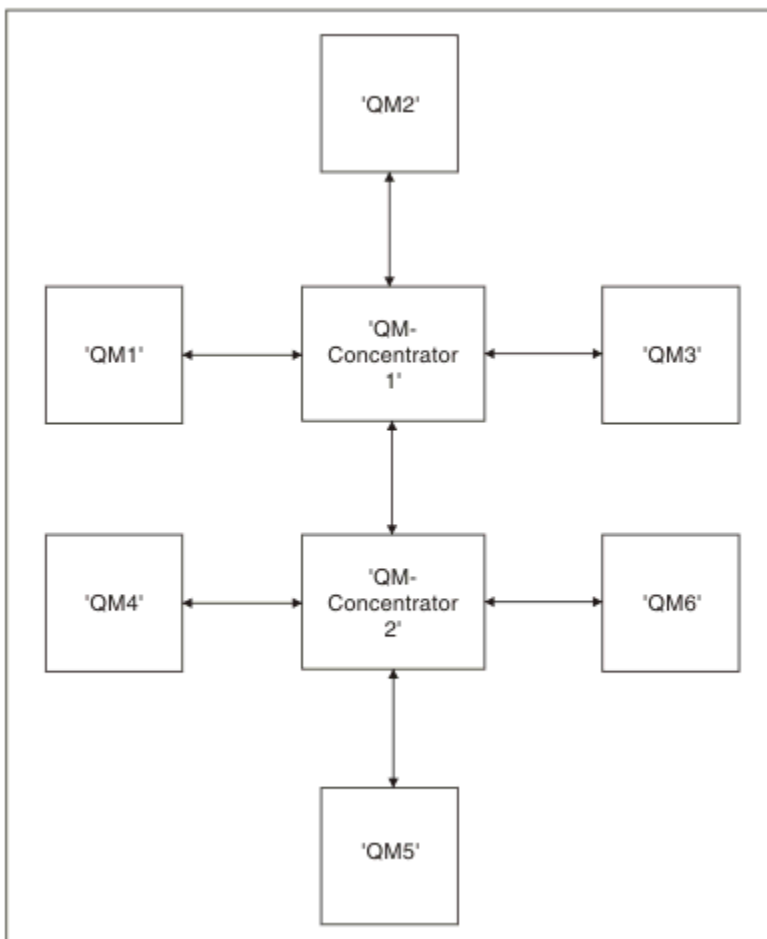


Figura 2. Diagrama de Rede que Mostra Concentradores QM

Projetando Clusters

Os clusters fornecem um mecanismo para interconectar gerenciadores de filas de uma maneira que simplifique a configuração inicial e o gerenciamento em andamento. Os clusters devem ser cuidadosamente projetados, para assegurar que funcionem corretamente e que obtenham os níveis necessários de disponibilidade e responsividade.

Antes de começar


Para obter uma introdução para armazenamento em cluster dos conceitos, consulte os seguintes tópicos:

- [Enfileiramento e clusters distribuídos](#)
- [“Comparação de Armazenamento em Cluster e Enfileiramento Distribuído” na página 30](#)
- [Componentes de um cluster](#)

Ao projetar seu cluster do gerenciador de filas você precisa tomar algumas decisões. Primeiro precisa decidir quais gerenciadores de filas no cluster devem reter os repositórios completos de informações de cluster. Qualquer gerenciador de filas que você criar pode trabalhar em um cluster. Você pode escolher qualquer número de gerenciadores de filas para este propósito mas o número ideal é dois. Para obter informações sobre a seleção de gerenciadores de filas para conter os repositórios completos, consulte [“Como escolher gerenciadores de filas do cluster para conter repositórios completos” na página 33.](#)

Consulte os seguintes tópicos para obter informações adicionais sobre como projetar o seu cluster:

- [“Clusters de exemplo” na página 39](#)
- [“Organizando um Cluster” na página 34](#)

- [“Convenções de nomenclatura do cluster”](#) na página 35
-  [“Queue sharing groups and clusters”](#) na página 36
- [“Sobrepondo Clusters”](#) na página 36


Como proceder a seguir

Consulte os seguintes tópicos para obter informações adicionais sobre a configuração e o trabalho com clusters:

- [Estabelecendo comunicação em um cluster](#)
- [Configurando um cluster do gerenciador de filas](#)
- [Roteando mensagens de clusters e para eles](#)
- [Usando clusters para gerenciamento de carga de trabalho](#)

Para obter mais informações para ajudá-lo a configurar seu cluster, consulte [“Dicas de armazenamento em cluster”](#) na página 37.

Planejando como usar diversas filas de transmissão do cluster

É possível definir explicitamente filas de transmissão ou deixar que o sistema gere as filas de transmissão para você. Se você mesmo definir as filas de transmissão, terá mais controle sobre as definições de fila.  No z/OS, você também tem mais controle sobre o conjunto de páginas no qual as mensagens são retidas.

Definindo as filas de transmissão


Há dois métodos para definir filas de transmissão:

- Automaticamente, usando o atributo do gerenciador de filas, DEFCLXQ, conforme a seguir:

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

DEFCLXQ(SCTQ) indica que a fila de transmissão padrão para todos os canais do emissor de clusters é SYSTEM.CLUSTER.TRANSMIT.QUEUE. Esse é o valor-padrão.

DEFCLXQ(CHANNEL) indica que, por padrão, cada canal do emissor de clusters usa uma fila de transmissão separada denominada SYSTEM.CLUSTER.TRANSMIT.*channel name*. Cada fila de transmissão é definida automaticamente pelo gerenciador de filas. Consulte o [“Filas de transmissão do cluster automaticamente definidas”](#) na página 26 para obter mais informações.

- Manualmente, definindo uma fila de transmissão com um valor especificado para o atributo CLCHNAME. O atributo CLCHNAME indica quais canais do emissor de clusters devem usar a fila de transmissão.  Se estiver definindo manualmente uma fila de transmissão no z/OS, consulte [“Planejamento para filas de transmissão do cluster definidas manualmente”](#) na página 28 para obter mais informações.

De qual tipo de segurança eu preciso?

Para iniciar um comutador, automática ou manualmente, é necessário autoridade para iniciar um canal.

Para definir a fila usada como uma fila de transmissão, é necessária a autoridade padrão do IBM MQ para definir a fila.

Quando é o momento adequado para implementar a mudança?

Ao mudar a fila de transmissão usada pelos canais do emissor de cluster, é necessário alocar um tempo no qual fazer a atualização, considerando os pontos a seguir:

- O tempo necessário para um canal alternar a fila de transmissão depende do número total de mensagens na fila de transmissão antiga, de quantas mensagens precisam ser movidas e do tamanho das mensagens.
- Os aplicativos podem continuar colocando mensagens na fila de transmissão enquanto a mudança estiver ocorrendo. Isso pode gerar um aumento no tempo de transição.
- É possível alterar o parâmetro CLCHNAME de qualquer fila de transmissão ou DEFCLXQ a qualquer momento, preferencialmente quando a carga de trabalho estiver baixa.

Observe que nada acontece imediatamente.

- As mudanças ocorrem somente quando um canal é iniciado ou reiniciado. Quando um canal é iniciado, ele verifica a configuração atual e alterna para uma nova fila de transmissão, se necessário.
- Existem várias mudanças que podem alterar a associação de um canal do emissor de clusters a uma fila de transmissão:
 - Alterando o valor do atributo CLCHNAME de uma fila de transmissão, tornando CLCHNAME menos específico ou em branco.
 - Alterando o valor do atributo CLCHNAME de uma fila de transmissão, tornando CLCHNAME mais específico.
 - Excluir uma fila com CLCHNAME especificado.
 - Alterando o atributo DEFCLXQ do gerenciador de filas.


Quanto tempo durará a comutação?

Durante o período de transição, todas as mensagens para o canal são movidas de uma fila de transmissão para outra. O tempo necessário para um canal alternar a fila de transmissão depende do número total de mensagens na fila de transmissão antiga e de quantas mensagens precisam ser movidas.

Para filas contendo alguns milhares de mensagens, deve levar menos de um segundo para mover as mensagens. O tempo real depende do número e tamanho das mensagens. Seu gerenciador de filas deve ser capaz de mover mensagens de muitos megabytes a cada segundo.

Os aplicativos podem continuar colocando mensagens na fila de transmissão enquanto a mudança estiver ocorrendo. Isso pode gerar um aumento no tempo de transição.

Cada canal do emissor de clusters afetado deve ser reiniciado para que a mudança entre em vigor. Portanto, é melhor mudar a configuração da fila de transmissão quando o gerenciador de filas não está ocupado e algumas mensagens são armazenadas nas filas de transmissão do cluster.

O Comando `runswchl`  ou o Comando `SWITCH CHANNEL (*) STATUS` em `CSQUTIL` em z/OS pode ser usado para consultar o status dos canais do emissor de clusters e quais mudanças pendentes estão pendentes em sua configuração da fila de transmissões

Como implementar a mudança

Consulte [Implementando o sistema usando diversas filas de transmissão do cluster para obter detalhes sobre como fazer a mudança para diversas filas de transmissão do cluster, automaticamente ou manualmente.](#)

Desfazendo a mudança




Consulte [Desfazendo uma mudança em uma fila de transmissão em z/OS](#) para obter detalhes sobre como recuperar as mudanças se você encontrar problemas.

Filas de transmissão do cluster automaticamente definidas


É possível ter o sistema gerando as filas de transmissão para você.

Antes de começar

 Para configurar as filas de transmissão do cluster manualmente no z/OS, consulte [“Planejamento para filas de transmissão do cluster definidas manualmente”](#) na página 28

Sobre esta tarefa

Se um canal não tem uma fila de transmissão do cluster definida manualmente associada a ele, e você especificar DEFCLXQ(CHANNEL), quando o canal inicia o gerenciador de filas define automaticamente uma fila dinâmica-permanente para o canal do emissor de clusters. A fila modelo SYSTEM.CLUSTER.TRANSMIT.MODEL.queue é usada para definir automaticamente a FILA de transmissão de cluster dinâmico permanente com o nome SYSTEM.cluster.TRANSMIT.Channe1Name.

Importante:  No IBM MQ 8.0, o gerenciador de filas não tem o SYSTEM . CLUSTER . TRANSMIT . MODEL . QUEUE Não é possível migrar diretamente do IBM MQ 8.0 para esta versão Para obter informações sobre como incluir o SYSTEM . CLUSTER . TRANSMIT . MODEL . QUEUE em um gerenciador de filas que é migrado do IBM MQ 8.0, consulte este tópico na documentação para a versão provisória que você usou para migrar o gerenciador de filas

Procedimento

1. Use o atributo do gerenciador de filas DEFCLXQ.

Para obter mais informações sobre esse atributo, consulte [ALTER QMGR](#).

Existem duas opções:

SCTQ

Essa opção é o padrão, e significa que você usa o único SYSTEM.CLUSTER.TRANSMIT.QUEUE.

CHANNEL

Significa que você usa diversas filas de transmissão do cluster.

2. Para alternar para a nova associação:

- Pare e reinicie o canal.
- O canal usa a nova definição de fila de transmissão.
- As mensagens são transferidas por um processo do comutador transicional da fila antiga para a nova fila de transmissão.

Observe que qualquer mensagem do aplicativo é colocada na definição antiga.

Quando o número de mensagens na fila antiga atinge zero, novas mensagens são colocadas diretamente na nova fila de transmissão.

3. Para monitorar quando o processo de comutação é concluído:

- a) Um comutador de fila de transmissão que é iniciado por um canal é executado em segundo plano, e o administrador pode monitorar o log da tarefa do gerenciador para determinar quando ela foi concluída.
- b) Monitore as mensagens no log da tarefa para mostrar o progresso do comutador.
- c) Para ter certeza de que somente os canais que você deseja estão usando esta fila de transmissão, emita o comando DIS CLUSQMGR (*) em que, por exemplo, a propriedade de fila de transmissão que define a fila de transmissão é APPQMGR . CLUSTER1 . XMITQ.

- d) 

Use o comando SWITCH CHANNEL (*) STATUS em CSQUTIL.

Esta opção informa quais mudanças pendentes estão pendente e quantas mensagens precisam ser movidas entre as filas de transmissão.

Resultados

É necessário configurar a fila de transmissão do cluster, ou filas.

Tarefas relacionadas

[“Planejamento para filas de transmissão do cluster definidas manualmente”](#) na página 28

No IBM MQ for z/OS, se você mesmo definir as filas de transmissão, terá mais controle sobre as definições e o conjunto de páginas no qual as mensagens são mantidas.

Referências relacionadas

[ALTER QMGR](#)

[EXIBIR CLUSQMGR](#)

 *Planejamento para filas de transmissão do cluster definidas manualmente*

No IBM MQ for z/OS, se você mesmo definir as filas de transmissão, terá mais controle sobre as definições e o conjunto de páginas no qual as mensagens são mantidas.

Antes de começar

Para configurar filas de transmissão do cluster automaticamente, consulte [“Filas de transmissão do cluster automaticamente definidas”](#) na página 26.

Sobre esta tarefa

Seu administrador define manualmente uma fila de transmissão e usa o atributo de fila CLCHNAME para definir qual canal do emissor de clusters, ou canais, usará essa fila como sua fila de transmissão.

Observe que CLCHNAME pode incluir um caractere curinga no início ou término, para permitir que uma única fila seja usada para múltiplos canais.

Procedimento

1. Por exemplo, insira o seguinte:

```
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)
```

Sugestão: É necessário planejar qual conjunto de páginas (e buffer pool) usar para suas filas de transmissão. É possível ter diferentes conjuntos de páginas para diferentes filas e fornecer isolamento entre elas, portanto, um preenchimento de conjunto de páginas não impacta filas de transmissão em outros conjuntos de páginas..

Consulte [Trabalhando com filas de transmissão do cluster e canais do emissor de clusters](#) para obter informações sobre como cada canal seleciona a fila apropriada.

Quando o canal é iniciado, ele alterna sua associação com a nova fila de transmissão. Para assegurar que nenhuma mensagem seja perdida, o gerenciador de filas transfere automaticamente as mensagens da fila de transmissão do cluster antigo para a nova fila de transmissão em ordem.

2. Use a função CSQUTIL SWITCH para mudar para a nova associação.

Consulte [Alternar a fila de transmissão associada aos canais do emissor de clusters \(SWITCH\)](#) para obter informações adicionais.

- a) PARE o canal, ou canais, cuja fila de transmissão deve ser mudada, de forma que estejam no status STOPPED.

Por exemplo:

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

- b) Mude o atributo CLCHNAME (XXXX) na fila de transmissão.
- c) Use a função SWITCH para alternar as mensagens ou monitorar o que está acontecendo.
Use the command

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

para mover as mensagens sem iniciar o canal.

- d) Inicie o canal, ou canais, e verifique se o canal está usando as filas corretas.
Por exemplo:

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

Sugestão: O processo a seguir usa a função CSQUTIL SWITCH. Para obter mais informações, consulte [Alternar a fila de transmissão associada a canais do emissor de clusters \(SWITCH\)](#).

Não é necessário usar essa função, mas usá-la oferece mais opções:

- Usar SWITCH CHANNEL (*) STATUS fornece uma maneira fácil de identificar o status de comutação de canais do emissor de clusters. Ele permite que seu administrador veja quais canais estão atualmente sendo alternados, e aqueles canais com um comutação pendente entram em vigor na próxima inicialização.

Sem este recurso, seu administrador precisa usar vários comandos DISPLAY e, em seguida, processar a saída resultante para verificar essas informações. Seu administrador pode também confirmar que uma mudança na configuração tem o resultado necessário.

- Se CSQUTIL for usado para iniciar o comutador, o CSQUTIL continua a monitorar o progresso desta operação, e finaliza apenas quando o comutador é concluído.

Isso pode tornar muito mais fácil a execução das operações em lote. Além disso, se CSQUTIL for executado para alternar entre vários canais, CSQUTIL executa essas ações sequencialmente; isto pode ter menos impacto para sua empresa do que vários comutadores executando em paralelo.

Resultados

Você configurou sua fila de transmissão do cluster, ou filas, no z/OS.

Controle de acesso e diversas filas de transmissão do cluster

Escolha entre três modos de verificação quando um aplicativo coloca mensagens nas filas do cluster remoto. Os modos estão sendo verificados remotamente em relação a fila de clusters, verificando localmente em relação a SYSTEM . CLUSTER . TRANSMIT . QUEUE ou verificando em relação a perfis locais para a fila de clusters ou gerenciador de filas de clusters.

O IBM MQ fornece a opção de verificar localmente, localmente e remotamente, se um usuário tem permissão para colocar uma mensagem em uma fila remota. Um aplicativo IBM MQ típico usa apenas a verificação local e conta com o gerenciador de filas remoto, confiando as verificações de acesso feitas no gerenciador de filas locais. Se a verificação remota não for usada, a mensagem é colocada na fila de destino com a autoridade do processo de canal de mensagens remoto. Para usar a verificação remoto, deve-se configurar a autoridade de colocação do canal de recebimento para segurança de contexto.


As verificações do local são feitas com relação à fila em que o aplicativo é aberto. No enfileiramento distribuído, o aplicativo geralmente abre uma definição de fila remota, e são feitas verificações de acesso com relação à definição de fila remota. Se a mensagem for colocada com um cabeçalho de roteamento completo, as verificações são feitas com relação à fila de transmissão. Se um aplicativo abrir uma fila de clusters que não está no gerenciador de filas locais, não existirá um objeto local para verificar. As verificações de controle de acesso são feitas em relação a fila de

transmissão do cluster, SYSTEM.CLUSTER.TRANSMIT.QUEUE. Mesmo com várias filas de transmissão do cluster, as verificações de controle de acesso local para filas do cluster remoto são feitas em SYSTEM.CLUSTER.TRANSMIT.QUEUE.

A opção de verificação local ou remota é uma escolha entre dois extremos. Verificar remotamente tem baixa granularidade. Cada usuário deve ter um perfil do controle de acesso em cada gerenciador de filas no cluster a ser colocado em qualquer fila de clusters. Verificar localmente tem alta granularidade. Cada usuário precisa de apenas um perfil de controle de acesso para a fila de transmissão do cluster no gerenciador ao qual estão conectados. Com esse perfil, eles podem colocar uma mensagem em qualquer fila de clusters, em qualquer gerenciador de filas, em qualquer cluster.

Os administradores têm uma outra maneira de configurar o controle de acesso para filas de clusters. É possível criar um perfil de segurança para uma fila de clusters em qualquer gerenciador de filas no cluster, usando o comando **setmqaut**. O perfil entra em vigor se você abrir uma fila de clusters remotos localmente, especificando apenas o nome da fila. Também é possível configurar um perfil para um gerenciador de filas remotas. Se você fizer isso, o gerenciador de filas pode verificar o perfil de um usuário que abre uma fila de clusters, fornecendo um nome completo.

Os novos perfis funcionam somente se você alterar a sub-rotina do gerenciador de filas, **ClusterQueueAccessControl** para RQMNome. O padrão é Xm1tq. Deve-se criar perfis para todos os aplicativos de filas de clusters existentes que usam filas de clusters. Se a sub-rotina for mudada para RQMName sem criar perfis, os aplicativos provavelmente falharão.

Sugestão: A verificação de acesso da fila de clusters não se aplica ao enfileiramento remoto. As verificações de acesso ainda são feitas com relação às definições locais. As mudanças significam que é possível seguir a mesma abordagem para configurar verificação de acesso em filas de clusters e tópicos de cluster.  As mudanças também alinham a abordagem de verificação de acesso para as filas de clusters de forma mais próxima com o z/OS. Os comandos para configurar a verificação de acesso em z/OS são diferentes, mas ambos verificam o acesso em relação a um perfil em vez de ao próprio objeto.

Conceitos relacionados

[“Armazenamento em cluster: isolamento de aplicativos usando diversas filas de transmissão do cluster” na página 49](#)

É possível isolar os fluxos de mensagens entre gerenciadores de filas em um cluster. É possível colocar mensagens que estão sendo transportadas por diferentes canais do emissor de clusters em diferentes filas de transmissão do cluster. É possível usar a abordagem em um único cluster ou com clusters sobrepostos. O tópico fornece exemplos e algumas melhores práticas para orientar a escolha de uma abordagem a usar.

Tarefas relacionadas

[Configuração do ClusterQueueAccessControl](#)

Comparação de Armazenamento em Cluster e Enfileiramento Distribuído

Compare os componentes que precisam ser definidos para conectar gerenciadores de filas usando enfileiramento e armazenamento em cluster distribuído.

Se você não usar clusters, seus gerenciadores de filas serão independentes e se comunicarão usando enfileiramento distribuído. Se um gerenciador de filas precisa enviar mensagens para um outro, você deve definir:

- Uma fila de transmissão
- Um canal para o gerenciador de filas remotas

[Figura 3 na página 31](#) mostra os componentes necessários para enfileiramento distribuído.

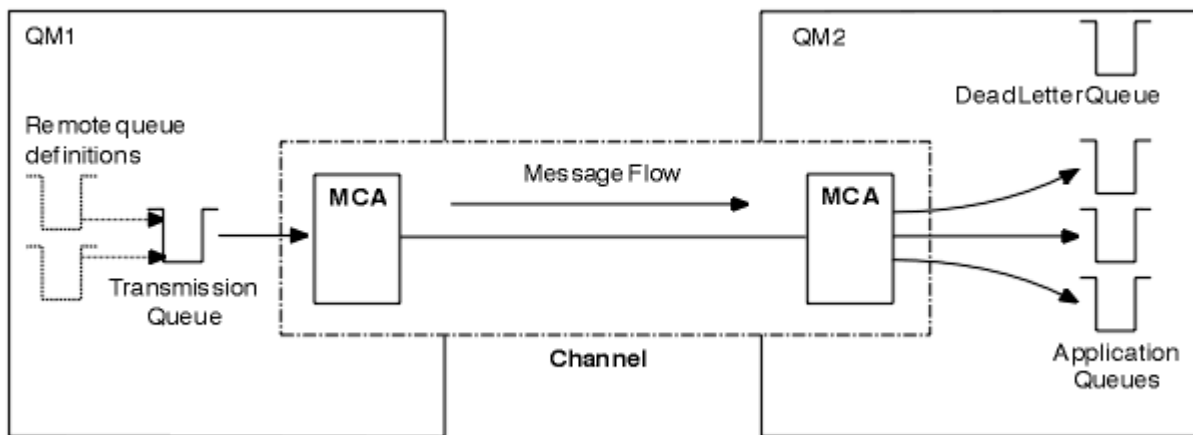


Figura 3. Enfileiramento distribuído

Se você agrupar gerenciadores de filas em um cluster, as filas em qualquer gerenciador de filas estarão disponíveis para qualquer outro gerenciador de filas no cluster. Qualquer gerenciador de filas pode enviar uma mensagem para qualquer outro gerenciador de filas no mesmo cluster sem definições explícitas. Você não fornece definições de canal, definições de fila remota ou filas de transmissão para cada destino. Cada gerenciador de filas em um cluster possui uma única fila de transmissão a partir da qual ele pode transmitir mensagens a qualquer outro gerenciador de filas no cluster. Cada gerenciador de filas em um cluster precisa definir apenas:

- Um canal do receptor de clusters no qual receber mensagens
- Um canal do emissor de clusters com o qual ele se apresenta e aprende sobre o cluster

Definições para Configurar um Cluster Versus um Enfileiramento Distribuído

Consulte Figura 4 na página 31, que mostra quatro gerenciadores de filas, cada um com duas filas. Considere quantas definições são necessárias para conectar estes gerenciadores de filas usando o enfileiramento distribuído. Compare quantas definições são necessárias para configurar a mesma rede como um cluster.

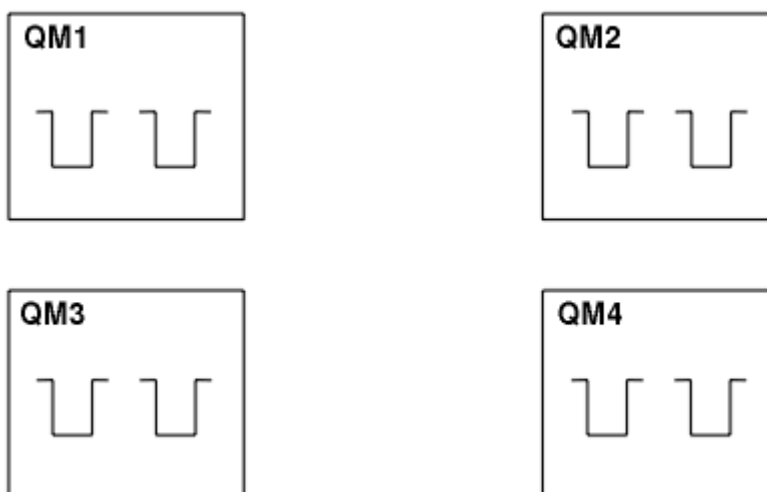


Figura 4. Uma Rede de Quatro Gerenciadores de Filas

Definições para Configurar uma Rede Usando Enfileiramento Distribuído

Para configurar a rede mostrada em Figura 3 na página 31 usando enfileiramento distribuído, você pode ter as seguintes definições:

Descrição	Número por Gerenciador de Filas	Número Total
Uma definição do canal emissor para um canal no qual enviar mensagens para gerenciador de filas sim outro não	3	12
Uma definição de canal receptor para um canal no qual receber mensagens de um gerenciador de filas sim outro não	3	12
Uma definição de fila de transmissão para uma fila de transmissão para um gerenciador de filas sim outro não	3	12
Uma definição de fila local para cada fila local	2	8
Uma definição de fila remota para cada fila remota na qual este gerenciador de filas deseja colocar mensagens	6	24

Você pode reduzir este número de definições usando definições de canal receptor genéricas. O número máximo de definições poderia ser tantos quanto 17 em cada gerenciador de filas, que é um total de 68 para esta rede.

Definições para Configurar uma Rede Usando Clusters

Para configurar a rede mostrada em [Figura 3 na página 31](#) usando clusters, são necessárias as seguintes definições:

Descrição	Número por Gerenciador de Filas	Número Total
Uma definição de canal do emissor de clusters para um canal no qual enviar mensagens para um gerenciador de fila de repositório	1	4
Uma definição de canal do receptor de clusters para um canal no qual receber mensagens de outros gerenciadores de filas no cluster	1	4
Uma definição de fila local para cada fila local	2	8

Para configurar este cluster de gerenciadores de filas (com dois repositórios completos), são necessárias quatro definições em cada gerenciador de filas, um total de dezesseis definições ao todo. Você também precisa alterar as definições do gerenciador de filas para dois dos gerenciadores de filas, para torná-los gerenciadores de filas do repositório completo para o cluster.

Somente uma definição de canal CLUSSDR e uma CLUSRCVR são necessárias. Quando o cluster é definido, é possível incluir ou remover gerenciadores de filas (diferentes dos gerenciadores de filas do repositório) sem qualquer interrupção nos outros gerenciadores de filas.

Usar um cluster reduz o número de definições necessárias para configurar uma rede contendo muitos gerenciadores de filas.

Com menos definições para fazer, há menos risco de erro:

- Os nomes do objeto sempre correspondem, por exemplo, o nome de canal em um par de emissor-receptor.
- O nome da fila de transmissão especificado em uma definição de canal sempre corresponde à definição de fila de transmissão correta ou ao nome da fila de transmissão especificado em uma definição de fila remota.
- Uma definição de QREMOTE sempre aponta para a fila correta no gerenciador de filas remotas.

Quando um cluster é configurado, é possível mover as filas de clusters de um gerenciador de filas para um outro dentro do cluster sem precisar fazer qualquer trabalho de gerenciamento de sistemas em qualquer outro gerenciador de filas. Não há chance de esquecer de excluir ou modificar as definições de canal, fila remota ou fila de transmissão. É possível incluir novos gerenciadores de filas em um cluster sem qualquer interrupção na rede existente.

Como escolher gerenciadores de filas do cluster para conter repositórios completos

Em cada cluster, deve-se escolher pelo menos um, e, de preferência, dois gerenciadores de filas para conter repositórios completos. Dois repositórios completos são suficientes para todos, senão na maioria das circunstâncias excepcionais. Se possível, escolha gerenciadores de filas que são hospedados em plataformas robustas e permanentemente conectadas, que não tenham interrupções coincidentes, e que estejam em uma posição central geograficamente. Além disso, considere dedicar os sistemas como hosts de repositório completo, e não usar esses sistemas para quaisquer outras tarefas.

Repositórios completos são gerenciadores de filas que mantêm uma imagem completa do estado do cluster. Para compartilhar essas informações, cada repositório completo é conectado por canais CLUSSDR (e suas definições correspondentes CLUSRCVR) a qualquer outro repositório completo no cluster. Deve-se definir manualmente esses canais.



Figura 5. Dois repositórios completos conectados.

Todos os gerenciadores de filas no cluster mantêm uma ideia do que ele conhece atualmente sobre o estado do cluster em um *repositório parcial*. Esses gerenciadores de filas publicam informações sobre si próprios e solicitam informações sobre outros gerenciadores de filas, usando quaisquer dois repositórios completos disponíveis. Se um repositório completo escolhido não estiver disponível, outro será usado. Quando o repositório completo escolhido se tornar disponível novamente, ele coletará as informações novas e alteradas mais recentes de outros para que eles continuem na etapa. Se todos os repositórios completos ficarem fora de serviço, os outros gerenciadores de filas usam as informações que possuem em seus repositórios parciais. No entanto, eles estão limitados a usar as informações que possuem; informações novas e solicitações de atualizações não podem ser processadas. Quando os repositórios completos se reconectam à rede, as mensagens são trocadas para atualizar todos os repositórios (completos e parciais).

Ao planejar a alocação de repositórios completos, inclua as seguintes considerações:

- Os gerenciadores de filas escolhidos para conter repositórios completos precisam ser confiáveis e gerenciados. Escolha gerenciadores de filas que são hospedados em uma plataforma robusta e permanentemente conectada.
- Considere as indisponibilidades planejadas para os sistemas que hospedam seus repositórios completos e assegure que eles não tenham interrupções coincidentes.
- Considere o desempenho da rede: escolha os gerenciadores de filas que estão em uma posição central geograficamente, ou que compartilham o mesmo sistema que outros gerenciadores de filas no cluster.
- Considere se um gerenciador de filas é um membro de mais de um cluster. Pode ser administrativamente conveniente usar o mesmo gerenciador de filas para hospedar os repositórios completos para vários clusters, desde que este benefício seja equilibrado com relação a quão ocupado você espera que o gerenciador de filas esteja.
- Considere dedicar alguns sistemas para conter apenas repositórios completos, e não use esses sistemas para qualquer outra tarefa. Isso assegura que esses sistemas requeiram apenas manutenção para a configuração do gerenciador de filas, e não são removidos do serviço para a manutenção de outros aplicativos de negócios. Ele também garante que a tarefa de manter o repositório não concorra com aplicativos de recursos do sistema. Isso pode ser particularmente benéfico em clusters grandes

(digamos, clusters de mais de um milhão de gerenciadores de filas), onde repositórios completos têm uma carga de trabalho muito maior em manter o estado do cluster.

Ter mais de dois repositórios completos é possível, mas raramente recomendado. Embora as definições de objeto (isto é, filas, tópicos e canais) fluam em todos os repositórios completos, as solicitações fluem apenas de um repositório parcial para um máximo de dois repositórios completos. Isso significa que, quando mais de dois repositórios completos são definidos, e quaisquer dois repositórios completos se tornarem indisponíveis, alguns repositórios parciais podem não receber as atualizações que eles esperam. Consulte [Clusters do MQ: por que apenas dois repositórios completos?](#)

Uma situação na qual você pode achar útil para definir mais de dois repositórios completos é ao migrar repositórios completos para um novo hardware ou novos gerenciadores de filas existentes. Neste caso, deve-se apresentar os repositórios completos de substituição e confirmar que eles foram totalmente preenchidos, antes de remover repositórios completos anteriores. Sempre que você incluir um repositório completo, lembre-se de que deve conectá-lo diretamente a qualquer outro repositório completo com canais CLUSSDR.

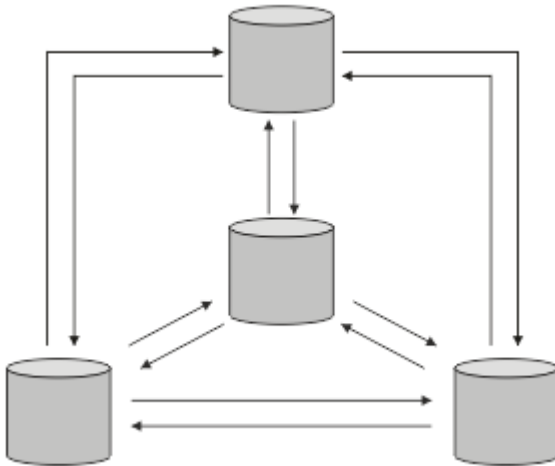


Figura 6. Mais de dois repositórios completos conectados

Informações relacionadas

[Clusters MQ: por que apenas dois repositórios completos?](#)

[Quão grande pode ser um cluster do MQ?](#)

Organizando um Cluster

Selecione quais gerenciadores de filas vincular a qual repositório completo. Considere o efeito de desempenho, a versão do gerenciador de filas e se vários canais CLUSSDR são desejáveis.

Tendo selecionado os gerenciadores de filas para conter repositórios completos, é necessário decidir quais gerenciadores de filas vincular a qual repositório completo. A definição de canal CLUSSDR vincula um gerenciador de filas a um repositório completo a partir do qual ele descobre sobre os outros repositórios completos no cluster. Daí em diante, o gerenciador de filas envia mensagens para qualquer um dos dois repositórios completos. Ele sempre tenta usar primeiro aquele para o qual ele possui uma definição de canal CLUSSDR. É possível escolher vincular um gerenciador de filas a um repositório completo. Na escolha, considere a topologia de sua configuração e o local físico ou geográfico dos gerenciadores de filas.

Como todas as informações do cluster são enviadas aos dois repositórios completos, podem existir situações nas quais você deseja criar uma segunda definição de canal CLUSSDR. É possível definir um segundo canal CLUSSDR em um cluster que possui muitos repositórios completos dispersos sobre uma ampla área. É possível, então, controlar para qual dos dois repositórios completos suas informações são enviadas.

Convenções de nomenclatura do cluster

Considere nomear gerenciadores de filas no mesmo cluster usando uma convenção de nomenclatura que identifica o cluster ao qual o gerenciador de filas pertence. Use uma convenção de nomenclatura semelhante para nomes de canais e estenda-a para descrever as características do canal.

Melhores práticas ao nomear Clusters do MQ

Embora os nomes de cluster possam ter até 48 caracteres, nomes de cluster relativamente curtos são úteis ao aplicar convenções de nomenclatura a outros objetos. Consulte [“Melhores práticas ao escolher nomes de canais de cluster”](#) na página 35.

Ao escolher um nome de cluster, geralmente é útil representar o 'propósito' do cluster (que provavelmente será de longa duração) em vez do 'conteúdo'. Por exemplo 'B2BPROD' ou 'ACTTEST' em vez de 'QM1_QM2_QM3_CLUS'.

Melhores práticas ao escolher nomes do Gerenciador de filas do cluster

Se você estiver criando um novo cluster e seus membros do zero, considere uma convenção de nomenclatura para os gerenciadores de filas que reflete seu uso do cluster. Cada gerenciador de filas deve ter um nome diferente. No entanto, é possível fornecer aos gerenciadores de filas em um cluster um conjunto de nomes semelhantes, para ajudar a identificar e lembrar agrupamentos lógicos (por exemplo, 'ACTTQM1, ACTTQM2).

Nomes de gerenciadores de filas relativamente curtos (por exemplo, menos de 8 caracteres) ajudam se você optar por usar a convenção descrita na próxima seção, ou algo semelhante, para nomes de canais.

Melhores práticas ao escolher nomes de canais de cluster

Como os gerenciadores de filas e os clusters podem ter nomes de até 48 caracteres e um nome de canal é limitado a 20 caracteres, tome cuidado ao nomear objetos pela primeira vez para evitar ter que mudar a convenção de nomenclatura no meio de um projeto (consulte a seção anterior).

Ao definir canais, lembre-se de que os canais do emissor de clusters criados automaticamente em qualquer gerenciador de filas no cluster obtêm seu nome do canal do receptor de clusters correspondente configurado no gerenciador de filas de recebimento no cluster e, portanto, eles devem ser exclusivos e fazer sentido *em gerenciadores de filas remotas no cluster*.

Uma abordagem comum é usar o nome do gerenciador de filas precedido pelo nome do cluster.. Por exemplo, se o nome do cluster for CLUSTER1 e os gerenciadores de filas forem QM1, QM2, então os canais receptores de cluster são CLUSTER1.QM1 e CLUSTER1.QM2.

Você pode estender essa convenção se os canais tiverem prioridades diferentes ou usarem protocolos diferentes Por exemplo:

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

Neste exemplo, S1 pode ser o primeiro canal SNA, N3 pode ser o canal NetBIOS com uma prioridade de rede de três e T4 pode ser IP TCP usando uma rede IPV4 .

Nomeando definições de canal compartilhado

Uma única definição de canal pode ser compartilhada entre vários clusters, caso em que as convenções de nomenclatura sugeridas aqui precisariam de modificação... No entanto, conforme descrito em [Gerenciando definições de canal](#) , geralmente é preferível definir canais distintos para cada cluster em qualquer caso.

Convenções de nomenclatura de canal mais antigas

Fora dos ambientes em cluster, tem sido historicamente comum usar uma convenção de nomenclatura 'FROMQM.T0.TARGETQM', portanto, você pode achar que os clusters existentes usaram

algo semelhante (como CLUSTER . TO . TARGET) Isso não é recomendado como parte de um novo esquema de nomenclatura de cluster porque reduz ainda mais os caracteres disponíveis para transmitir informações 'úteis' dentro do nome do canal.

z/OS

Nomes de canais no IBM MQ for z/OS

É possível definir recursos genéricos do VTAM ou nomes genéricos do *Dynamic Domain Name Server* (DDNS). É possível definir nomes de conexão usando nomes genéricos. No entanto, ao criar uma definição do receptor de cluster, não use um nome de conexão genérico.

O problema com o uso de nomes de conexão genéricos para definições de receptor de cluster é o seguinte: Se você definir um CLUSRCVR com um CONNAME genérico, não haverá garantia de que seus canais do CLUSSDR apontem para os gerenciadores de filas que você pretende.. Seu CLUSSDR inicial pode acabar apontando para qualquer gerenciador de filas no grupo de filas compartilhadas, não necessariamente um que hospeda um repositório completo. Se um canal começar a tentar uma conexão novamente, ele poderá reconectar a um gerenciador de filas diferente com o mesmo nome genérico, interrompendo o fluxo de mensagens.

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.

É possível definir recursos genéricos do VTAM ou nomes genéricos do *Dynamic Domain Name Server* (DDNS). É possível definir nomes de conexão usando nomes genéricos. No entanto, ao criar uma definição do receptor de cluster, não use um nome de conexão genérico.

O problema com o uso de nomes de conexão genéricos para definições de receptor de cluster é o seguinte: Se você definir um CLUSRCVR com um CONNAME genérico, não haverá garantia de que seus canais do CLUSSDR apontem para os gerenciadores de filas que você pretende.. Seu CLUSSDR inicial pode acabar apontando para qualquer gerenciador de filas no grupo de filas compartilhadas, não necessariamente um que hospeda um repositório completo. Se um canal começar a tentar uma conexão novamente, ele poderá reconectar a um gerenciador de filas diferente com o mesmo nome genérico, interrompendo o fluxo de mensagens.

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.

Sobrepondo Clusters

Sobrepor clusters fornece capacidades administrativas adicionais. Use as listas de nomes para reduzir o número de comandos necessários para administrar clusters de sobreposição.

É possível criar clusters que se sobrepõem. Há várias razões pelas quais é possível definir clusters de sobreposição; por exemplo:

- Para permitir que diferentes organizações tenham suas próprias administrações.

- Para permitir que aplicativos independentes sejam administrados separadamente.
- Para criar classes de serviço.

No [Figura 7 na página 37](#), o gerenciador de filas STF2 é um membro de ambos os clusters. Quando um gerenciador de filas é um membro de mais de um cluster, é possível aproveitar listas de nomes para reduzir o número de definições necessárias. As listas de nomes contêm uma lista de nomes, por exemplo, nomes de cluster. É possível criar uma lista de nomes nomeando os clusters. Especifique a lista de nomes no comando ALTER QMGR para STF2 para torná-lo um gerenciador de filas de repositório completo para ambos os clusters.

Se você tiver mais de um cluster em sua rede, deverá fornecer a eles nomes diferentes. Se dois clusters com o mesmo nome forem sempre mesclados, não será possível separá-los novamente. Também é uma boa ideia fornecer aos clusters e canais nomes diferentes. É possível diferenciá-los facilmente quando se olha para a saída a partir dos comandos do DISPLAY. Nomes do gerenciador de filas devem ser exclusivos em um cluster para ele funcionar corretamente.

Definindo Classes de Serviço

Imagine uma universidade que possui um gerenciador de filas para cada membro da equipe e cada estudante. As mensagens entre membros da equipe devem viajar por canais com uma alta prioridade e alta largura da banda. As mensagens entre estudantes devem viajar em canais mais baratos e mais lentos. Você pode configurar esta rede usando técnicas de enfileiramento distribuído tradicionais. O IBM MQ selecionará quais canais usar ao consulta o nome da fila de destino e o nome do gerenciador de filas.

Para diferenciar claramente entre a equipe e os estudantes, você poderá agrupar seus gerenciadores de filas em dois clusters conforme mostrado em [Figura 7 na página 37](#). O IBM MQ move as mensagens para a fila de reuniões no cluster da equipe somente pelos canais que estão definidos nesse cluster. As mensagens para a fila de fofoca no cluster dos estudantes passam pelos canais definido nesse cluster e recebem a classe de serviço apropriada.

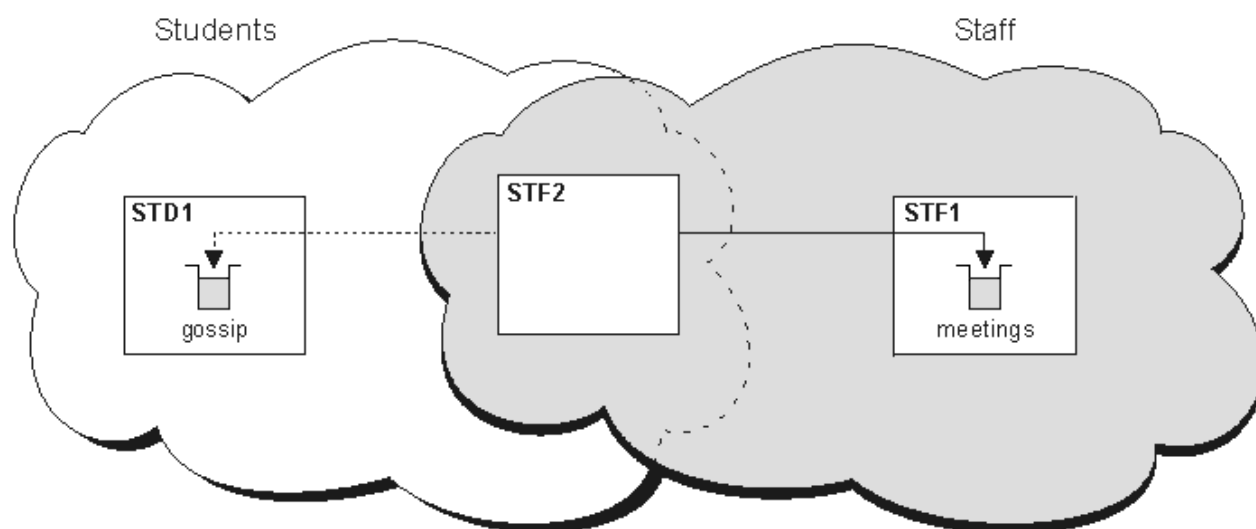


Figura 7. Classes de Serviço

Dicas de armazenamento em cluster

Pode ser necessário fazer algumas mudanças no sistemas ou aplicativos antes de usar o armazenamento em cluster. Há semelhanças e diferenças no comportamento de enfileiramento distribuído.

- Deve-se incluir definições de configuração manual para gerenciadores de filas fora de um cluster, para que eles acessem filas de clusters.
- Se você mesclar dois clusters com o mesmo nome, não poderá separá-los novamente. Portanto, é aconselhável dar a todos os clusters um nome exclusivo.
- Se uma mensagem chegar em um gerenciador de filas mas não houver nenhuma fila para recebê-la, a mensagem será colocada na fila de mensagens não entregues. Se não houver fila de mensagens não

entregues, o canal falhará e tentará novamente. O uso da fila de mensagens não entregues é o mesmo do enfileiramento distribuído.

- A integridade das mensagens persistentes será mantida. As mensagens não são duplicadas ou perdidas como resultado do uso de clusters.
- Usar clusters reduz a administração do sistema. Os clusters tornam fácil conectar redes grandes com muito mais gerenciadores de filas do que seriam capaz de contemplar usando enfileiramento distribuído. Há um risco de que você pode consumir recursos de rede excessivos se você tentar ativar a comunicação entre cada gerenciador de filas em um cluster.
- Se você usar o IBM MQ Explorer, que apresenta os gerenciadores de filas em uma estrutura em árvore, a visualização para clusters grandes poderá ser complicada.
- **Multi** O propósito das listas de distribuição deve usar um único comando MQPUT para enviar a mesma mensagem a vários destinos. Listas de distribuição são suportadas no IBM MQ for Multiplatforms. É possível usar listas de distribuição com clusters de gerenciador de filas. Em um cluster, todas as mensagens são expandidas no tempo de MQPUT. A vantagem, em termos de tráfego de rede, não é tão maior que em um ambiente que não é de armazenamento em cluster. A vantagem das listas de distribuição é que os inúmeros canais e filas de transmissão não precisam ser definidos manualmente.
- Se você for usar clusters para equilibrar sua carga de trabalho, examine seus aplicativos. Veja se eles requerem que mensagens sejam processadas por um gerenciador de filas específico ou em uma sequência específica. Tais aplicativos são ditos como tendo afinidades de mensagens. Talvez seja necessário modificar seus aplicativos antes de poder usá-los em clusters complexos.
- É possível optar por usar a opção MQOO_BIND_ON_OPEN em um MQOPEN para forçar mensagens a serem enviadas para um destino específico. Se o gerenciador de filas de destino não estiver disponível, as mensagens não serão entregues até que o gerenciador de filas se torne disponível novamente. As mensagens não são roteadas para outro gerenciador de filas devido ao risco de duplicação.
- Se um gerenciador de filas tiver que hospedar um repositório de cluster, será necessário saber seu nome de host ou endereço IP. É necessário especificar essas informações no parâmetro CONNAME ao criar a definição de CLUSSDR em outros gerenciadores de filas unidos ao cluster. Se você usar DHCP, o endereço IP está sujeito à mudança, porque o DHCP pode alocar um novo endereço IP toda vez que você reiniciar um sistema. Portanto, não se deve especificar o endereço IP nas definições do CLUSSDR. Mesmo se todas as definições de CLUSSDR especificassem o nome do host em vez do endereço IP, as definições ainda não seriam ser confiáveis. O DHCP não atualiza necessariamente a entrada de diretório DNS para o host com o novo endereço. Se tiver que nomear gerenciadores de filas como repositórios completos em sistemas que usam DHCP, instale um software que garanta que seu diretório DNS fique atualizado.
- Não use nomes genéricos, por exemplo, recursos genéricos de VTAM ou nomes genéricos de Dynamic Domain Name Server (DDNS), como os nomes de conexão para seus canais. Se você o fizer, seus canais poderão conectar-se a um gerenciador de filas diferente do esperado.
- É possível obter uma mensagem apenas de uma fila de clusters local, mas você pode colocar uma mensagem em qualquer fila em um cluster. Se você abrir uma fila para usar o comando MQGET, o gerenciador de filas abre a fila local.
- Não é necessário alterar nenhum de seus aplicativos se você configurar um cluster simples do IBM MQ. O aplicativo pode nomear a fila de destino na chamada MQOPEN e não é necessário saber sobre o local do gerenciador de filas. Se você configurar um cluster para gerenciamento de carga de trabalho, deverá revisar seus aplicativos e modificá-los conforme necessário.
- É possível visualizar dados de monitoramento e status atuais para um canal ou fila que usa os comandos DISPLAY CHSTATUS e DISPLAY QSTATUS **runmqsc**. As informações de monitoramento podem ser usadas para ajudar a calibrar o desempenho e o funcionamento do sistema. O monitoramento é controlado pelos atributos de canal, fila e gerenciador de filas. O monitoramento de canais do emissor de clusters autodefinidos é possível com o atributo do gerenciador de filas MONACLS.

Conceitos relacionados

Clusters

[“Comparação de Armazenamento em Cluster e Enfileiramento Distribuído” na página 30](#)

Compare os componentes que precisam ser definidos para conectar gerenciadores de filas usando enfileiramento e armazenamento em cluster distribuído.

Componentes de um Cluster

Tarefas relacionadas

Configurando um cluster do gerenciador de filas

Configurando um novo cluster

Quanto Tempo os repositórios do gerenciador de filas retêm informações?

Os repositórios do gerenciador de filas retêm informações por 30 dias. Um processo automático atualiza de forma eficiente as informações que estão sendo usadas.

Quando um gerenciador de filas envia algumas informações sobre si, os gerenciadores de fila de repositório completo e parcial armazenam as informações por 30 dias. As informações são enviadas, por exemplo, quando um gerenciador de filas informa a criação de uma nova fila. Para evitar que estas informações expirem, os gerenciadores de filas reenviam automaticamente todas as informações sobre eles após 27 dias. Se um repositório parcial enviar uma nova solicitação de informação parcialmente concluída para o tempo de vida de 30 dias, o tempo de expiração permanece o original de 30 dias.

Quando as informações expiram, elas não são removidas imediatamente do repositório. Em vez disso, elas são mantidas por um período de carência de 60 dias. Se nenhuma atualização for recebida dentro do período de carência, as informações são removidas. O período de carência permite o fato de que um gerenciador de filas pode ter sido colocado temporariamente fora de serviço na data de validade. Se um gerenciador de filas ficar desconectado a partir de um cluster por mais de 90 dias, ele deixará de ser parte do cluster. No entanto, se ele se reconectar à rede, tornará parte do cluster novamente. Os repositórios completos não usam informações que expiraram para satisfazer novos pedidos de outros gerenciadores de filas.

De modo semelhante, quando um gerenciador de filas envia um pedido por informações atualizadas a partir de um repositório completo, o pedido dura 30 dias. Após 27 dias, o IBM MQ verifica a solicitação. Se ele tiver sido referenciado durante os 27 dias, ele será atualizado automaticamente. Se não, será deixado para expirar e será atualizado pelo gerenciador de filas se for necessário novamente. Expirar solicitações impede uma construção de solicitações de informações a partir de gerenciadores de filas inativos.

Nota: Você deve fazer o download e instalar o PTF para [APAR PH43191](#), que corrige erros do sistema no cálculo do tempo de expiração de uma assinatura. Esses erros podem fazer com que a assinatura expire antecipadamente (resultando na emissão da mensagem CSQX456I) ou expire após a expiração do objeto (resultando em erros errôneos do MQRC 2085 (MQRC_UNKNOWN_OBJECT)).

Para clusters grandes, ele pode ser interrompido se vários gerenciadores de filas reenviam automaticamente todas as informações sobre ele ao mesmo tempo. Consulte [Atualizando em um grande cluster pode afetar o desempenho e disponibilidade do cluster](#).

Conceitos relacionados

[“Armazenamento em Cluster: Usando Melhores Práticas de REFRESH CLUSTER” na página 70](#)

Use o comando **REFRESH CLUSTER** para descartar todas as informações sobre um cluster mantidas localmente e reconstruir essas informações dos repositórios completos no cluster. Não deve ser necessário usar esse comando, exceto em circunstâncias excepcionais. Se você precisar usá-lo, existem considerações especiais sobre como fazer isso. Essas informações são um guia baseado em teste e feedback de clientes.

Clusters de exemplo

O primeiro exemplo mostra o menor cluster possível de dois gerenciadores de filas. O segundo e terceiro exemplos mostram duas versões de um cluster de três gerenciadores de filas.

O menor cluster possível contém somente dois gerenciadores de filas. Neste caso, ambos os gerenciadores de filas contêm repositórios completos. É necessário apenas algumas definições para configurar o cluster e ainda existe um alto grau de autonomia em cada gerenciador de filas.

DEMOCLSTR

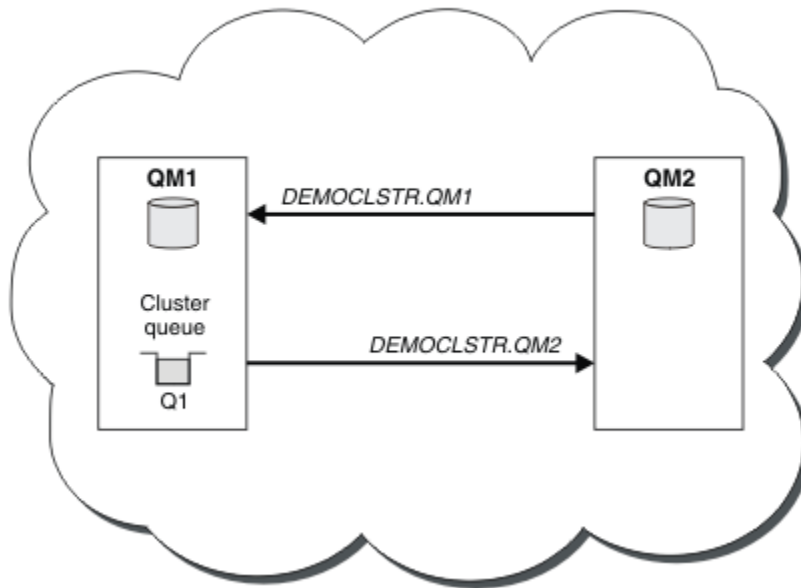



Figura 8. Um Cluster Pequeno de Dois Gerenciadores de Filas

- Os gerenciadores de filas podem ter nomes longos como LONDON e NEWYORK.  No IBM MQ for z/OS, os nomes do gerenciador de filas são limitados a quatro caracteres.
- Cada gerenciador de filas é geralmente configurado em uma máquina separada. No entanto, você pode ter vários gerenciadores de filas na mesma máquina.

Para obter instruções sobre como configurar um cluster de exemplo semelhante, veja [Configurando um novo cluster](#).

Figura 9 na página 41 mostra os componentes de um cluster chamado CLSTR1.

- Neste cluster, há três gerenciadores de filas, QM1, QM2 e QM3.
- O QM1 e o QM2 hospedam repositórios de informações sobre todos os gerenciadores de filas e objetos relacionados ao cluster no cluster. Eles são referidos como *gerenciadores de fila de repositório completo*. Os repositórios são representados no diagrama pelos cilindros sombreados.
- O QM2 e o QM3 hospedam algumas filas que são acessíveis para qualquer outro gerenciador de filas no cluster. As filas que estão acessíveis para qualquer outro gerenciador de filas no cluster são chamadas de *filas de clusters*. As filas de clusters são representadas no diagrama pelas filas sombreadas. As filas de clusters são acessíveis a partir de qualquer local no cluster. O código de armazenamento em cluster do IBM MQ assegura que as definições de filas remotas para filas de clusters sejam criadas em qualquer gerenciador de filas que se refere a elas.

Como com o enfileiramento distribuído, um aplicativo usa a chamada MQPUT para colocar uma mensagem em uma fila de clusters em qualquer gerenciador de filas no cluster. Um aplicativo usa a chamada MQGET para recuperar as mensagens de uma fila de clusters apenas no gerenciador de filas em que a fila reside.

- Cada gerenciador de filas tem uma definição criada manualmente para a extremidade de recebimento de um canal chamado `cluster_name.queue_manager_name` no qual ele pode receber mensagens. No gerenciador de filas de recebimento, o `cluster_name.queue_manager_name` é um canal de receptor de clusters. Um canal do receptor de clusters é como um canal receptor usado no enfileiramento distribuído; ele recebe as mensagens para o gerenciador de filas. Além disso, também recebe as informações sobre o cluster.

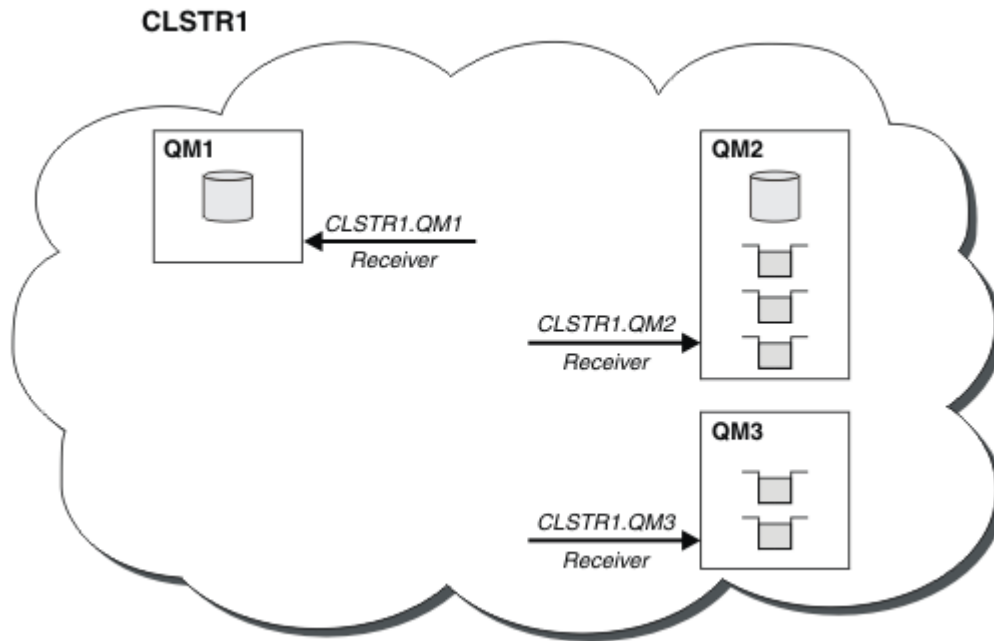


Figura 9. Um Cluster de Gerenciadores de Filas

- Em [Figura 10 na página 42](#), cada gerenciador de filas também tem uma definição para a extremidade de envio de um canal. Ele se conecta ao canal do receptor de clusters de um dos gerenciadores de filas de repositório completo. No gerenciador de filas de envio, o `cluster_name.queue_manager_name` é um canal de emissor de clusters. O QM1 e o QM3 possuem canais do emissor de clusters conectando-se a CLSTR1.QM2, consulte a linha pontilhada "2".

O QM2 possui um canal do emissor de clusters conectando-se a CLSTR1.QM1, consulte a linha pontilhada "3". Um canal do emissor de clusters é como um canal emissor usado no enfileiramento distribuído; ele envia mensagens para o gerenciador de filas de recebimento. Além disso, também envia as informações sobre o cluster.

Quando a extremidade do receptor de cluster e a extremidade do emissor de cluster de um canal são definidas, o canal inicia automaticamente.

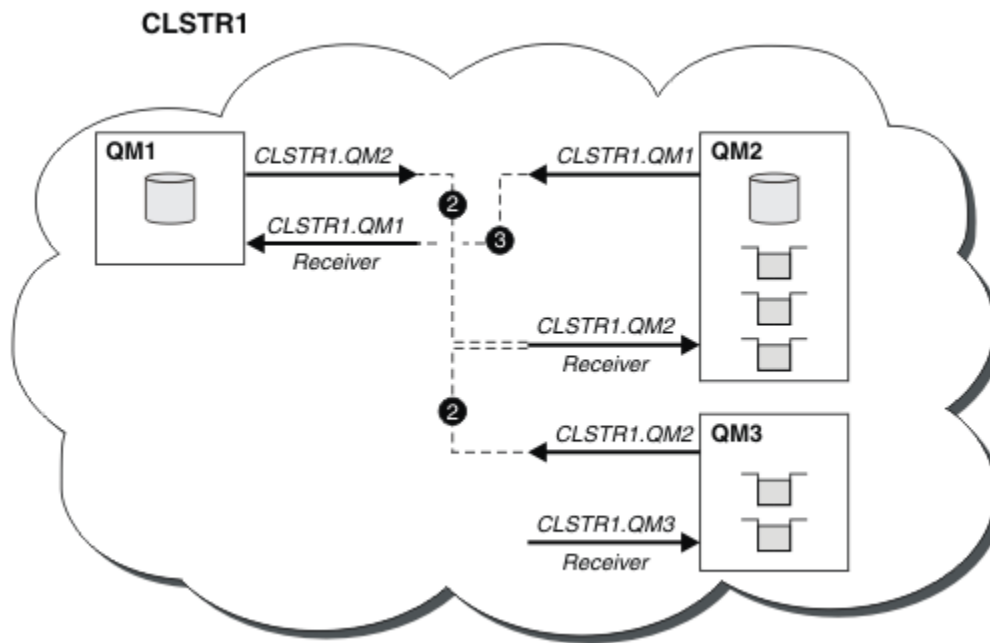


Figura 10. Um cluster de Gerenciadores de Filas com Canais Emissores

Definir um canal do emissor de clusters no gerenciador de filas locais introduz esse gerenciador de filas em um dos gerenciadores de filas de repositório completo. O gerenciador de filas de repositório completo atualiza as informações em seu repositório completo de acordo. Em seguida, ele cria automaticamente um canal do emissor de clusters de volta no gerenciador de filas original e envia essas informações do gerenciador de filas sobre o cluster. Assim, um gerenciador de filas aprende sobre um cluster e um cluster aprende sobre um gerenciador de filas.

Consulte Figura 9 na página 41 novamente. Suponha que um aplicativo conectado ao gerenciador de filas QM3 deseje enviar algumas mensagens para as filas no QM2. A primeira vez que o QM3 deve acessar essas filas, ele as descobre ao consultar um repositório completo. O repositório completo neste caso é QM2, que é acessado usando o canal do emissor CLSTR1.QM2. Com as informações do repositório, ele pode criar automaticamente definições remotas para essas filas. Se as filas estiverem no QM1, esse mecanismo ainda funciona, pois QM2 é um repositório completo. Um repositório completo possui um registro completo de todos os objetos no cluster. Neste último caso, o QM3 também criaria automaticamente um canal do emissor de clusters correspondente ao canal do receptor de clusters no QM1, permitindo a comunicação direta entre os dois.

Figura 11 na página 43 mostra o mesmo cluster, com os dois canais do emissor de clusters que foram criados automaticamente. Os canais de emissor de clusters são representados pelas duas linhas tracejadas que se unem com o canal do receptor de clusters CLSTR1.QM3. Ele também mostra a fila de transmissão do cluster, SYSTEM.CLUSTER.TRANSMIT.QUEUE, a qual o QM1 usa para enviar suas mensagens. Todos os gerenciadores de filas no cluster possuem uma fila de transmissão de cluster, a partir da qual eles podem enviar mensagens a qualquer outro gerenciador de filas no mesmo cluster.

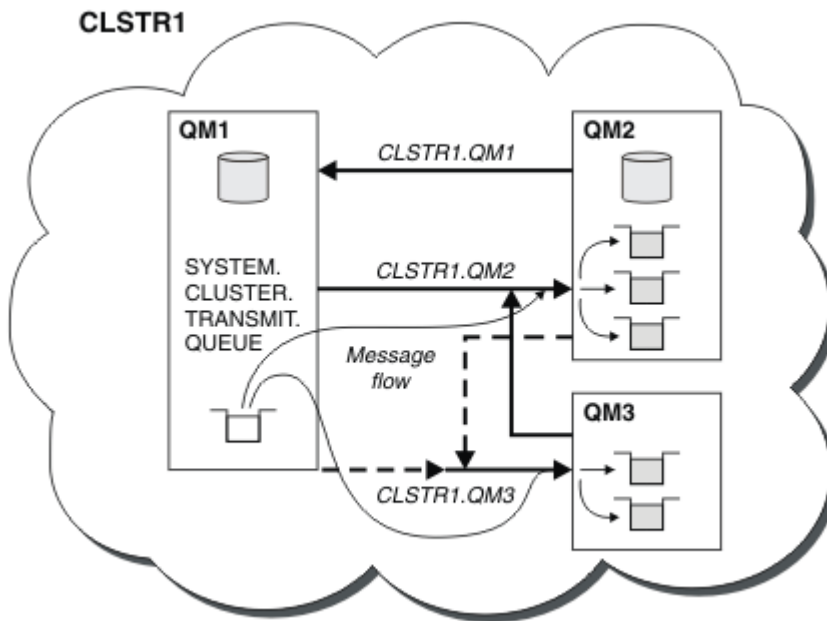


Figura 11. Um Cluster de Gerenciadores de Filas, Mostrando Canais Autodefinidos

Nota: Outros diagramas mostram somente as extremidades de recebimento de canais para as quais você faz definições manuais. As extremidades de envio são omitidas porque são definidas na maioria das vezes automaticamente quando necessário. A autodefinição da maioria dos canais do emissor de clusters é crucial para a função e a eficiência dos clusters.

Conceitos relacionados

[“Comparação de Armazenamento em Cluster e Enfileiramento Distribuído” na página 30](#)

Compare os componentes que precisam ser definidos para conectar gerenciadores de filas usando enfileiramento e armazenamento em cluster distribuído.

[Componentes de um Cluster](#)

Tarefas relacionadas

[Configurando um cluster do gerenciador de filas](#)

[Configurando um novo cluster](#)

Armazenamento em cluster: melhores práticas

Os clusters fornecem um mecanismo para interconexão de gerenciadores de filas. As melhores práticas descritas nesta seção são baseadas em teste e feedback dos clientes.

Uma configuração de cluster bem-sucedida depende de um bom agrupamento e de um entendimento completo dos fundamentos do IBM MQ, como um bom gerenciamento de aplicativo e design de rede. Certifique-se de que esteja familiarizado com as informações nos tópicos relacionados antes de continuar.

Conceitos relacionados

[Enfileiramento distribuído e clusters](#)

[Clusters](#)

Tarefas relacionadas

[“Projetando Clusters” na página 24](#)

Os clusters fornecem um mecanismo para interconectar gerenciadores de filas de uma maneira que simplifique a configuração inicial e o gerenciamento em andamento. Os clusters devem ser cuidadosamente projetados, para assegurar que funcionem corretamente e que obtenham os níveis necessários de disponibilidade e responsividade.

Monitorando clusters

Armazenamento em cluster: considerações especiais para clusters sobrepostos

Este tópico fornece orientação para planejamento e administração de clusters do IBM MQ. Essas informações são um guia baseado em teste e feedback de clientes.

Propriedade do cluster

Familiarize-se com os clusters sobrepostos antes de ler as informações a seguir. Veja [“Sobrepondo Clusters”](#) na página 36 e [Configurando caminhos de mensagem entre clusters](#) para obter as informações necessárias.

Ao configurar e gerenciar um sistema que consista em clusters sobrepostos, é melhor aderir ao seguinte:

- Embora os clusters do IBM MQ sejam "fracamente acoplados" conforme descrito anteriormente, é útil considerar um cluster como uma única unidade de administração. Esse conceito é usado porque a interação entre definições de gerenciadores de filas individuais é fundamental para o funcionamento harmonioso do cluster. Por exemplo: ao usar filas de clusters com balanceamento de carga de trabalho de trabalho, é importante que um único administrador ou equipe entenda o conjunto completo de possíveis destinos para mensagens, o que depende das definições difundidas por todo o cluster. De modo mais simples, pares de canais de emissor/receptor de cluster devem ser compatíveis globalmente.
- Considerando esse conceito anterior; quando vários clusters se encontram (que devem ser administrados por equipes/pessoas separadas), é importante ter políticas claras estabelecidas para controlar a administração dos gerenciadores de filas de gateway.
- É útil tratar clusters sobrepostos como um único espaço de nomes: nomes de canal e nomes de gerenciador de filas devem ser exclusivos em um único cluster. A administração é muito mais fácil quando exclusiva em toda a topologia. É melhor seguir uma convenção de nomenclatura apropriada; convenções possíveis são descritas em [“Convenções de nomenclatura do cluster”](#) na página 35.
- Por vezes, a cooperação administrativa e de gestão do sistema é essencial. " Por exemplo, cooperação entre organizações que possuem clusters diferentes que precisam se sobrepor. Um entendimento claro de quem possui o quê e regras e convenções aplicáveis ajudam o armazenamento em cluster a executar sem problemas ao sobrepor clusters.

Clusters sobrepostos: gateways

Em geral, é mais fácil administrar um único cluster do que vários clusters. Portanto, criar grandes números de clusters pequenos (um para cada aplicativo, por exemplo) costuma ser algo a se evitar.

No entanto, para fornecer classes de serviço, é possível implementar clusters sobrepostos. Por exemplo:

- Se você tiver clusters concêntricos em que o menor é para Publicação / Assinatura. Veja [Como dimensionar sistemas](#) para obter mais informações.
- Se alguns gerenciadores de filas forem administrados por equipes diferentes. Consulte a seção anterior [“Propriedade do cluster”](#) na página 44 para obter mais informações
- Se isso fizer sentido de um ponto de vista organizacional ou geográfico.
- Se clusters equivalentes funcionarem com resolução de nome, por exemplo, ao implementar TLS em um cluster existente.

Não há nenhum benefício de segurança de clusters sobrepostos; permitindo que clusters administrados por duas equipes diferentes se sobreponham, junte efetivamente as equipes, bem como a topologia:

- Qualquer nome anunciado em tal cluster é acessível para o outro cluster.
- Qualquer nome anunciado em um cluster pode ser anunciado no outro para atrair mensagens elegíveis.
- Qualquer objeto não anunciado em um gerenciador de filas adjacente ao gateway pode ser resolvido a partir de quaisquer clusters dos quais o gateway é um membro

O namespace é a união de ambos os clusters e deve ser tratado como um único namespace. Portanto, a propriedade de um cluster sobreposto é compartilhado entre todos os administradores de ambos os clusters.

Quando um sistema contém vários clusters, pode haver um requisito de rotear mensagens de gerenciadores de filas em um cluster para filas nos gerenciadores de filas em outro cluster. Nessa situação, a vários clusters devem estar interconectados de alguma forma: um bom padrão a seguir é o uso de gerenciadores de filas de gateway entre clusters. Esse acordo evita a criação de uma rede de canais ponto-a-ponto difícil de gerenciar e fornece um bom local para gerenciar tais problemas como políticas de segurança. Há dois modos distintos de alcançar este acordo:

1. Coloque um (ou mais) gerenciador de filas em ambos os clusters usando uma segunda definição do receptor de cluster. Esse acordo envolve menos definições administrativas, mas, conforme já mencionado, significa que a propriedade de um cluster de sobreposição é compartilhada entre todos os administradores de ambos os clusters.
2. Emparelhe um gerenciador de filas no cluster um com um gerenciador de filas no cluster dois usando os canais tradicionais ponto a ponto.

Em qualquer desses casos, várias ferramentas podem ser usadas para rotear o tráfego apropriadamente. Em particular, aliases de fila ou de gerenciador de filas podem ser usados para rotear outro cluster e um alias de gerenciador de filas com propriedade em branco do **RQMNAME** redireciona o balanceamento de carga de trabalho onde ele é procurado.

Conceitos relacionados

[“Convenções de nomenclatura do cluster”](#) na página 35

Considere nomear gerenciadores de filas no mesmo cluster usando uma convenção de nomenclatura que identifica o cluster ao qual o gerenciador de filas pertence. Use uma convenção de nomenclatura semelhante para nomes de canais e estenda-a para descrever as características do canal.

Armazenamento em cluster: considerações sobre o design de topologia

Este tópico fornece orientação para planejamento e administração de clusters do IBM MQ. Essas informações são um guia baseado em teste e feedback de clientes.

Ao pensar onde aplicativos de usuário e processos administrativos internos serão localizados antecipadamente, muitos problemas podem ser evitados ou minimizada em uma data posterior. Este tópico contém informações sobre as decisões de projeto que podem melhorar o desempenho e simplificar tarefas de manutenção conforme o agrupamento aumenta.

- [“Desempenho da infraestrutura de cluster”](#) na página 45
- [“Repositórios completos”](#) na página 46
- [“Os aplicativos devem utilizar filas nos repositórios completos?”](#) na página 47
- [“Gerenciando definições de canal”](#) na página 48
- [“Balanceamento de carga sobre vários canais”](#) na página 48

Desempenho da infraestrutura de cluster

Quando um aplicativo tenta abrir uma fila em um gerenciador de filas em um cluster, o gerenciador de filas registrará seu interesse com os repositórios completos para aquela fila para que ele possa saber onde fica a fila no cluster. Todas as atualizações para o local ou à configuração da fila serão automaticamente enviadas pelos repositórios completos para o gerenciador de filas interessado. Esse registro de interesse é internamente conhecido como assinatura (essas assinaturas não são as mesmas que as assinaturas do IBM MQ utilizadas para mensagens de publicação/assinatura no IBM MQ)

Todas as informações sobre um cluster passam por todos os repositórios completos. Repositórios completos, portanto, sempre estão sendo utilizados em um cluster para o tráfego de mensagens administrativas. O alto uso de recursos do sistema ao gerenciar essas assinaturas, e a transmissão delas e das mensagens de configuração resultantes, pode causar um carregamento considerável na infraestrutura de armazenamento em cluster. Existem vários pontos a considerar ao assegurar que este carregamento seja entendido e minimizado sempre que possível:

- Quanto mais gerenciadores de filas individuais houver usando uma fila de clusters, mais assinaturas haverá no sistema e, portanto, maior será a sobrecarga administrativa quando ocorrerem mudanças e os assinantes interessados precisarem ser notificados, especialmente nos gerenciadores de filas de repositório completo. Uma maneira de minimizar o tráfego desnecessário e a carga total do repositório é através da conexão de aplicativos semelhantes (ou seja, aplicativos que funcionam com as mesmas filas) a um número menor de gerenciadores de filas.
- Além do número de assinaturas no sistema que afetam o desempenho, a taxa de mudança na configuração de objetos em cluster pode afetar o desempenho, por exemplo, a alteração frequente de uma configuração de fila em cluster.
- Quando um gerenciador de filas é um membro de vários clusters (ou seja, ele faz parte de um sistema de cluster de sobreposição) qualquer interesse manifestado em uma fila resulta em uma assinatura para cada cluster da qual ela é membro, ainda que os mesmos gerenciadores de filas sejam os repositórios completos para mais de um dos clusters. Essa organização aumenta a carga no sistema, e é uma razão para considerar se vários clusters de sobreposição são necessários, em vez de um único cluster.
- O tráfego de mensagem do aplicativo (ou seja, as mensagens enviadas por aplicativos IBM MQ para as filas de clusters) não vai por meio dos repositórios completos para atingir os gerenciadores de filas de destino. Esse o tráfego de mensagens é enviado diretamente entre o gerenciador de filas no qual a mensagem entra no cluster e o gerenciador de filas no qual a fila do cluster existe. Não é, portanto, necessário acomodar altas taxas de tráfego de mensagem do aplicativo com relação aos gerenciadores de filas de repositório completo, a menos que os gerenciadores de filas do repositório completo sejam um desses dois gerenciadores de filas referidos. Por essa razão, é recomendável que gerenciadores de filas de repositório completo não sejam utilizados para o tráfego de mensagem do aplicativo em clusters em que a carga de infraestrutura de cluster seja significativa.

Repositórios completos

Um repositório é uma coleção de informações sobre os gerenciadores de filas que são membros de um cluster. Um gerenciador de filas que hospeda um conjunto completo de informações sobre cada gerenciador de filas no cluster tem um repositório completo. Para obter mais informações sobre repositórios completos e repositórios parciais, consulte [Repositório de cluster](#).

repositórios completos devem ser mantidos em servidores confiáveis e o mais altamente disponível possível, e pontos únicos de falha devem ser evitados. O design do cluster deve sempre ter dois repositórios completos. Se houver uma falha no repositório completo, o cluster ainda poderá operar.

Detalhes de quaisquer atualizações aos recursos do cluster feitas por um gerenciador de filas em um cluster; por exemplo, filas em cluster são enviadas a partir desse gerenciador de filas para dois repositórios completos no máximo nesse cluster (ou para um, se houver apenas um gerenciador de filas de repositório completo no cluster). Esses repositórios completos retêm as informações e as propagam para todos os gerenciadores de filas no cluster que mostram interesse nelas (ou seja, as assinam). Para assegurar que cada membro do cluster tenha uma visualização atualizada dos recursos do cluster lá, cada gerenciador de filas deve ser capaz de se comunicar com pelo menos um gerenciador de filas de repositório completo a qualquer momento.

Se, por algum motivo, um gerenciador de filas não puder se comunicar com qualquer repositório completo, ele poderá continuar funcionando no cluster com base em seu nível de informações já em cache por um período de tempo, mas nenhuma nova atualização nem acesso a recursos do cluster anteriormente não utilizados estará disponível.

Por isso, procure manter os dois repositórios completos disponíveis o tempo todo. No entanto, esse acordo não significa que medidas extremas devam ser tomadas, pois o cluster funciona de forma adequada por um breve período sem um repositório completo.

Há outro motivo pelo qual um cluster deve ter dois gerenciadores de filas de repositório completo, além da disponibilidade de informações de cluster: esse motivo é assegurar que as informações do cluster mantidas no cache de repositório completo existam em dois locais para fins de recuperação. Se houver apenas um repositório completo, e ele perder suas informações sobre o cluster, uma intervenção manual em todos os gerenciadores de filas no cluster será necessária para fazer o cluster funcionar novamente.

Se houver dois repositórios completos, uma vez que as informações são sempre publicadas e assinadas a partir de dois repositórios completos, o repositório completo com falha poderá ser recuperado com um mínimo de esforço.

- É possível executar manutenção em gerenciadores de filas de repositório completo em um design de cluster de dois repositórios completos sem afetar os usuários desse cluster: o cluster continuará funcionando com apenas um repositório, portanto, quando for possível desativar repositórios, aplique a manutenção e reative os repositórios um por vez. Mesmo que haja uma interrupção no segundo repositório completo, os aplicativos em execução não são afetados por um mínimo de três dias.
- A menos que haja uma boa razão para utilizar um repositório de terceiros, como utilizar um repositório completo geograficamente local por razões geográficas, use o design de dois repositórios. Ter três repositórios completos significa que você nunca sabe quais dois estão atualmente em uso, e pode haver problemas administrativos causados por interações entre os diferentes parâmetros de gerenciamento de carga de trabalho. Não é recomendável ter mais de dois repositórios completos.
- Se você ainda precisar de melhor disponibilidade, considere hospedar os gerenciadores de filas de repositório completo como gerenciadores de filas de várias instâncias ou usar suporte específico da plataforma de alta disponibilidade para melhorar sua disponibilidade.
- Você deve interconectar por completo todos os gerenciadores de fila de repositório completo com canais do remetente definidos manualmente. Cuidado especial deve ser tomado quando o cluster tiver, por alguma razão justificável, mais de dois repositórios completos. Nessa situação, costuma ser possível perder um ou mais canais e ele não estar imediatamente aparente. Quando interconexão completa não ocorrer, costumam surgir problemas difíceis de diagnosticar. Eles são difíceis de diagnosticar porque alguns repositórios completos não contêm todos os dados no repositório e, portanto, resultam em os gerenciadores de filas no cluster terem visualizações diferentes do cluster dependendo dos repositórios completos aos quais eles se conectam.

Os aplicativos devem utilizar filas nos repositórios completos?

Um repositório completo é, de um modo geral, exatamente como qualquer outro gerenciador de filas, portanto, é possível hospedar as filas do aplicativo no repositório completos e conectar os aplicativos diretamente a esses gerenciadores de filas. Os aplicativos devem utilizar filas nos repositórios completos?

A resposta comumente aceita é "Não?". Embora essa configuração seja possível, muitos clientes preferem manter esses gerenciadores de filas dedicado à manutenção do cache do cluster do repositório completo. Pontos a considerar ao decidir quanto a qualquer uma das opções são descritos aqui, mas, em última instância, a arquitetura do cluster deve ser adequada às exigências específicas do ambiente.

- Upgrades: geralmente, a fim de utilizar recursos do novo cluster em novas liberações do IBM MQ, os gerenciadores de filas de repositório completo desse cluster devem ser atualizados primeiro. Quando um aplicativo no cluster deseja usar novos recursos, pode ser útil ser capaz de atualizar os repositórios completos (e algum subconjunto de repositórios parciais) sem testar um número de aplicativos colocalizados.
- Manutenção: de modo similar, se você precisar aplicar manutenção urgente aos repositórios completos, eles poderão ser reiniciados ou atualizados com um comando **REFRESH** sem afetar os aplicativos.
- Desempenho: conforme os clusters crescem e as demandas de manutenção do cache do cluster de repositório completo aumentam, manter os aplicativos separados reduz o risco de afetar o desempenho do aplicativo por meio de contenção para recursos do sistema.
- requisitos de hardware: geralmente, repositórios completos não precisam ser potentes; por exemplo, um simples servidor UNIX com uma boa expectativa de disponibilidade é suficiente. Como alternativa, para clusters muito grandes ou constantemente em alteração, o desempenho do computador do repositório completo deve ser considerado.
- Requisitos de software: requisitos geralmente são a principal razão para escolher hospedar filas de aplicativo em um repositório completo. Em um cluster pequeno, a colocalização pode significar um requisito geral menor de gerenciadores de filas/servidores.

Gerenciando definições de canal

Mesmo dentro de um único cluster, pode haver várias definições de canal apresentando várias rotas entre dois gerenciadores de filas.

Há, às vezes, uma vantagem em ter canais paralelos dentro de um único cluster, mas essa decisão de design deve ser considerada cuidadosamente, além de incluir complexidade, esse design pode resultar em subutilização de canais, o que reduz o desempenho. Essa situação ocorre porque o teste geralmente envolve o envio muitas mensagens a um ritmo constante, de modo que canais paralelos são totalmente utilizados. Mas, com condições reais de um fluxo de mensagens não constante, o algoritmo de balanceamento de carga faz o desempenho cair conforme o fluxo de mensagens muda de um canal para outro.

Quando um gerenciador de filas é um membro de vários clusters, existe a opção de utilizar uma única definição de canal com uma lista de nomes de cluster, em vez de definir um canal CLUSRCVR separado para cada cluster. No entanto, essa configuração pode causar dificuldades de administração mais tarde; considere, por exemplo, o caso em que o TLS deve ser aplicado a um cluster, mas não a outro. Nesse caso, é preferível criar definições separadas, e a convenção de nomenclatura sugerida no [“Convenções de nomenclatura do cluster”](#) na página 35 suporta isso.

Balanceamento de carga sobre vários canais

Essas informações são feitas para um entendimento avançado do assunto. Para a explicação básica do assunto (que deve ser entendido antes de utilizar as informações aqui), consulte [Usando clusters para gerenciamento de carga](#) e [Balanceamento de carga em clusters](#) e [O algoritmo de gerenciamento de carga de trabalho do cluster](#).

O algoritmo de gerenciamento de carga de trabalho do cluster fornece um grande conjunto de ferramentas, mas elas não devem ser todas usadas umas com as outras sem um completo entendimento de como elas funcionam e interagem. Pode não ser imediatamente óbvio como os canais são importantes para o processo de balanceamento de carga de trabalho: O algoritmo round-robin de gerenciamento de carga de trabalho se comporta como se vários canais de cluster para um gerenciador de filas que possui uma fila em cluster fossem tratados como várias instâncias dessa fila. Esse processo é explicado em mais detalhes no exemplo a seguir:

1. Há dois gerenciadores de filas hospedando uma fila em um cluster: QM1 e QM2.
2. Há cinco canais de receptor de cluster para QM1.
3. Há apenas um canal receptor de cluster para QM2.
4. Quando **MQPUT** ou **MQOPEN** no QM3 escolhe uma instância, o algoritmo tem cinco vezes mais chances de enviar a mensagem para QM1 do que para QM2.
5. A situação na etapa 4 ocorre porque o algoritmo vê seis opções para escolher entre (5+1) e round-robins em todos os cinco canais para QM1 e o canal único para QM2.

Outro comportamento sutil é que, mesmo ao colocar mensagens em uma fila de cluster que tenha uma instância configurada no gerenciador de filas locais, o IBM MQ usa o estado do canal receptor de cluster local para decidir se as mensagens devem ser colocadas na instância local da fila ou em instâncias remotas da fila. Nesse cenário:

1. Ao colocar mensagens, o algoritmo de gerenciamento de carga não olhar para as filas de clusters individuais, ele analisa os canais do cluster que podem chegar a esses destinos.
2. Para chegar aos destinos locais, os canais do receptor local são incluídos nessa lista (embora não sejam utilizados para enviar a mensagem).
3. Quando um canal receptor local é interrompido, o algoritmo de gerenciamento de carga de trabalho prefere uma instância alternativa por padrão se o seu CLUSRCVR não estiver parado. Se houver várias instâncias locais de CLUSRCVR para o destino e pelo menos uma não estiver parada, a instância local permanecerá elegível.

Armazenamento em cluster: isolamento de aplicativos usando diversas filas de transmissão do cluster
É possível isolar os fluxos de mensagens entre gerenciadores de filas em um cluster. É possível colocar mensagens que estão sendo transportadas por diferentes canais do emissor de clusters em diferentes filas de transmissão do cluster. É possível usar a abordagem em um único cluster ou com clusters sobrepostos. O tópico fornece exemplos e algumas melhores práticas para orientar a escolha de uma abordagem a usar.

Ao implementar um aplicativo, existe a opção de escolher quais recursos ele compartilha com outros aplicativos IBM MQ e quais recursos ele não compartilha. Há vários tipos de recursos que podem ser compartilhados, os principais sendo o próprio servidor, o gerenciador de filas, canais e filas. É possível escolher configurar aplicativos com menos recursos compartilhados; alocar filas, canais, gerenciadores de filas ou mesmo servidores separados para aplicativos individuais. Se você fizer isso, a configuração do sistema geral ficará maior e mais complexa. O uso de clusters IBM MQ reduz a complexidade de gerenciamento de mais servidores, gerenciadores de filas e filas e canais, mas introduz outro recurso compartilhado, a fila de transmissão do cluster, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.

O [Figura 12 na página 50](#) é uma fatia externa de uma grande implementação do IBM MQ que ilustra o significado do compartilhamento do `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. No diagrama, o aplicativo, `Client App`, é conectado ao gerenciador de filas `QM2` no cluster `CL1`. Uma mensagem do `Client App` é processada pelo aplicativo, `Server App`. A mensagem é recuperada pelo `Server App` a partir da fila de clusters `Q1` no gerenciador de filas `QM3` no `CLUSTER2`. Como o cliente e os aplicativos do servidor não estão no mesmo cluster, a mensagem é transferida pelo gerenciador de filas de gateway `QM1`.

O modo normal de configurar um gateway do cluster é tornar o gerenciador de filas de gateway um membro de todos os clusters. No gerenciador de filas de gateway estão definidas as filas de alias em cluster para filas de clusters em todos os clusters. Os aliases de filas em cluster estão disponíveis em todos os clusters. Mensagens colocadas nos aliases de filas do cluster são roteadas através do gerenciador de filas de gateway para seus destinos corretos. O gerenciador de filas de gateway coloca mensagens enviadas nas filas de alias em cluster sobre o `SYSTEM.CLUSTER.TRANSMIT.QUEUE` comum no `QM1`.

A arquitetura de roda e raio requer que todas as mensagens entre clusters passem pelo gerenciador de filas de gateway. O resultado é que todas as mensagens fluem por meio da fila única de transmissão do cluster no `QM1`, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.

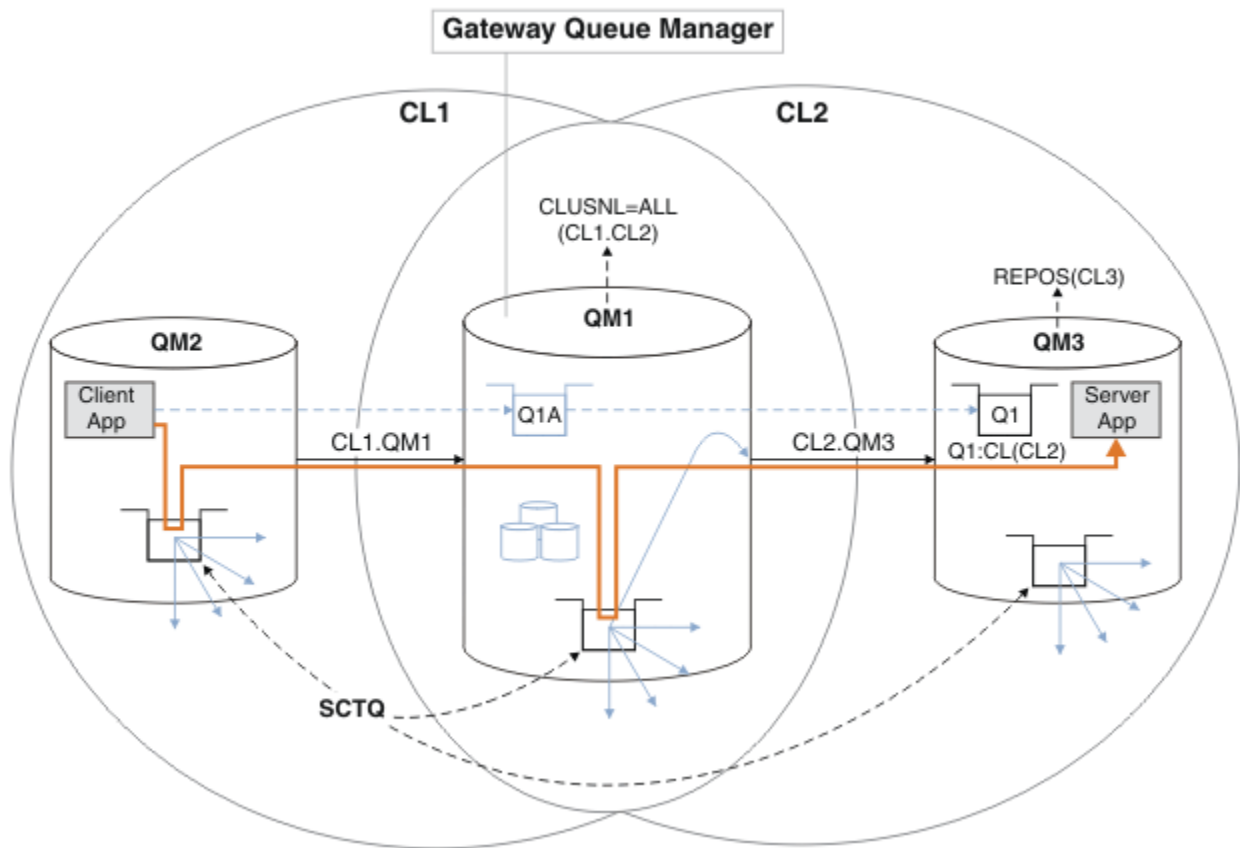
De uma perspectiva de desempenho, uma fila única não é um problema. Uma fila de transmissão comum geralmente não representa um gargalo de desempenho. O rendimento do processamento da mensagem no gateway é, em grande parte, determinado pelo desempenho dos canais que se conectam a ele. O rendimento geralmente não é afetado pelo número de filas ou pelo número de mensagens nas filas que usam os canais.

De algumas outras perspectivas, usar uma única fila de transmissão para vários aplicativos tem desvantagens:

- Não é possível isolar o fluxo de mensagens para um destino do fluxo de mensagens para outro destino. Não é possível separar o armazenamento de mensagens antes de elas serem encaminhadas, mesmo que os destinos estejam em clusters diferentes em gerenciadores de filas diferentes.

Se um cluster de destino ficar indisponível, as mensagens para esse destino acumulam-se na única fila de transmissão e acabam por preenchê-la. Quando a fila de transmissão está cheia, ela impede que as mensagens sejam colocadas na fila de transmissão para qualquer destino de cluster.

- Não é fácil monitorar a transferência de mensagens para destinos de cluster diferentes. Todas as mensagens estão na única fila de transmissão. Exibir a profundidade da fila de transmissão dá pouca indicação quanto a se as mensagens estão sendo transferidas para todos os destinos.



Nota: As setas em [Figura 12 na página 50](#) e seguindo as figuras são de tipos diferentes. Setas sólidas representam fluxos de mensagens. Os rótulos em setas sólidas são os nomes de canal de mensagens. As setas sólidas cinzentas são fluxos de mensagens potenciais do `SYSTEM.CLUSTER.TRANSMIT.QUEUE` em canais de emissor de clusters. Linhas tracejadas pretas conectam rótulos aos seus destinos. As setas tracejadas cinza são referências; por exemplo, de uma chamada `MQOPEN` feita por `Client App` para a definição de fila do alias do cluster `Q1A`.

Figura 12. Aplicativo cliente-servidor implementado na arquitetura hub e spoke usando os clusters IBM MQ

No [Figura 12 na página 50](#), os clientes do `Server App` abrem a fila `Q1A`. As mensagens são colocadas em `SYSTEM.CLUSTER.TRANSMIT.QUEUE` no `QM2`, transferidas para `SYSTEM.CLUSTER.TRANSMIT.QUEUE` no `QM1` e, em seguida, transferidas para `Q1` no `QM3`, onde são recebidas pelo aplicativo `Server App`.

A mensagem do `Client App` passa por filas de transmissão de cluster do sistema no `QM2` e no `QM1`. No [Figura 12 na página 50](#), o objetivo é isolar o fluxo de mensagens no gerenciador de filas de gateway do aplicativo cliente, de modo que suas mensagens não sejam armazenadas no `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. É possível isolar os fluxos de mensagens em qualquer um dos outros gerenciadores de filas em cluster. Também é possível isolar os fluxos de mensagens na outra direção, de volta para o cliente. Para manter as descrições das soluções resumida, as descrições consideram apenas um único fluxo do aplicativo cliente.

Soluções para isolar o tráfego de mensagens do cluster em um gerenciador de filas de gateway do cluster

Uma maneira de resolver o problema é usar os aliases do gerenciador de filas, ou definições de fila remota, para criar uma ponte entre os clusters. Crie uma definição de fila remota em cluster, uma fila de transmissão e um canal para separar cada fluxo de mensagens no gerenciador de filas de gateway; veja [Incluindo uma definição de fila remota para isolar as mensagens enviadas de um gerenciador de filas de gateway](#).

No IBM WebSphere MQ 7.5, os gerenciadores de filas do cluster não estão limitados a uma única fila de transmissão do cluster. Há duas opções:

1. Definir filas de transmissão do cluster adicionais manualmente e definir quais canais do emissor de mensagens transferem mensagens de cada fila de transmissão; veja [Incluindo uma fila de transmissão do cluster para isolar o tráfego de mensagens enviadas de um gerenciador de filas de gateway](#).
2. Permita que o gerenciador de filas crie e gerencie filas de transmissão do cluster adicionais automaticamente. Ele define uma fila de transmissão do cluster diferente para cada canal do emissor de clusters; veja [Alterando o padrão para separar filas de transmissão do cluster para isolar o tráfego de mensagem](#).

É possível combinar filas de transmissão de cluster definidas manualmente para alguns canais do emissor de clusters, deixando o gerenciador de filas administrar o restante. A combinação de filas de transmissão é a abordagem adotada em [Incluindo uma fila de transmissão do cluster para isolar o tráfego de mensagens enviadas de um gerenciador de filas de gateway](#). Nessa solução, a maioria das mensagens entre clusters usam o SYSTEM.CLUSTER.TRANSMIT.QUEUE comum. Um aplicativo é crítico, e todos os seus fluxos de mensagens são isolados de outros fluxos usando uma fila de transmissão do cluster definida manualmente.

A configuração em [Incluindo uma fila de transmissão do cluster para isolar o tráfego de mensagens do cluster enviadas de um gerenciador de filas de gateway](#) é limitada. Ela não separa o tráfego de mensagens que vai para uma fila de clusters no mesmo gerenciador de filas no mesmo cluster que outra fila de clusters. É possível separar o tráfego de mensagens para filas individuais usando as definições de fila remota que fazem parte do enfileiramento distribuído. Com clusters, usando diversas filas de transmissão do cluster, é possível separar o tráfego de mensagens que vai para diferentes canais do emissor de clusters. Várias filas de cluster no mesmo cluster, no mesmo gerenciador de filas, compartilhar um canal do emissor de clusters. As mensagens para as filas são armazenados na mesma fila de transmissão antes de serem encaminhadas do gerenciador de filas de gateway. Na configuração em [Incluindo um cluster e uma fila de transmissão do cluster para isolar o tráfego de mensagens do cluster enviadas de um gerenciador de filas de gateway](#), a limitação é evitada incluindo outro cluster e tornando o gerenciador de filas e a fila de clusters um membro do novo cluster. O novo gerenciador de filas pode ser o único gerenciador de filas no cluster. Seria possível incluir mais gerenciadores de filas para o cluster e usar o mesmo cluster para isolar as filas do cluster nesses gerenciadores de filas também.

Conceitos relacionados

[“Controle de acesso e diversas filas de transmissão do cluster” na página 29](#)

Escolha entre três modos de verificação quando um aplicativo coloca mensagens nas filas do cluster remoto. Os modos estão sendo verificados remotamente em relação a fila de clusters, verificando localmente em relação a SYSTEM.CLUSTER.TRANSMIT.QUEUE ou verificando em relação a perfis locais para a fila de clusters ou gerenciador de filas de clusters.

[Trabalhando com filas de transmissão do cluster e canais do emissor de clusters](#)

[“Sobrepondo Clusters” na página 36](#)

Sobrepor clusters fornece capacidades administrativas adicionais. Use as listas de nomes para reduzir o número de comandos necessários para administrar clusters de sobreposição.

Tarefas relacionadas

[Autorizando a Colocação de Mensagens em Filas de Cluster Remotas](#)

[Incluindo uma definição de fila remota para isolar mensagens enviadas de um gerenciador de filas de gateway](#)

[Incluindo uma fila de transmissão do cluster para isolar o tráfego de mensagens do cluster enviadas de um gerenciador de filas de gateway](#)

[Incluindo um cluster e uma fila de transmissão do cluster para isolar o tráfego de mensagens do cluster enviadas de um gerenciador de filas de gateway](#)

[Mudando o padrão para separar filas de transmissão de cluster para isolar o tráfego de mensagens](#)

[Criando dois clusters de sobreposição com um gerenciador de filas de gateway](#)

[Configurando caminhos de mensagem entre clusters](#)

[Assegurando](#)

Referências relacionadas

[setmqaut](#)

Armazenamento em Cluster: Planejando Como configurar Filas de Transmissão do Cluster

Você é guiado pelas opções de filas de transmissão do cluster. É possível configurar uma fila padrão comum, filas padrão separadas ou filas definidas manualmente.

Antes de começar

Revise [“Como escolher o tipo de fila de transmissão do cluster a usar”](#) na página 55.

Sobre esta tarefa

Você tem algumas escolhas a fazer ao planejar como configurar um gerenciador de filas para selecionar uma fila de transmissão do cluster.

1. Qual é a fila de transmissão do cluster padrão para transferências de mensagens do cluster?
 - a. Uma fila de transmissão de cluster comum, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.
 - b. Filas de transmissão do cluster separadas. O gerenciador de filas gerencia as filas de transmissão do cluster separadas. Ele os cria como filas permanentes e dinâmicas a partir da fila de modelos, `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE`. Ele cria uma fila de transmissão do cluster para cada canal do emissor de clusters que ele usa.
2. Para as filas de transmissão do cluster que você decidir criar manualmente, você tem duas opções adicionais:
 - a. Definir uma fila de transmissão separada para cada canal do emissor de clusters que você decidir configurar manualmente. Neste caso, configure o atributo de fila **CLCHNAME** da fila de transmissão para o nome de um canal do emissor de clusters. Selecione o canal do emissor de clusters para transferir mensagens a partir desta fila de transmissão.
 - b. Combinar tráfego de mensagens para um grupo de canais do emissor de clusters na mesma fila de transmissão do cluster; consulte [Figura 13 na página 53](#). Nesse caso, configure o atributo de fila **CLCHNAME** de cada fila de transmissão comum para um nome de canal do emissor de clusters genérico. Um nome de canal do emissor de clusters genérico é um filtro para agrupar nomes de canais do emissor de clusters. Por exemplo, o `SALES.*` agrupa todos os canais do emissor de clusters que possuem nomes iniciados com `SALES.` É possível colocar diversos caracteres curingas em qualquer parte da sequência de filtragem. O caractere curinga é um asterisco, `"*`". Ele representa desde nenhuma a qualquer número de caracteres.

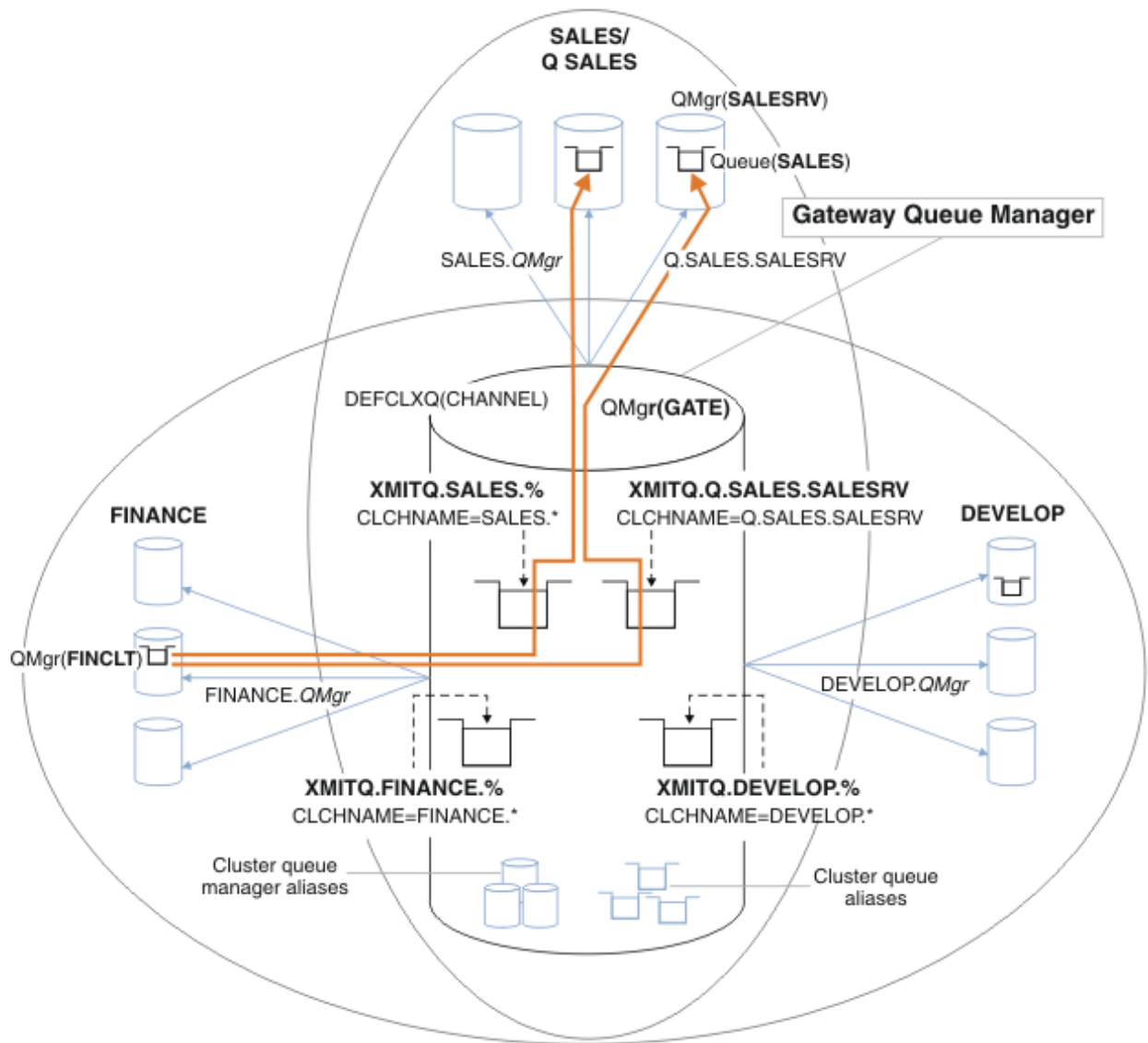


Figura 13. Exemplo de filas de transmissão específicas para diferentes clusters departamentais do IBM MQ

Procedimento

1. Selecione o tipo de fila de transmissão do cluster padrão para uso.

- Escolha uma única fila de transmissão do cluster, ou filas separadas para cada conexão do cluster.

Deixe a configuração padrão ou execute o comando **MQSC**:

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

2. Isole quaisquer fluxos de mensagens que não devem compartilhar uma fila de transmissão de cluster com outros fluxos.

- Consulte o “Armazenamento em Cluster: Exemplo de Configuração de Diversas Filas de Transmissão do Cluster” na página 56. No exemplo, a fila SALES, que deve ser isolada, é um membro do cluster SALES, no SALESRV. Para isolar a fila SALES, crie um novo cluster Q.SALES, torne o gerenciador de filas SALESRV um membro e modifique a fila SALES para pertencer a Q.SALES.
- Os gerenciadores de filas que enviam mensagens para o SALES também devem ser membros do novo cluster. Se você usar um alias de fila em cluster e um gerenciador de filas de gateway, como

no exemplo, em muitos casos é possível limitar as mudanças para tornar o gerenciador de filas de gateway um membro do novo cluster.

- Entretanto, a separação de fluxos do gateway para o destino não separa fluxos para o gateway a partir do gerenciador de filas de origem. Mas, às vezes, é suficiente separar fluxos do gateway e não fluxos para o gateway. Se não for suficiente, então inclua o gerenciador de filas de origem no novo cluster. Se você deseja que as mensagens passem pelo gateway, mova o alias do cluster para o novo cluster e continue a enviar mensagens para o alias do cluster no gateway, e não diretamente para o gerenciador de filas de destino.

Siga estas etapas para isolar fluxos de mensagens:

- a) Configure os destinos dos fluxos para que cada fila de destino seja a única fila em um cluster específico, nesse gerenciador de filas.
 - b) Crie os canais de emissor de clusters e receptor de clusters para qualquer novo cluster que você criar seguindo uma convenção de nomenclatura sistemática.
 - Consulte o [“Armazenamento em cluster: considerações especiais para clusters sobrepostos”](#) na página 44.
 - c) Defina uma fila de transmissão do cluster para cada destino isolado em cada gerenciador de filas que envia mensagens para a fila de destino.
 - Uma convenção de nomenclatura para filas de transmissão de cluster é usar o valor do atributo de nome do canal de cluster, CLCHNAME, com o prefixo XMITQ.
3. Crie filas de transmissão do cluster para atender aos requisitos de controle ou monitoramento.
- Requisitos típicos de controle e monitoramento resultam em uma fila de transmissão por cluster ou em uma fila de transmissão por gerenciador de filas. Se você seguir a convenção de nomenclatura para canais de cluster, *ClusterName.QueueManagerName*, será fácil criar nomes de canais genéricos que selecionam um cluster de gerenciadores de filas ou todos os clusters de um gerenciador de filas são um membro de; consulte [“Armazenamento em Cluster: Exemplo de Configuração de Diversas Filas de Transmissão do Cluster”](#) na página 56.
 - Estenda a convenção de nomenclatura para filas de transmissão do cluster para atender nomes de canais genéricos, substituindo o símbolo de asterisco por um sinal de porcentagem. Por exemplo,

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

Conceitos relacionados

[Trabalhando com filas de transmissão do cluster e canais do emissor de clusters](#)

[“Controle de acesso e diversas filas de transmissão do cluster”](#) na página 29

Escolha entre três modos de verificação quando um aplicativo coloca mensagens nas filas do cluster remoto. Os modos estão sendo verificados remotamente em relação a fila de clusters, verificando localmente em relação a SYSTEM.CLUSTER.TRANSMIT.QUEUE ou verificando em relação a perfis locais para a fila de clusters ou gerenciador de filas de clusters.

[“Sobrepondo Clusters”](#) na página 36

Sobrepor clusters fornece capacidades administrativas adicionais. Use as listas de nomes para reduzir o número de comandos necessários para administrar clusters de sobreposição.

Tarefas relacionadas

[Incluindo uma definição de fila remota para isolar mensagens enviadas de um gerenciador de filas de gateway](#)

[Incluindo uma fila de transmissão do cluster para isolar o tráfego de mensagens do cluster enviadas de um gerenciador de filas de gateway](#)

[Incluindo um cluster e uma fila de transmissão do cluster para isolar o tráfego de mensagens do cluster enviadas de um gerenciador de filas de gateway](#)

[Mudando o padrão para separar filas de transmissão de cluster para isolar o tráfego de mensagens](#)

[Criando dois clusters de sobreposição com um gerenciador de filas de gateway](#)

[Configurando caminhos de mensagem entre clusters](#)

Como escolher o tipo de fila de transmissão do cluster a usar

Como escolher entre diferentes opções de configuração de fila de transmissão do cluster.

É possível escolher qual fila de transmissão do cluster está associada a um canal do emissor de clusters

1. É possível ter todos os canais do emissor de clusters associados à fila de transmissão do cluster padrão único, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`; essa opção é o padrão
2. É possível configurar todos os canais do emissor de clusters para serem automaticamente associados a uma fila de transmissão do cluster separada. As filas são criadas pelo gerenciador de filas a partir da fila de modelos `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` e nomeadas `SYSTEM.CLUSTER.TRANSMIT.ChannelName`. Os canais usarão sua fila de transmissão de cluster de nome exclusivo se o atributo gerenciador de filas do **DEFCLXQ** for configurado como `CHANNEL`.
3. É possível configurar canais do emissor de clusters específicos a serem atendidos por uma única fila de transmissão do cluster. Selecione esta opção criando uma fila de transmissão e configurando o seu atributo **CLCHNAME** para o nome do canal do emissor de clusters.
4. Você pode selecionar grupos de canais do emissor de clusters a serem atendidos por uma única fila de transmissão do cluster. Selecione esta opção criando uma fila de transmissão e configurando seu atributo **CLCHNAME** para um nome de canal genérico, como `ClusterName.*`. Se você nomear os canais de cluster seguindo as convenções de nomenclatura em [“Armazenamento em cluster: considerações especiais para clusters sobrepostos”](#) na página 44, esse nome selecionará todos os canais de cluster conectados aos gerenciadores de filas no cluster `ClusterName`.

É possível combinar qualquer uma das opções de fila de transmissão do cluster padrão para alguns canais do emissor de clusters com qualquer número de configurações de fila de transmissão do cluster específicas e genéricas.

Melhores Práticas

Na maioria dos casos, para instalações existentes do IBM MQ, a configuração padrão é a melhor opção. Um gerenciador de filas de clusters armazena mensagens de cluster em uma única fila de transmissão de cluster, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. Você tem a opção de alterar o padrão para o armazenar mensagens para diferentes gerenciadores de fila e diferentes clusters em filas de transmissão separadas, ou de definir suas próprias filas de transmissão.

Na maioria dos casos, para novas instalações do IBM MQ, a configuração padrão também é a melhor opção. O processo de passar de uma configuração padrão para o padrão alternativo de ter uma fila de transmissão para cada canal do emissor de clusters é automático. A reversão também é automática. A escolha de uma ou outra não é crítica, é possível revertê-la.

A razão para escolher uma configuração diferente está mais relacionada a controle e gerenciamento do que a funcionalidade ou desempenho. Com algumas exceções, configurar diversas filas de transmissão do cluster não beneficia o comportamento do gerenciador de filas. Isso resulta em mais filas e requer que você modifique os procedimentos de monitoramento e gerenciamento já configurados que fazem referência à fila de transmissão única. É por isso que, em geral, permanecer com a configuração padrão é a melhor opção, a menos que você tenha fortes razões de controle ou gerenciamento para uma opção diferente.

As exceções estão ambas preocupadas com o que acontece se aumentar o número de mensagens armazenadas no `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. Se você tomar todas as providências para separar as mensagens para um destino das mensagens para outro destino, problemas de canal e entrega com um destino não deverão afetar a entrega para outro destino. No entanto, o número de mensagens armazenadas no `SYSTEM.CLUSTER.TRANSMIT.QUEUE` pode aumentar devido à não entrega de mensagens com rapidez suficiente para um destino. O número de mensagens no `SYSTEM.CLUSTER.TRANSMIT.QUEUE` para um destino pode afetar a entrega de mensagens para outros destinos.

Para evitar problemas que resultam do preenchimento de uma única fila de transmissão, busque construir capacidade suficiente em sua configuração. Então, se um destino falhar e uma lista não processada de mensagens começar a se acumular, você terá tempo para corrigir o problema.

Se as mensagens forem encaminhadas por meio de um gerenciador de filas de hub, como um gateway de cluster, eles compartilham uma fila de transmissão comum, SYSTEM . CLUSTER . TRANSMIT . QUEUE. Se o número de mensagens armazenadas no SYSTEM . CLUSTER . TRANSMIT . QUEUE no gerenciador de filas de gateway atingir a sua profundidade máxima, o gerenciador de filas começa a rejeitar novas mensagens para a fila de transmissão até que sua profundidade seja reduzida. O congestionamento afeta as mensagens para todos os destinos que são roteados através do gateway. As backup fazem backup das filas de transmissão de outros gerenciadores de filas que estão enviando mensagens ao gateway. O problema se manifesta pela gravação de mensagens em logs de erro do gerenciador de filas, pela queda de rendimento da mensagem e por mais tempo decorrido entre o envio de uma mensagem e o momento em que uma mensagem chega ao seu destino.

O efeito de congestionamento em uma única fila de transmissão pode se tornar aparente mesmo antes de ela estar cheia. Se você tiver um tráfego de mensagens misto, com algumas mensagens grandes não persistentes e algumas mensagens pequenas, o tempo para entregar mensagens pequenas aumenta conforme a fila de transmissão enche. O atraso é devido à gravação de grandes mensagens não persistentes em disco que normalmente não seriam gravadas no disco. Se houver fluxos de mensagens com tempo crítico, compartilhando uma fila de transmissão do cluster com outros fluxos de mensagens mistos, pode ser útil configurar um caminho de mensagem especial para isolá-lo de outros fluxos de mensagem; veja Incluindo um cluster e uma fila de transmissão do cluster para isolar o tráfego de mensagens do cluster enviadas de um gerenciador de filas de gateway.

As outras razões para configurar filas de transmissão do cluster separadas são atender aos requisitos de governança ou simplificar o monitoramento de mensagens enviadas para destinos de cluster diferentes. Por exemplo, você pode ser preciso demonstrar que as mensagens para um destino nunca compartilham uma fila de transmissão com mensagens para outro destino.

Altere o atributo gerenciador de filas **DEFCLXQ** que controla a fila de transmissão do cluster padrão, para criar diferentes filas de transmissão de cluster para cada canal de emissor de clusters. Vários destinos podem compartilhar um canal do emissor de clusters, portanto, você deve planejar seus clusters para que atendam totalmente a esse objetivo. Aplique o método Incluindo um cluster e uma fila de transmissão do cluster para isolar o tráfego de mensagens do cluster enviadas de um gerenciador de filas de gateway sistematicamente a todas as suas filas de clusters. O resultado que você está buscando é que nenhum destino do cluster compartilhe um canal do emissor de clusters com outro destino do cluster. Como consequência, nenhuma mensagem para um destino do cluster compartilha sua fila de transmissão de cluster com uma mensagem para outro destino.

Criar uma fila de transmissão de clusters separada para um fluxo de mensagens específico torna fácil monitorar o fluxo de mensagens para esse destino. Para usar uma nova fila de transmissão do cluster, defina a fila, associe-a a um canal do emissor de clusters e parar e inicie o canal. A mudança não precisa ser permanente. É possível isolar o fluxo de mensagens por um tempo para monitorar a fila de transmissão e então revertê-lo para usar a fila de transmissão padrão novamente.

Tarefas relacionadas

Armazenamento em Cluster: Exemplo de Configuração de Diversas Filas de Transmissão do Cluster
Nesta tarefa você aplica as etapas para planejar diversas filas de transmissão do cluster para três clusters de sobreposição. Os requisitos são separar fluxos de mensagens para uma fila de clusters, de todos os outros fluxos de mensagens, e armazenar mensagens para clusters diferentes em diferente filas de transmissão do cluster.

Armazenamento em cluster: alternando filas de transmissão do cluster

Planeje como as mudanças para as filas de transmissão do cluster de um gerenciador de filas de produção existente serão colocadas em vigor.

Armazenamento em Cluster: Exemplo de Configuração de Diversas Filas de Transmissão do Cluster

Nesta tarefa você aplica as etapas para planejar diversas filas de transmissão do cluster para três clusters de sobreposição. Os requisitos são separar fluxos de mensagens para uma fila de clusters, de todos os outros fluxos de mensagens, e armazenar mensagens para clusters diferentes em diferente filas de transmissão do cluster.

Sobre esta tarefa

As etapas nesta tarefa mostram como aplicar o procedimento em “Armazenamento em Cluster: Planejando Como configurar Filas de Transmissão do Cluster” na página 52 e alcançam a configuração mostrada em [Figura 14](#) na página 57. É um exemplo de três clusters de sobreposição, com um gerenciador de filas de gateway, que é configurado com filas de transmissão do cluster separadas. Os comandos MQSC para definir os clusters são descritos em “Criando os Clusters de Exemplo” na página 59.

Para o exemplo, há dois requisitos. Um é separar o fluxo de mensagens do gerenciador de filas de gateway para o aplicativo de vendas que registra as vendas. O segundo é consultar quantas mensagens estão aguardando para serem enviadas a diferentes áreas departamentais em qualquer ponto no tempo. Os clusters SALES, FINANCE e DEVELOP já estão definidos. As mensagens de cluster são atualmente encaminhadas do SYSTEM.CLUSTER.TRANSMIT.QUEUE.

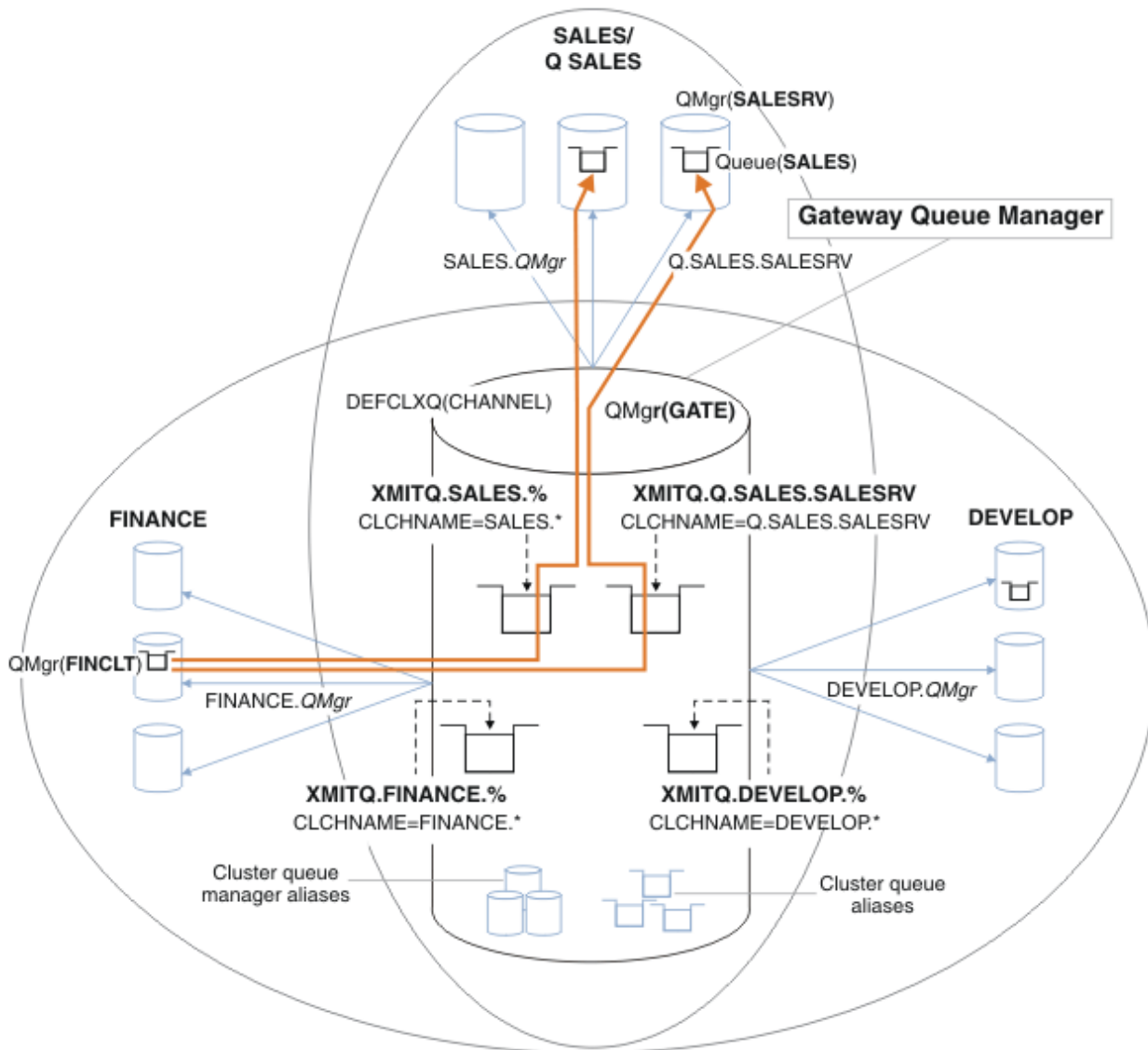


Figura 14. Exemplo de filas de transmissão específicas para diferentes clusters departamentais do IBM MQ

As etapas para modificar os clusters são as seguintes: Para as definições, consulte [Mudanças para isolar a fila de vendas em um cluster novo e separar as filas de transmissão do cluster de gateway..](#)

Procedimento

1. A primeira etapa de configuração é "Selecione o tipo de fila de transmissão do cluster padrão para uso".

A decisão é criar filas de transmissão do cluster padrão separadas executando o comando **MQSC** a seguir no gerenciador de filas GATE.

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

Não há nenhuma razão restrita para a escolha deste padrão, pois a intenção é definir manualmente as filas de transmissão do cluster. A opção possui um valor de diagnóstico simples. Se uma definição manual for realizada de forma restrita, e uma mensagem fluir por uma fila de transmissão do cluster padrão, ela será mostrada na criação de uma fila de transmissão do cluster dinâmica permanente.

2. A segunda etapa de configuração é "Isole quaisquer fluxos de mensagens que não devem compartilhar uma fila de transmissão de cluster com outros fluxos".

Nesse caso o aplicativo de vendas que recebe mensagens da fila SALES em SALESRV requer isolamento. Apenas o isolamento de mensagens do gerenciador de filas de gateway é necessário. As três subetapas são:

- a) "Configure os destinos dos fluxos para que cada fila de destino seja a única fila em um cluster específico, nesse gerenciador de filas".

O exemplo requer incluir o gerenciador de filas SALESRV em um novo cluster do departamento de vendas. Se você tiver poucas filas que requerem isolamento, você pode decidir sobre a criação de um cluster específico para a fila SALES. Uma possível convenção de nomenclatura para o nome do cluster é nomear tais clusters, *Q.QueueName*, por exemplo *Q.SALES*. Uma abordagem alternativa, que pode ser mais prática se você tiver um grande número de filas a serem isoladas, é criar clusters de filas isoladas onde e quando necessário. Os nomes de clusters podem ser *QUEUES.n*.

No exemplo, o novo cluster é chamado *Q.SALES*. Para incluir o novo cluster, consulte as definições em Mudança para isolar a fila de vendas em um novo cluster e separar as fila de transmissão do cluster de gateway O resumo das mudanças de definições é o seguinte:

- i) Inclua o *Q.SALES* na lista de nomes de clusters nos gerenciadores de filas do repositório. A lista de nomes é indicada no parâmetro do gerenciador de filas **REPOSNL**.
- ii) Inclua o *Q.SALES* na lista de nomes de clusters no gerenciador de filas de gateway. A lista de nomes é referida em todas as definições de alias de filas de clusters e alias de gerenciadores de filas do cluster no gerenciador de filas de gateway.
- iii) Crie uma lista de nomes no gerenciador de filas SALESRV, o qual é membro de ambos os clusters, e altere a adesão do cluster à fila SALES:

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

A fila SALES é um membro de ambos os clusters, apenas para a transição. Uma vez que a nova configuração está em execução, remova a fila SALES do cluster SALES. Consulte Figura 15 na página 62.

- b) "Crie os canais de emissor de clusters e receptor de clusters para qualquer novo cluster que você criar seguindo uma convenção de nomenclatura sistemática".
 - i) Inclua o canal cluster-receptor *Q.SALES.RepositoryQMGr* a cada um dos gerenciadores de filas do repositório
 - ii) Inclua o canal do emissor de clusters *Q.SALES.OtherRepositoryQMGr* em cada um dos gerenciadores de filas do repositório, para conectar ao outro gerenciador de repositório. Inicie esses canais.
 - iii) Inclua os canais do receptor de clusters *Q.SALES.SALESRV* e o *Q.SALES.GATE* em qualquer um dos gerenciadores de filas do repositório que está em execução.

- iv) Inclua os canais do emissor de clusters Q . SALES . SALESRV e Q . SALES . GATE nos gerenciadores de filas SALESRV e GATE. Conecte o canal do emissor de clusters ao gerenciador de filas do repositório no qual você criou os gerenciadores de receptores de cluster.
- c) "Defina uma fila de transmissão do cluster para cada destino isolado em cada gerenciador de filas que enviarmensagens para a fila de destino".

No gerenciador de filas de gateway defina a fila de transmissão do cluster XMITQ . Q . SALES . SALESRV para o canal do emissor de clusters Q . SALES . SALESRV:

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

3. A terceira etapa de configuração é "Crie filas de transmissão do cluster para atender aos requisitos de controle ou monitoramento".

No gerenciador de filas de gateway defina as filas de transmissão do cluster:

```
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
```

Como proceder a seguir

Altere para a nova configuração no gerenciador de filas de gateway.

A alternância é acionada iniciando os novos canais e reiniciando os canais que agora estão associados a diferentes filas de transmissão. Como alternativa, é possível parar e iniciar o gerenciador de filas de gateway.

1. Pare os canais a seguir no gerenciador de filas de gateway:

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

2. Inicie os canais a seguir no gerenciador de filas de gateway:

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

Quando o comutador estiver completo, remova a fila SALES do cluster SALES, consulte [Figura 15 na página 62](#).

Conceitos relacionados

[Como escolher o tipo de fila de transmissão do cluster a usar](#)

[Como escolher entre diferentes opções de configuração de fila de transmissão do cluster.](#)

Tarefas relacionadas

[Armazenamento em cluster: alternando filas de transmissão do cluster](#)

Planeje como as mudanças para as filas de transmissão do cluster de um gerenciador de filas de produção existente serão colocadas em vigor.

Criando os Clusters de Exemplo

As definições e instruções para criar o cluster de exemplo e modificá-lo para isolar a fila SALES e as mensagens separadas no gerenciador de filas de gateway.

Sobre esta tarefa

Os comandos completos do **MQSC** para criar os clusters FINANCE, SALES e Q . SALES são fornecidos em [Definições para os clusters básicos, Mudanças para isolar a fila de vendas em um novo cluster e separar](#)

as filas de transmissão do cluster de gateway Remove a fila de vendas no gerenciador de filas SALESRV do cluster de vendas O cluster DEVELOP é omitido das definições para manter as definições mais curtas.

Procedimento

1. Crie os clusters SALES e FINANCE e o gerenciador de filas de gateway.

a) Crie os gerenciadores de filas.

Execute o comando: `crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE QmgrName` para cada um dos nomes do gerenciador de filas no [Tabela 4 na página 60](#).

Descrição	Nome do gerenciador de filas	Número da Porta
Repositório de finanças	FINR1	1414
Repositório de finanças	FINR2	1415
Cliente de finanças	FINCLT	1418
Repositório de vendas	SALER1	1416
Repositório de vendas	SALER2	1417
Servidor de vendas	SALESRV	1419
Gateway	GATE	1420

b) Inicie todos os gerenciadores de filas

Execute o comando: `strmqm QmgrName` para cada um dos nomes do gerenciador de filas no [Tabela 4 na página 60](#).

c) Crie as definições para cada um dos gerenciadores de filas

Execute o comando: `runmqsc QmgrName < filename` em que os arquivos são listados em [Definições para os clusters básicos](#) e o nome do arquivo corresponde ao nome do gerenciador de filas.

Definições para Clusters Básicos

finr1.txt

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

finr2.txt

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

finclt.txt

```
DEFINE LISTENER(1418) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1418) REPLACE
```

```
START LISTENER(1418)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINCLT) CHLTYPE(CLUSRCVR) CONNAME('localhost(1418)')
CLUSTER(FINANCE) REPLACE
DEFINE QMODEL(SYSTEM.SAMPLE.REPLY) REPLACE
```

saler1.txt

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

saler2.txt

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

salesrv.txt

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

gate.txt

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

2. Teste a configuração executando o programa de solicitação de amostra.

a) Inicie o programa de monitor acionador no gerenciador de filas SALESRV

No Windows, abra uma janela de comando e execute o comando `runmqtrm -m SALESRV`

b) Execute o programa de solicitação de amostra e envie uma solicitação.

No Windows, abra uma janela de comando e execute o comando `amqsreq A.SALES FINCLT`

A mensagem de solicitação é ecoada de volta e, após 15 segundos, o programa de amostra é concluído.

3. Crie as definições para isolar a fila SALES no cluster Q.SALES e mensagens de cluster separadas para o cluster SALES e FINANCE no gerenciador de filas de gateway.

Execute o comando: `runmqsc QmgrName < filename` em que os arquivos são listados na lista a seguir e o nome do arquivo quase corresponde ao nome do gerenciador de filas.

Mudanças para Isolar a Fila de Vendas em um Novo Cluster e Separar as Filas de Transmissão do Cluster do Gateway **chgsaler1.txt**

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

chgsaler2.txt

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

chgsalesrv.txt

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

chgate.txt

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

4. Remova a fila SALES do cluster SALES.

Execute o comando **MQSC** em [Figura 15](#) na página 62:

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

Figura 15. Remover a Fila de Vendas no Gerenciador de Filas SALESRV do Cluster de Vendas

5. Alterne os canais para as novas filas de transmissão.

O requisito é parar e iniciar todos os canais que o gerenciador de filas GATE está usando. Para fazer isso com o menor número de comandos, pare e inicie o gerenciador de filas

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

Como proceder a seguir

1. Execute novamente o programa de solicitação de amostra para verificar se a nova configuração funciona; consulte a etapa “2” na [página 61](#)

2. Monitore as mensagens que fluem através de todas as filas de transmissão do cluster no gerenciador de filas GATE:

a. Altere a definição de cada uma das filas de transmissão do cluster para ativar o monitoramento de filas.

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

b. Verifique se o monitoramento de estatísticas do gerenciador de filas é OFF para minimizar a saída e configurar o intervalo de monitoramento a um valor mais baixo para realizar vários testes convenientemente.

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

c. Reinicie o gerenciador de filas GATE.

d. Execute o programa de solicitação de amostra algumas vezes para verificar se o mesmo número de mensagens está fluindo por meio do SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV e do SYSTEM.CLUSTER.TRANSMIT.QUEUE. As solicitações fluem por meio do SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV e respondem por meio do SYSTEM.CLUSTER.TRANSMIT.QUEUE.

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

e. Os resultados durante dois intervalos são os seguintes:

```
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.

```

Uma solicitação e uma mensagem de resposta foram enviadas no primeiro intervalo e duas no segundo. É possível supor que as mensagens de solicitação foram colocadas no SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV e as mensagens de resposta no SYSTEM.CLUSTER.TRANSMIT.QUEUE.

Armazenamento em cluster: alternando filas de transmissão do cluster

Planeje como as mudanças para as filas de transmissão do cluster de um gerenciador de filas de produção existente serão colocadas em vigor.

Antes de começar

Se você reduzir o número de mensagens que o processo de comutação precisa transferir para a nova fila de transmissão, a comutação é concluída mais rapidamente. Leia [Como o processo de comutar o canal do emissor de clusters para uma fila de transmissão diferente funciona](#) para as razões para tentar esvaziar a fila de transmissão antes de prosseguir.

Sobre esta tarefa

Existem duas maneiras de fazer as mudanças a filas de transmissão do cluster entrarem em vigor.

1. Deixar o gerenciador de filas fazer as mudanças automaticamente. Esse é o padrão. O gerenciador de filas comuta canais do emissor de clusters com mudanças na fila de transmissão pendentes quando um canal do emissor de clusters é iniciado na próxima vez.
2. Fazer as mudanças manualmente. É possível fazer as mudanças em um canal do emissor de clusters quando ele está parado. É possível comutá-lo de uma fila de transmissão do cluster para outra antes que o canal do emissor de clusters seja iniciado.

Quais fatores que você leva em conta ao decidir qual das duas opções escolher e como você gerencia o computador?

Procedimento

- Opção 1: deixar o gerenciador de filas fazer as mudanças automaticamente; veja [“Comutando canais do emissor de clusters ativos para outro conjunto de filas de transmissão de cluster”](#) na página 66.

Escolha esta opção se desejar que o gerenciador de filas faça a comutação para você.

Uma maneira alternativa de descrever essa opção é dizer que o gerenciador de filas comuta um canal do emissor de clusters sem você forçar o canal a parar. Você tem a opção de forçar o canal a parar e, em seguida, iniciar o canal, para fazer a comutação ocorrer mais cedo. O comutador inicia quando o canal é iniciado e é executado enquanto o canal está em execução, o que é diferente da opção 2. Na opção 2, o comutador ocorre quando o canal está parado.

Se você escolher essa opção, permitindo que a comutação aconteça automaticamente, o processo de comutação inicia quando um canal do emissor de clusters é iniciado. Se o canal não estiver parado, ele começa depois de o canal se tornar inativo, se houver uma mensagem para processar. Se o canal for interrompido, inicie-o com o comando `START CHANNEL`.

O processo de comutação é concluído assim que não há mais mensagens para o canal do emissor de clusters na fila de transmissão que o canal estava atendendo. Assim que isso acontece, mensagens recém-chegadas para o canal do emissor de clusters são armazenadas diretamente na nova fila de transmissão. Até então, as mensagens são armazenadas na fila de transmissão antiga e o processo de comutação transfere mensagens da fila de transmissão antiga para a nova fila de transmissão. O canal do emissor de clusters encaminha as mensagens da nova fila de transmissão de cluster durante todo o processo de comutação.

Quando o processo de comutação termina depende do estado do sistema. Se você estiver fazendo as mudanças em uma janela de manutenção, avalie antecipadamente se o processo de comutação será concluído a tempo. A sua conclusão a tempo depende de o número de mensagens que estão aguardando a transferência da fila de transmissão antiga atingir zero.

A vantagem do primeiro método é que ele é automático. Uma desvantagem é que, se o tempo para fazer as mudanças de configuração for limitado a uma janela de manutenção, você precisa ter certeza de que pode controlar o sistema para concluir o processo de comutação dentro da janela de manutenção. Se não puder ter certeza, a opção 2 pode ser melhor.

- Opção 2: fazer as mudanças manualmente; veja [“Comutando um canal do emissor de clusters parado para outra fila de transmissão do cluster”](#) na página 67.

Escolha essa opção se desejar controlar todo o processo de alternância manualmente ou se desejar comutar um canal inativo ou parado. É uma boa escolha se você estiver comutando alguns canais do emissor de clusters e você quiser fazer a comutação durante uma janela de manutenção.

Uma descrição alternativa dessa opção é dizer que você comuta o canal do emissor de clusters enquanto o canal do emissor de clusters está parado.

Se você escolher essa opção, terá controle total sobre quando a comutação ocorre.

Você pode ter certeza quanto à conclusão do processo de comutação em um período de tempo fixo, dentro da janela de manutenção. O tempo que a comutação leva depende de quantas mensagens devem ser transferidos de uma fila de transmissão para a outra. Se as mensagens continuarem chegando, pode levar algum tempo para o processo transferir todas as mensagens. Você tem a opção de comutar o canal sem transferir mensagens da fila de transmissão antiga. A comutação é "instantânea".

Ao reiniciar o canal do emissor de clusters, ele inicia o processamento de mensagens na fila de transmissão que você recém designou a ele.

A vantagem do segundo método é que você tem controle sobre o processo de comutação. A desvantagem é que você deve identificar os canais do emissor de clusters a serem comutados, executar os comandos necessários e resolver quaisquer canais em dúvida que possam estar impedindo a parada do canal do emissor de clusters.

Conceitos relacionados

[Como escolher o tipo de fila de transmissão do cluster a usar](#)

[Como escolher entre diferentes opções de configuração de fila de transmissão do cluster.](#)

[Como Funciona o Processo de Alternar o Canal do Emissor de Clusters para uma Fila de Transmissão Diferente](#)

Tarefas relacionadas

[Armazenamento em Cluster: Exemplo de Configuração de Diversas Filas de Transmissão do Cluster](#)

Nesta tarefa você aplica as etapas para planejar diversas filas de transmissão do cluster para três clusters de sobreposição. Os requisitos são separar fluxos de mensagens para uma fila de clusters, de todos os outros fluxos de mensagens, e armazenar mensagens para clusters diferentes em diferente filas de transmissão do cluster.

Comutando canais do emissor de clusters ativos para outro conjunto de filas de transmissão de cluster

Esta tarefa fornece três opções para comutar canais do emissor de clusters ativos. Uma opção é permitir que o gerenciador de filas faça a comutação automaticamente, o que não afeta aplicativos em execução. As outras opções são parar e iniciar os canais manualmente ou reiniciar o gerenciador de filas.

Antes de começar

Mude a configuração da fila de transmissão do cluster. É possível alterar o atributo do gerenciador de filas **DEFCLXQ** ou incluir ou modificar o atributo **CLCHNAME** das filas de transmissão.

Se você reduzir o número de mensagens que o processo de comutação precisa transferir para a nova fila de transmissão, a comutação é concluída mais rapidamente. Leia [Como o processo de comutar o canal do emissor de clusters para uma fila de transmissão diferente funciona](#) para as razões para tentar esvaziar a fila de transmissão antes de prosseguir.

Sobre esta tarefa

Siga as etapas na tarefa como base para trabalhar em seu próprio plano para fazer mudanças de configuração da fila de transmissão.

Procedimento

1. Opcional: Registrar o status do canal atual

Crie um registro do status de canais atuais e salvos que estão atendendo as filas de transmissão do cluster. Os comandos a seguir exibem o status associado às filas de transmissão do cluster do sistema. Inclua seus próprios comandos para exibir o status associado às filas de transmissão que você definiu. Use uma convenção, como `XMITQ.Channe1Name`, para nomear as filas de transmissão do cluster definidas para facilitar a exibição do status do canal para essas filas de transmissão.

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

2. Comutar filas de transmissão

- Não faça nada. O gerenciador de filas comuta canais do emissor de clusters quando eles reiniciam após serem parados ou ficarem inativos.

Escolha esta opção se você não tiver regras ou preocupações quanto a alterar uma configuração do gerenciador de filas. Aplicativos em execução não são afetados pelas mudanças.

- Reiniciar o gerenciador de filas. Todos os canais do emissor de clusters são parados e reiniciados automaticamente sob demanda.

Escolha esta opção para iniciar todas as mudanças imediatamente. Os aplicativos em execução são interrompidos pelo gerenciador de filas quando ele é encerrado e reiniciado.

- Parar os canais do emissor de clusters individuais e reiniciá-los.

Escolha esta opção para mudar alguns canais imediatamente. Aplicativos em execução sofrem um pequeno atraso na transferência de mensagens entre a parada e o reinício do canal de mensagens. O canal do emissor de clusters permanece em execução, exceto durante o tempo em que você o parar. Durante o processo de comutação, as mensagens são entregues à fila de transmissão antiga, transferidas para a nova fila de transmissão pelo processo de alternância e encaminhadas da nova fila de transmissão pelo canal do emissor de clusters.

3. Opcional: Monitorar os canais conforme eles são comutados

Exiba o status do canal e a profundidade da fila de transmissão durante a comutação. O exemplo a seguir exibe o status para filas de transmissão do cluster do sistema.

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

- ### 4. Opcional: Monitore as mensagens AMQ7341 A fila de transmissão para o canal *ChannelName* mudou da fila *QueueName* para a *QueueName* que estão gravadas no log de erros do gerenciador de filas.

Comutando um canal do emissor de clusters parado para outra fila de transmissão do cluster

Se você optar por fazer mudanças manualmente, faça as mudanças em um canal do emissor de clusters quando ele for interrompido e altere-o de uma fila de transmissão do cluster para outra antes de o canal do emissor de clusters ser iniciado.

Antes de começar

É possível fazer algumas mudanças na configuração, e agora deseja torná-las efetivas sem iniciar os canais do emissor de clusters afetados. Como alternativa, você faz as mudanças na configuração necessárias como uma das etapas da tarefa.

Se você reduzir o número de mensagens que o processo de comutação precisa transferir para a nova fila de transmissão, a comutação é concluída mais rapidamente. Leia [Como o processo de comutar o canal do emissor de clusters para uma fila de transmissão diferente funciona](#) para as razões para tentar esvaziar a fila de transmissão antes de prosseguir.

Sobre esta tarefa

Esta tarefa comuta as filas de transmissão atendidas por canais do emissor de clusters parados ou inativos. É possível executar essa tarefa porque um canal do emissor de clusters está parado e você deseja comutar a fila de transmissão imediatamente. Por exemplo, por alguma razão, um canal do emissor de clusters não está iniciando ou tem algum outro problema de configuração. Para resolver o problema, você decide criar um canal do emissor de clusters, e associa a fila de transmissão para o canal do emissor de clusters antigo ao novo canal do emissor de clusters que você definiu.

Um cenário mais provável é você desejar controlar quando a reconfiguração de filas de transmissão do cluster é executada. Para controlar totalmente a reconfiguração, pare os canais, mude a configuração e, em seguida, comute as filas de transmissão.

Procedimento

1. Parar os canais que você pretende comutar

- a) Pare os canais em execução ou inativos que você pretende comutar. Parar um canal do emissor de clusters inativo impede que ele inicie enquanto você estiver fazendo mudanças na configuração.

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

2. Opcional: Faça as mudanças de configuração.

Por exemplo, consulte [“Armazenamento em Cluster: Exemplo de Configuração de Diversas Filas de Transmissão do Cluster”](#) na página 56.

3. Comute os canais do emissor de clusters para as novas filas de transmissão do cluster.

Multi No [Multiplataformas](#), emita o comando a seguir:

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

z/OS No z/OS, use a função SWITCH do comando CSQUTIL para comutar as mensagens ou monitorar o que está acontecendo. Utilize o seguinte comando.

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

Para obter mais informações, consulte [Função SWITCH](#).

O comando **runswchl**, CSQUTIL ou SWITCH transfere quaisquer mensagens da fila de transmissão antiga para a nova fila de transmissão. Quando o número de mensagens na fila de transmissão antiga para este canal atingir zero, o comutador é concluído. O comando é síncrono. O comando grava mensagens de progresso para a janela durante o processo de comutação.

Durante a fase de transferência, mensagens novas e existentes destinadas ao canal do emissor de clusters são transferidas para a nova fila de transmissão.

Como o canal do emissor de clusters está parado, as mensagens se acumulam na nova fila de transmissão. Compare o canal do emissor de clusters parado com a etapa “2” na página 66 em [“Comutando canais do emissor de clusters ativos para outro conjunto de filas de transmissão de cluster”](#) na página 66. Nessa etapa, o canal do emissor de clusters está em execução, portanto, as mensagens não necessariamente se acumulam na nova fila de transmissão.

4. Opcional: Monitorar os canais conforme eles são comutados

Em uma janela de comando diferente, exiba a profundidade da fila de transmissão durante a comutação. O exemplo a seguir exibe o status para filas de transmissão do cluster do sistema.

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

5. Opcional: Monitore as mensagens AMQ7341 A fila de transmissão para o canal *ChannelName* mudou da fila *QueueName* para a *QueueName* que estão gravadas no log de erros do gerenciador de filas.

6. Reinicie os canais do emissor de clusters que você parou.

Os canais não iniciam automaticamente, uma vez que você os parou, colocando-os no status STOPPED.

```
START CHANNEL(ChannelName)
```

Referências relacionadas

[runswchl](#)

[RESOLVE CHANNEL](#)

[STOP CHANNEL](#)

Armazenamento em cluster: melhores práticas de migração e modificação

Este tópico fornece orientação para planejamento e administração de clusters do IBM MQ. Essas informações são um guia baseado em teste e feedback de clientes.

1. “[Movendo objetos em um cluster](#)” na página 69 (Melhores práticas para mover objetos dentro de um cluster sem instalar quaisquer fix packs ou novas versões do IBM MQ).
2. “[Upgrades e instalações de manutenção](#)” na página 70 (Melhores práticas para manter uma arquitetura de cluster de trabalho em funcionamento, ao mesmo tempo aplicando manutenção ou upgrades e testando a nova arquitetura).

Movendo objetos em um cluster

Aplicativos e suas filas

Quando for preciso mover uma instância da fila hospedada em um gerenciador de filas para que ela seja hospedada em outro, é possível trabalhar com os parâmetros de balanceamento de carga para assegurar uma transição suave.

Crie uma instância da fila onde ela deve ser passar a ser hospedada, mas use as configurações de balanceamento de carga de trabalho do cluster para continuar enviando mensagens para a instância original até que seu aplicativo esteja pronto para mudar. Isso é feito com as seguintes etapas:

1. Configure a propriedade **CLWLRANK** da fila existente para um valor alto, por exemplo, cinco.
2. Crie a nova instância da fila e configure sua propriedade **CLWLRANK** para zero.
3. Conclua qualquer outra configuração do novo sistema, por exemplo, implementar e iniciar aplicativos consumidores com relação à nova instância da fila.
4. Configure a propriedade **CLWLRANK** da nova instância de fila para ser maior do que a instância original, por exemplo, nove.
5. Permita que a instância da fila original processe quaisquer mensagens enfileiradas no sistema e, em seguida, exclua a fila.

Movendo gerenciadores de fila inteiros

Se o gerenciador de filas ficar no mesmo host, mas o endereço IP estiver mudando, o processo é o seguinte:

- DNS, quando usado corretamente, pode ajudar a simplificar o processo. Para obter informações sobre o uso do DNS, configurando o atributo do canal [Connection name \(CONNNAME\)](#), consulte [ALTER CHANNEL](#).
- Se estiver movendo um repositório completo, assegure de ter pelo menos um outro repositório completo que esteja sendo executado sem problemas (sem problemas com status de canal, por exemplo) antes de fazer mudanças.
- Suspenda o gerenciador de filas usando o comando [SUSPEND QMGR](#) para evitar acúmulo de tráfego.
- Modifique o endereço IP do computador. Se sua definição de canal CLUSRCVR utilizar um endereço IP no campo CONNNAME, modifique essa entrada de endereço IP. O cache DNS pode precisar ser esvaziado para garantir que as atualizações estejam disponíveis globalmente.
- Quando o gerenciador de filas reconecta-se aos repositórios completos, as autodefinições de canal resolvem-se automaticamente.
- Se o gerenciador de filas tiver hospedado um repositório completo e o endereço IP mudar, é importante assegurar que as parciais sejam comutadas assim que possível para apontar quaisquer canais CLUSSDR definidos manualmente para o novo local. Até que essa comutação seja realizada, esses gerenciadores de filas podem estar apto a entrar em contato apenas com o repositório completo (inalterado) restante e poderão ser vistas mensagens de aviso sobre a definição de canal incorreta.
- Continue o gerenciador de filas usando o comando [RESUME QMGR](#).

Se o gerenciador de filas precisar ser movido para um novo host, é possível copiar os dados do gerenciador de filas e restaurá-los de um backup. No entanto, esse processo não é recomendado, a menos que não haja outras opções; pode ser melhor criar um gerenciador de filas em uma nova

máquina e replicar filas e aplicativos conforme descrito na seção anterior. Essa situação fornece um bom mecanismo distribuição/reversão.

Se você estiver determinado a mover um gerenciador de filas completo usando backup, siga essas melhores práticas:

- Trate o processo inteiro como uma restauração do gerenciador de filas a partir do backup, aplicando quaisquer processos que geralmente usa para recuperação do sistema conforme apropriado para seu ambiente de sistema operacional.
- Use o comando **REFRESH CLUSTER** após a migração para descartar todas as informações do cluster mantidas localmente (incluindo quaisquer canais definidos automaticamente que estejam em dúvida) e forçar a sua reconstrução.

Nota: Para grandes clusters, usando o comando **REFRESH CLUSTER** pode ser disruptivo para o cluster enquanto ele está em andamento e novamente em intervalos de 27 dias, quando os objetos de cluster enviam automaticamente atualizações de status para todos os gerenciadores de filas de seu interesse. Consulte [Atualizando em um grande cluster pode afetar o desempenho e disponibilidade do cluster](#).

Ao criar um gerenciador de filas e replicar a configuração a partir de um gerenciador de filas existente no cluster (conforme descrito anteriormente neste tópico), nunca trate os dois gerenciadores de filas diferentes como sendo iguais. Em particular, não dê a um novo gerenciador de filas o mesmo nome e endereço IP de gerenciador de filas. Tentar "largar" um gerenciador de filas substituto é uma causa comum de problemas em clusters do IBM MQ. O cache espera receber atualizações incluindo o atributo **QMID** e o estado pode estar corrompido.

Se dois gerenciadores de filas diferentes forem criados acidentalmente com o mesmo nome, recomenda-se utilizar o comando [CLUSTER DE RESET QMID](#) para ejetar a entrada incorreta do cluster.

Upgrades e instalações de manutenção

Evite o cenário do big bang (por exemplo, parar toda a atividade de clusters e gerenciadores de filas, aplicar todos os upgrades e manutenção a todos os gerenciadores de filas e, em seguida, iniciar tudo ao mesmo tempo). Os clusters são projetados para ainda funcionarem com múltiplas versões do gerenciador de filas coexistentes, de modo que uma abordagem de manutenção bem planejada e em fases é recomendada.

Tenha um plano de backup:

- Você fez backups?
- Evite usar a funcionalidade do novo cluster imediatamente: aguarde até ter certeza de que todos os gerenciadores de filas estão atualizados para o novo nível e de que você não irá recuperar nenhum deles. Usar a nova função de cluster em um cluster em que alguns gerenciadores de filas ainda estejam em um nível anterior pode levar a um comportamento indefinido.

Um repositório armazena um registro que ele recebe em sua própria versão. Se o registro recebido estiver em uma versão mais recente, os atributos da versão mais recente serão descartados quando o registro for armazenado. Um gerenciador de filas IBM MQ 9.3 que recebe informações sobre um gerenciador de filas IBM MQ 9.4 armazena apenas as informações do IBM MQ 9.3. Um repositório do IBM MQ 9.4 que recebe um registro do IBM MQ 9.3 armazena valores padrão para atributos introduzidos na versão mais recente. Os padrões definem os valores para os atributos que não estão incluídos no registro que ele recebe.

Migre repositórios completos primeiro. Embora eles consigam encaminhar informações que não compreendem, não conseguem fazer com que elas persistam, portanto, não é a abordagem recomendada, a menos que seja absolutamente necessário. Para obter mais informações, consulte [Migração do cluster de gerenciador de filas](#).

Armazenamento em Cluster: Usando Melhores Práticas de REFRESH CLUSTER

Use o comando **REFRESH CLUSTER** para descartar todas as informações sobre um cluster mantidas localmente e reconstruir essas informações do repositórios completos no cluster. Não deve ser necessário usar esse comando, exceto em circunstâncias excepcionais. Se você precisar usá-lo, existem

considerações especiais sobre como fazer isso. Essas informações são um guia baseado em teste e feedback de clientes.

Somente execute REFRESH CLUSTER se realmente precisar fazer isso

A tecnologia de cluster do IBM MQ assegura que qualquer mudança na configuração de cluster, como uma mudança em uma fila em cluster, automaticamente se torne conhecida para qualquer membro do cluster que precise ter essa informação. Não há necessidade de tomar novas medidas administrativas para realizar essa propagação de informações.

Se tais informações não chegarem aos gerenciadores de filas no cluster quando for necessário, por exemplo, uma fila em cluster não for conhecida por outro gerenciador de filas no cluster quando um aplicativo tentar abri-la pela primeira vez, isso sugere um problema na infraestrutura do cluster. Por exemplo, é possível que um canal não possa ser iniciado entre um gerenciador de filas e um gerenciador de filas de repositório completo. Portanto, qualquer situação em que inconsistências sejam observadas devem ser investigadas. Se possível, resolva a situação sem usar o comando **REFRESH CLUSTER**.

Em raras circunstâncias que estão documentadas em outro lugar nesta documentação do produto, ou quando solicitado pelo suporte do IBM, é possível usar o comando **REFRESH CLUSTER** para descartar todas as informações sobre um cluster mantidas localmente e reconstruir essas informações a partir dos repositórios completos no cluster.

Atualizar em um cluster grande pode afetar o desempenho e a disponibilidade do cluster

O uso do comando **REFRESH CLUSTER** pode ser disruptivo para o cluster enquanto ele estiver em andamento, por exemplo, criando um aumento repentino no trabalho para os repositórios completos conforme eles processam a repropagação de recursos do cluster de gerenciador de filas. Se você estiver atualizando em um cluster grande (ou seja, muitas centenas de gerenciadores de filas), deve evitar usar do comando no trabalho cotidiano, se possível, e usar métodos alternativos para corrigir inconsistências específicas. Por exemplo, se uma fila de clusters não estiver sendo corretamente propagada por todo o cluster, uma técnica de investigação inicial de atualizar a definição de fila em cluster, como alterar sua descrição, repropaga a configuração da fila no cluster. Esse processo pode ajudar a identificar o problema e potencialmente resolver uma inconsistência temporária.

Se métodos alternativos não puderem ser usados, e você tiver de executar **REFRESH CLUSTER** em um cluster grande, faça isso fora de horários de pico ou durante uma janela de manutenção para evitar afetar as cargas de usuário. Também evite atualizar um cluster grande em um único lote; em vez disso, coordene a atividade conforme explicado em [“Evitar problemas de desempenho e de disponibilidade quando os objetos de cluster enviam atualizações automáticas”](#) na página 71.

Evitar problemas de desempenho e de disponibilidade quando os objetos de cluster enviam atualizações automáticas

Depois que um objeto novo cluster é definido em um gerenciador de filas do cluster, uma atualização para esse objeto é gerada a cada 27 dias a partir do horário de definição e enviada para cada repositório completo no cluster e adiante para quaisquer outros gerenciadores de filas interessados. Ao emitir o comando **REFRESH CLUSTER** a um gerenciador de filas, reconfigure o relógio para essa atualização automática em todos os objetos definidos localmente no cluster especificado.

Se você atualizar um cluster grande (ou seja, muitas centenas de gerenciadores de filas) em um único lote, ou em outras circunstâncias, como a recriação de um sistema de backup de configuração, após 27 dias, todos os gerenciadores de filas serão novamente informados de todas as suas definições de objeto para os repositórios completos ao mesmo tempo. Isso novamente pode fazer com que o sistema opere de modo significativamente mais lento, ou mesmo fique indisponível, até que todas as atualizações sejam concluídas. Portanto, quando você precisa atualizar ou recriar múltiplos gerenciadores de filas em um cluster grande, é necessário escalonar a atividade durante várias horas ou vários dias para que atualizações automáticas subsequentes não afetem regularmente o desempenho do sistema.

A fila de histórico de cluster do sistema

Quando um **REFRESH CLUSTER** é executado, o gerenciador de filas faz uma captura instantânea do estado do cluster antes da atualização e a armazena no `SYSTEM.CLUSTER.HISTORY.QUEUE (SCHQ)` se ele estiver definido no gerenciador de filas. Essa captura instantânea é apenas para fins de serviço do IBM, no caso de problemas posteriores com o sistema.

O SCHQ é definido por padrão em gerenciadores de filas distribuídos na inicialização. Para migração do z/OS, o SCHQ deve ser definido manualmente.

Mensagens na SCHQ expiram após três meses.

Conceitos relacionados

[“Contraprestações de REFRESH CLUSTER para clusters de publicação/assinatura” na página 108](#)

A emissão do comando **REFRESH CLUSTER** resulta em o gerenciador de filas temporariamente descartar informações mantidas localmente sobre um cluster, incluindo quaisquer tópicos de cluster e suas assinaturas de proxy associadas.

Referências relacionadas

[Problemas do aplicativo vistos ao executar REFRESH CLUSTER](#)

[Referência de Comandos MQSC: REFRESH CLUSTER](#)

Armazenamento em cluster: disponibilidade, várias instâncias e recuperação de desastre

Este tópico fornece orientação para planejamento e administração de clusters do IBM MQ. Essas informações são um guia baseado em teste e feedback de clientes.

IBM MQ Clusterização em si não é uma solução de alta disponibilidade, mas, em algumas circunstâncias, pode ser usada para melhorar a disponibilidade de serviços usando o IBM MQ, por exemplo, tendo várias instâncias de uma fila em gerenciadores de filas diferentes. Esta seção fornece orientação sobre como assegurar que a infraestrutura do IBM MQ seja o mais altamente disponível possível para que possa ser usada em tal arquitetura.

Nota: Outras soluções de alta disponibilidade e de recuperação de desastre estão disponíveis para IBM MQ, consulte [Configurando alta disponibilidade, recuperação e reinicialização](#).

Disponibilidade de recursos de cluster

A razão para a recomendação normal de manter dois repositórios completos é que a perda de um não é crítica para o funcionamento tranquilo do cluster. Mesmo que ambos fiquem indisponíveis, há um período de carência de 60 dias para o conhecimento existente em repositórios parciais, embora recursos novos ou não acessados anteriormente (filas, por exemplo) não estejam disponíveis neste evento.

Usando clusters para melhorar a disponibilidade do aplicativo

Um cluster pode ajudar no design aplicativos altamente disponíveis (por exemplo, um aplicativo de servidor do tipo solicitação/resposta) usando várias instâncias da fila e do aplicativo. Se necessário, atributos de prioridade podem dar preferência ao aplicativo 'ativo', a menos que um gerenciador de filas ou canal, por exemplo, fique indisponível. Isso é eficiente para alternar rapidamente para continuar processando novas mensagens quando um problema ocorre.

No entanto, mensagens que foram entregues a um gerenciador de filas específico em um cluster são suspensas somente nessa instância da fila, e não estão disponíveis para processamento até que esse gerenciador de filas seja recuperado. Por essa razão, para verdadeira alta disponibilidade de dados, é possível considerar outras tecnologias, como gerenciadores de filas de várias instâncias.


Gerenciadores de Filas de Várias Instâncias

O Software High Availability (várias instâncias) é uma oferta integrada para manter suas mensagens existentes disponíveis. Veja [Usando IBM MQ com configurações de alta disponibilidade](#), [Criar um gerenciador de filas de várias instâncias](#) e a seção a seguir para obter mais informações. Qualquer gerenciador de filas em um cluster pode se tornar altamente disponível usando esta técnica, contanto que todos os gerenciadores de filas no cluster estejam em execução pelo menos IBM WebSphere MQ 7.0.1. Se quaisquer gerenciadores de filas no cluster estiverem em níveis anteriores, eles poderão perder a conectividade com os gerenciadores de filas de várias instâncias se fizerem failover para um IP secundário.

Como já descrito neste tópico, contanto que dois repositórios completos estejam configurados, eles são quase por natureza altamente disponíveis. Se necessário, os gerenciadores de filas de várias instâncias/alta disponibilidade de software IBM MQ podem ser usados para repositórios completos. Não há motivo forte para usar esses métodos, e, na verdade, para indisponibilidades temporárias, esses métodos podem causar custo de desempenho adicional durante o failover. Usar a alta disponibilidade de software em vez de executar dois repositórios completos é desaconselhável porque, no caso de uma interrupção de canal único, por exemplo, isso não necessariamente realizaria failover, mas poderia deixar repositórios parciais incapazes de consultar recursos de cluster.

Recuperação de Desastre

É difícil realizar uma boa recuperação de Desastre, por exemplo, a recuperação de quando os discos que armazenam dados de um gerenciador de filas são corrompidos; o IBM MQ pode ajudar, mas ele não pode fazer isso automaticamente. A única opção de recuperação de desastres "true" em IBM MQ (exceto qualquer sistema operacional ou outras tecnologias de replicação subjacentes) é a restauração a partir de um backup. Há alguns pontos específicos do cluster a considerar nestas situações:

- Tome cuidado ao testar cenários de recuperação de desastre. Por exemplo, se estiver testando a operação de gerenciador de filas de backup, tenha cuidado ao colocá-los on-line, uma vez que é possível acidentalmente unir o cluster em tempo real e começar a "roubar" mensagens hospedando filas com o mesmo nome nos gerenciadores de fila de cluster em tempo real.
- Teste de recuperação de desastre não deve interferir em um cluster em tempo real em execução. Técnicas para evitar interferência incluem:
 - Concluir a separação da rede ou a separação no nível de firewall.
 -  Não iniciar inicialização do canal ou o espaço de endereço do z/OS **chinit**.
 - Não emitir certificado TLS em tempo real para o sistema de recuperação de desastre até que, ou a menos que, um cenário de recuperação de desastre real ocorra.
- Ao restaurar um backup de um gerenciador de filas no cluster, é possível que o backup esteja fora de sincronia com o restante do cluster. O comando **REFRESH CLUSTER** pode resolver atualizações e sincronizar-se com o cluster, mas o comando **REFRESH CLUSTER** deve ser usado como último recurso. Consulte “Armazenamento em Cluster: Usando Melhores Práticas de REFRESH CLUSTER” [na página 70](#). Revise qualquer documentação de processo interna e a documentação do IBM MQ para ver se uma etapa simples foi ignorada antes de recorrer ao uso do comando.
- Como para qualquer recuperação, os aplicativos devem lidar com reprodução e perda de dados. Deve-se decidir limpar as filas até um estado conhecido ou se há informações suficientes em outro lugar para gerenciar reproduções.

Planejando sua rede publicar/assinar distribuída

É possível criar uma rede de gerenciadores de filas nos quais as assinaturas criadas em um gerenciador de filas receberão as mensagens correspondentes publicadas por um aplicativo conectado a outro gerenciador de filas na rede. Para escolher uma topologia adequada, considere seus requisitos de controle manual, tamanho da rede, frequência de alteração, disponibilidade e escalabilidade.

Antes de começar

Esta tarefa presume que você entenda o que são redes de publicar/assinar distribuídas e como elas funcionam. Para uma visão geral técnica, consulte [Redes de publicar/assinar distribuídas](#).

Sobre esta tarefa

Há três topologias básicas para uma rede de publicar/assinar:

- Cluster encaminhado direto
- Cluster roteado para host de tópico
- hierarquia

Para as duas primeiras topologias, o ponto de partida é um cluster de configuração do IBM MQ. A terceira topologia pode ser criada com ou sem um cluster. Veja [“Planejando filas distribuídas e clusters”](#) na página 20 para obter informações sobre como planejar a rede do gerenciador de filas subjacente.

Um *Cluster roteado diretamente* é a topologia mais simples de configurar quando um cluster já está presente. Qualquer tópico que você defina em qualquer gerenciador de filas é disponibilizado automaticamente em cada gerenciador de filas no cluster, e as publicações são roteadas diretamente a partir de qualquer gerenciador de filas no qual um aplicativo de publicação se conecta para cada um dos gerenciadores de filas nos quais as assinaturas correspondentes existem. Essa simplicidade de configuração depende de o IBM MQ manter um nível elevado de compartilhamento de informação e da conectividade entre cada gerenciador de filas no cluster. Para redes pequenas e simples (isto é, um número pequeno de gerenciadores de filas e um conjunto razoavelmente estático de publicadores e assinantes), isso é aceitável. No entanto, quando utilizados em ambientes maiores ou mais dinâmicos, a sobrecarga pode ser proibitiva. Consulte o [“Roteamento direto em clusters de publicar/assinar”](#) na página 79.

Um *Cluster roteado para host de tópico* fornece o mesmo benefício que um cluster roteado direto, tornando qualquer tópico que você define em qualquer gerenciador de filas no cluster automaticamente disponível em cada gerenciador de filas no cluster. No entanto, os clusters roteados por host de tópico requerem que você escolha cuidadosamente os gerenciadores de filas que hospedam cada tópico, porque todas as informações e publicações para esse tópico passam por esses gerenciadores de filas de host de tópico. Isso significa que o sistema não precisa manter fluxos de canais e informações entre todos os gerenciadores de filas. No entanto também significa que as publicações podem não ser mais enviadas diretamente para os assinantes, mas roteadas por meio de um gerenciador de filas do host do tópico. Por esses motivos, pode ser imposta uma carga adicional ao sistema, especialmente nos gerenciadores de filas que hospedam os tópicos, portanto, é necessário um planejamento cuidadoso da topologia. Essa topologia é particularmente efetiva para redes que contêm vários gerenciadores de filas ou que hospedam um conjunto dinâmico de publicadores e assinantes (ou seja, publicadores e assinantes que são frequentemente incluído ou removido). Hosts de tópico adicionais podem ser definidos para melhorar a disponibilidade de rotas e para escalar horizontalmente a carga de trabalho de publicação. Consulte o [“Roteamento de host do tópico em clusters de publicação/assinatura”](#) na página 84.

Uma *hierarquia* requer a configuração mais manual para instalar e é a topologia mais difícil de modificar. Deve-se configurar manualmente os relacionamentos entre cada gerenciador de filas na hierarquia e suas relações diretas. Depois de os relacionamentos terem sido configurados, as publicações (como para as duas topologias anteriores) serão roteadas para assinaturas em outros gerenciadores de filas na hierarquia. As publicações são roteadas usando os relacionamentos de hierarquia. Isso permite que topologias muito específicas sejam configuradas para atender a requisitos diferentes, mas também pode resultar em publicações que requerem muitos "hops" por meio de gerenciadores de filas intermediários para atingir as assinaturas. Há sempre apenas uma rota através de uma hierarquia para uma publicação, portanto, a disponibilidade de cada gerenciador de filas é crítica. Hierarquias normalmente são preferíveis apenas quando não é possível configurar um único cluster; por exemplo, ao abranger diversas organizações. Consulte o [“Roteamento em hierarquias de publicação/assinatura”](#) na página 109.

Se necessário, as três topologias acima podem ser combinados para resolver requisitos topográficos específicos. Para obter um exemplo, veja [Combinando os espaços de tópico de diversos clusters](#).

Para escolher uma topologia apropriada para sua rede distribuída de publicação/assinatura, é preciso considerar as seguintes questões gerais:

- Qual será o tamanho da sua rede?
- De quanto controle manual você precisa sobre sua configuração?
- Quão dinâmico será o sistema, tanto em termos de tópicos e assinaturas quanto em termos de gerenciadores de fila?
- Quais são seus requisitos de disponibilidade e escalabilidade?
- Todos os gerenciadores de filas podem ser conectados diretamente uns aos outros?

Procedimento

- Estime o tamanho que sua rede precisa ter.
 - a) Estime o número de tópicos de que você precisa.
 - b) Estime o número de publicadores e assinantes que você espera ter.
 - c) Estime quantos gerenciadores de filas estarão envolvidos em atividades de publicar/assinar.

Veja também [“Publicar/assinar cluster: melhores práticas”](#) na página 94, principalmente as seguintes seções:

- [Como dimensionar seu sistema](#)
- [Razões para limitar o número de gerenciadores de filas do cluster envolvidos na atividade de publicação/assinatura](#)
- [Como decidir quais tópicos agrupar](#)

Se a sua rede for ter vários gerenciadores de filas e manipular muitos publicadores e assinantes, provavelmente será necessário usar um cluster roteado para host de tópico ou uma hierarquia. Clusters roteados diretamente requerem quase nenhuma configuração manual, e podem ser uma boa solução para redes pequenas ou estáticas.

- Considere de quanto controle manual você precisa sobre qual gerenciador de filas hospeda cada tópico, publicador ou assinante.
 - a) Considere se alguns de seus gerenciadores de filas são menos capazes do que outros.
 - b) Considere se os links de comunicação para alguns de seus gerenciadores de filas são mais frágeis do que para outros.
 - c) Identifique casos em que você espera que um tópico tenha muitas publicações e poucos assinantes.
 - d) Identifique casos em que você espera que um tópico tenha vários assinantes e poucas publicações.

Em todas as topologias, as publicações são entregues para assinaturas em outros gerenciadores de filas. Em um cluster roteado diretamente, essas publicações seguem o caminho mais curto para as assinaturas. Em um cluster roteado para host de tópico ou uma hierarquia, você controla a rota que as publicações assumem. Se os gerenciadores de filas diferirem em termos de capacidade, ou tiverem diferentes níveis de disponibilidade e de conectividade, você provavelmente desejará designar cargas específicas a gerenciadores de filas específicos. É possível fazer isso usando cluster roteado por host de tópico ou hierarquia.

Em todas as topologias, co-localizar os aplicativos de publicação no mesmo gerenciador de filas que as assinaturas sempre que possível minimiza sobrecargas e maximiza o desempenho. Para clusters roteados por host de tópico, considere colocar publicadores ou assinantes nos gerenciadores de filas que hospedam o tópico. Isso elimina quaisquer "hops" extras entre os gerenciadores de filas para transmitir uma publicação para um assinante. Essa abordagem é particularmente eficaz em casos em que um tópico tem muitos publicadores e poucos assinantes ou muitos assinantes e poucos publicadores. Veja, por exemplo, [Roteamento de host de tópico usando publicadores ou assinantes centralizados](#).

Veja também [“Publicar/assinar cluster: melhores práticas”](#) na página 94, principalmente as seguintes seções:

- [Como decidir quais tópicos agrupar](#)
- [Local do publicador e da assinatura](#)

- Considere o quão dinâmica a atividade de rede será.
 - a) Estime com que frequência os assinantes serão incluídos e removidos em tópicos diferentes.

Sempre que uma assinatura for incluída ou removida de um gerenciador de filas, e ela for a primeira ou a última assinatura para essa sequência de tópico específica, essas informações serão comunicadas para outros gerenciadores de filas na topologia. Em um cluster roteado diretamente e uma hierarquia, essas informações de assinatura são propagadas para todos os gerenciadores

de fila na topologia, tenham eles publicadores no tópico ou não. Se a topologia consistir em vários gerenciadores de filas, isso pode ser uma sobrecarga de desempenho significativa. Em um cluster roteado para host de tópico, essas informações só são propagadas para os gerenciadores de filas que hospedam um tópico em cluster que é mapeado para a sequência de tópicos da assinatura.

Veja também a seção Alteração de assinatura e sequências de tópicos dinâmicas do “Publicar/ assinar cluster: melhores práticas” na página 94.

Nota: Em sistemas muito dinâmicos, em que o conjunto de muitas sequências de tópicos exclusivas é alterado com rapidez e constantemente, pode ser melhor trocar o modelo para um modo de "publicar em qualquer lugar". Consulte Desempenho de assinatura em redes publicar/ assinar.

b) Considere o quão dinâmicos os gerenciadores de filas são na topologia.

Uma hierarquia requer que cada mudança no gerenciador de filas na topologia seja manualmente inserida ou removida da hierarquia, com cautela ao alterar gerenciadores de filas em níveis mais altos na hierarquia. Geralmente, os gerenciadores de filas em uma hierarquia também usam conexões de canal configuradas manualmente. Deve-se manter essas conexões, incluindo e removendo canais conforme os gerenciadores de fila são incluídos e removidos da hierarquia.

Em um cluster de publicação/assinatura, gerenciadores de filas são conectados automaticamente a qualquer outro gerenciador de filas necessário quando são unidos ao cluster pela primeira vez, e automaticamente se tornam cientes de tópicos e assinaturas.

- Considere as suas necessidades de disponibilidade de rota e escalabilidade de tráfego de publicação.
 - a) Decida se precisa sempre ter uma rota disponível de um gerenciador de filas de publicação para um gerenciador de filas de assinatura, mesmo quando um gerenciador de filas estiver indisponível.
 - b) Considere o quão escalável você precisa que a rede seja. Decida se o nível de tráfego de publicação é muito alto para ser roteado através de um único gerenciador de filas ou canal e se esse nível de tráfego de publicação deve ser tratado por uma ramificação único tópico ou pode ser distribuído em várias ramificações de tópico.
 - c) Considere se você precisa manter a ordenação de mensagens.

Uma vez que um cluster roteado diretamente envia mensagens diretamente de gerenciadores de filas de publicação para gerenciadores de filas de assinatura, não é preciso considerar a disponibilidade de gerenciadores de filas intermediários ao longo da rota. Da mesma forma, escalar para gerenciadores de filas intermediários não é uma consideração. No entanto, conforme já mencionado, a sobrecarga da manutenção automática de canais e fluxos de informação entre todos os gerenciadores de filas no cluster pode afetar muito o desempenho, em especial em um ambiente grande ou dinâmico.

Um cluster roteados por host de tópico pode ser ajustado para tópicos individuais. É possível garantir que cada ramificação da árvore de tópico que tenha uma publicação de carga seja definida em um gerenciador de filas diferente, e que cada gerenciador de filas seja eficiente e disponível o bastante para a carga de trabalho esperada para essa ramificação da árvore de tópicos. Também é possível aprimorar ainda mais a disponibilidade e a escala horizontal definindo cada tópico em vários gerenciadores de filas. Isso permite que o sistema roteie ao redor de gerenciadores de filas de host de tópico indisponíveis e balanceie a carga de trabalho do tráfego de publicação entre eles. Entretanto, ao definir um determinado tópico em vários gerenciadores de filas, também são introduzidas as seguintes restrições:

- Você perde ordenação de mensagens entre publicações.
- Não é possível usar publicações retidas. Consulte o “Considerações de design para publicações retidas em clusters de publicação/assinatura” na página 107.

Não é possível configurar alta disponibilidade ou escalabilidade de roteamento em uma hierarquia através de várias rotas.

Veja também a seção Tráfego de publicação de “Publicar/assinar cluster: melhores práticas” na página 94.

- Com base nesses cálculos, use os links fornecidos para ajudar a decidir se você irá usar um cluster roteado para host de tópico, um cluster roteado diretamente, uma hierarquia ou uma mistura dessas topologias.

Como proceder a seguir

Agora você está pronto para configurar sua rede de publicar/assinar distribuída.

Tarefas relacionadas

[Configurando um cluster do gerenciador de filas](#)

[Configurando enfileiramento distribuído](#)

[Configurando um cluster de publicação/assinatura](#)

[Conectando um gerenciador de filas a hierarquia publicar/assinar](#)

Projetando clusters publicar/assinar

Há duas topologias básicas do cluster de publicação/assinatura: *roteamento direto* e *roteamento de host de tópico*. Cada uma possui benefícios diferentes. Ao projetar seu cluster de publicação/assinatura, escolha a topologia que melhor atenda às suas necessidades de rede esperadas.

Para uma visão geral das duas topologias de cluster de publicação/assinatura, consulte [Clusters de publicação/Assinatura](#). Para ajudá-lo a avaliar suas necessidades de rede, veja [“Planejando sua rede publicar/assinar distribuída”](#) na página 73 e [“Publicar/assinar cluster: melhores práticas”](#) na página 94.

Em geral, ambas as topologias do cluster fornecem os seguintes benefícios:

- Configuração simples na parte superior de uma topologia de cluster ponto a ponto.
- Tratamento automático de gerenciadores de filas entrando e saindo do cluster.
- Facilidade de ajuste de escala para assinaturas e publicadores adicionais, incluindo gerenciadores de filas extras e distribuição de assinaturas e publicadores adicionais entre eles.

No entanto, as duas topologias possuem benefícios diferentes conforme os requisitos se tornam mais específicos.

Clusters de Publicação/Assinatura Encaminhados Direto

Com roteamento direto, qualquer gerenciador de filas no cluster envia publicações de aplicativos conectados diretamente para qualquer outro gerenciador de filas no cluster com uma assinatura correspondente.

Um cluster de publicação/assinatura encaminhado direto fornece os seguintes benefícios:

- Mensagens destinadas para uma assinatura em um gerenciador de filas específico no mesmo cluster são transportadas diretamente para aquele gerenciador de filas e não precisam passar por um gerenciador de filas intermediário. Isso pode melhorar o desempenho em comparação com uma topologia encaminhada direta ou uma topologia hierárquica.
- Uma vez que todos os gerenciadores de filas são diretamente conectados uns aos outros, não há um único ponto de falha na infraestrutura de roteamento dessa topologia. Se um gerenciador de filas não estiver disponível, assinaturas em outros gerenciadores de filas no cluster ainda poderão receber mensagens de publicadores em gerenciadores de filas disponíveis.
- É muito simples de configurar, especialmente em um cluster existente.

Coisas a serem consideradas ao usar um cluster de publicação/assinatura encaminhado direto:

- Todos os gerenciadores de filas no cluster se tornam cientes de todos os outros gerenciadores de filas no cluster.
- Os gerenciadores de filas em um cluster que hospedam uma ou mais assinaturas para um tópico em cluster criam automaticamente os canais do emissor de clusters para todos os outros gerenciadores de filas no cluster, mesmo quando os gerenciadores de filas não estão publicando mensagens em nenhum tópico em cluster.

- A primeira assinatura em um gerenciador de filas para uma sequência de tópicos em um tópico em cluster resulta no envio de uma mensagem para todos os outros gerenciadores de filas no cluster. Da mesma forma, a última assinatura em uma cadeia de tópico a ser excluída também resulta em uma mensagem. Quanto mais sequências de tópicos individuais são usadas sob um tópico em cluster, e quanto maior a taxa de mudança de assinaturas, mais comunicação entre gerenciador de filas ocorre.
- Cada gerenciador de filas no cluster mantém o conhecimento de sequências de tópico assinadas das quais está informado, mesmo quando o gerenciador de filas não está publicando nem assinando esses tópicos.

Pelas razões acima, todos os gerenciadores de filas em um cluster com um tópico de encaminhamento direto definido terão uma sobrecarga adicional. Quanto mais gerenciadores de filas estão no cluster, maior a sobrecarga. Do mesmo modo, quanto mais sequências de tópico estão inscritas, e o maior a sua taxa de mudança, maior é a sobrecarga. Isso pode resultar em muita carga sobre gerenciadores de filas em execução em sistemas pequenos em um cluster de publicação/assinatura com roteamento direto grande ou dinâmico. Veja [Desempenho de publicar/assinar com roteamento direto](#) para obter informações adicionais.

Quando você sabe que um cluster não pode acomodar as sobrecargas de publicar/assinar em cluster com roteamento direto, é possível usar, em vez disso, [publicar/assinar com roteamento de host do tópico](#). Como alternativa, em situações extremas, é possível desativar completamente a funcionalidade do cluster de publicação/assinatura, definindo o atributo do gerenciador de filas **PSCLUS** para DISABLED em cada gerenciador de filas no cluster. Consulte o [“Inibindo publicação/assinatura em cluster” na página 104](#). Isso impede a criação de qualquer tópico em cluster, portanto, assegura que a rede não incorra em qualquer sobrecarga associada a publicar/assinar em cluster.

Clusters de Publicação/Assinatura Roteados do Host do Tópico

Com o roteamento do host do tópico, os gerenciadores de filas nos quais tópicos em cluster são definidos administrativamente tornam-se roteadores para publicações. Publicações de gerenciadores de filas não de hosting no cluster são roteadas através do gerenciador de filas de hosting para qualquer gerenciador de filas no cluster com uma assinatura correspondente.

Um cluster de publicação/assinatura encaminhado para host do tópico fornece os seguintes benefícios adicionais sobre um cluster de publicação/assinatura encaminhado direto:

- Apenas os gerenciadores de filas em que são definidos os tópicos encaminhados por host de tópico estão cientes de todos os outros gerenciadores de filas no cluster.
- Somente os gerenciadores de filas de host de tópico precisam poder se conectar a todos os outros gerenciadores de filas no cluster, e geralmente só irão se conectar a eles quando houver assinaturas. Portanto, há significativamente menos canais em execução entre gerenciadores de filas.
- Gerenciadores de fila do cluster que hospedam uma ou mais assinaturas para um tópico em cluster criam automaticamente os canais do emissor de clusters apenas para gerenciadores de filas que hospedam um tópico de cluster que é mapeado para a sequência de tópicos da assinatura.
- A primeira assinatura em um gerenciador de filas para uma sequência de tópicos em um tópico em cluster resulta em uma mensagem ser enviada para um gerenciador de filas no cluster que hospeda o tópico em cluster. Da mesma forma, a última assinatura em uma cadeia de tópico a ser excluída também resulta em uma mensagem. Quanto mais sequências de tópicos individuais estão sendo usadas em um tópico de cluster, e quanto maior a taxa de mudança de assinaturas, mais comunicação do gerenciador de entre filas ocorre, mas somente entre hosts de assinatura e hosts de tópico.
- Mais controle sobre a configuração física. Com o roteamento direto, todos os gerenciadores de filas precisam participar do cluster de publicação/assinatura, aumentando a sobrecarga. Com roteamento de host de tópico, somente os gerenciadores de filas de host de tópico estão cientes de outros gerenciadores de filas e suas assinaturas. Você escolhe explicitamente os gerenciadores de filas de host de tópico, portanto, pode assegurar que esses gerenciadores de filas estejam em execução no equipamento adequado, e pode usar sistemas menos potentes para os outros gerenciadores de filas.

Coisas a serem consideradas ao usar um cluster de publicação/assinatura roteado do host do tópico:

- Um "hop" extra entre um gerenciador de filas de publicação e um gerenciador de filas de assinatura é introduzido quando o publicador ou o assinante não está localizado em um gerenciador de filas de hosting de tópico. A latência causada pelo "hop" extra pode significar que o roteamento do host do tópico é menos eficiente do que o roteamento direto.
- Em clusters grandes, o roteamento de host de tópico alivia importantes problemas de desempenho e de ajuste de escala que podem ocorrer com roteamento direto.
- Pode-se optar por definir todos os tópicos em um único gerenciador de filas ou em um muito pequeno de gerenciadores de filas. Se você fizer isto, certifique-se de que os gerenciadores de filas do host do tópico estejam hospedados em sistemas eficientes com boa conectividade.
- É possível definir o mesmo tópico em mais de um gerenciador de filas. Isso melhora a disponibilidade do tópico e também a escalabilidade, pois a carga de trabalho do IBM MQ equilibra as publicações para um tópico entre todos os hosts para esse tópico. Observe, no entanto, que definir o mesmo tópico em mais de um gerenciador de filas perde a ordem das mensagens para esse tópico.
- Ao hospedar os diferentes tópicos em diferentes gerenciadores de filas, é possível pode melhorar a escalabilidade sem perder a ordem de mensagens.

Tarefas relacionadas

Cenário: Criando um Cluster de Publicação / Assinatura

Configurando um cluster de publicação/assinatura

Ajustando redes de publicar/assinar distribuídas

Resolução de problemas de publicação / assinatura distribuída

Roteamento direto em clusters de publicar/assinar

Publicações de qualquer gerenciador de filas de publicação são encaminhadas diretamente para qualquer outro gerenciador de filas no cluster com uma assinatura correspondente.

Para uma introdução de como as mensagens são encaminhadas entre os gerenciadores de filas em hierarquias de publicação/subscrição e clusters, consulte Redes de publicação/assinatura distribuídas.

Um cluster de publicação/assinatura encaminhado direto comporta-se da maneira a seguir:

- Todos os gerenciadores de filas automaticamente sabem sobre todos os outros gerenciadores de filas.
- Todos os gerenciadores de filas com assinaturas para tópicos em cluster criam canais para todos os outros gerenciadores de filas no cluster e informam-nos de suas assinaturas.
- Mensagens publicadas por um aplicativo são roteadas do gerenciador de filas ao qual ele está conectado, direto para cada gerenciador de filas em que uma assinatura correspondente existir.

O diagrama a seguir mostra um cluster gerenciador de filas que não é usado atualmente para atividades de publicação/assinatura ou ponto a ponto. Observe que cada gerenciador de filas no cluster se conecta apenas para e dos gerenciadores de filas de repositório completo.

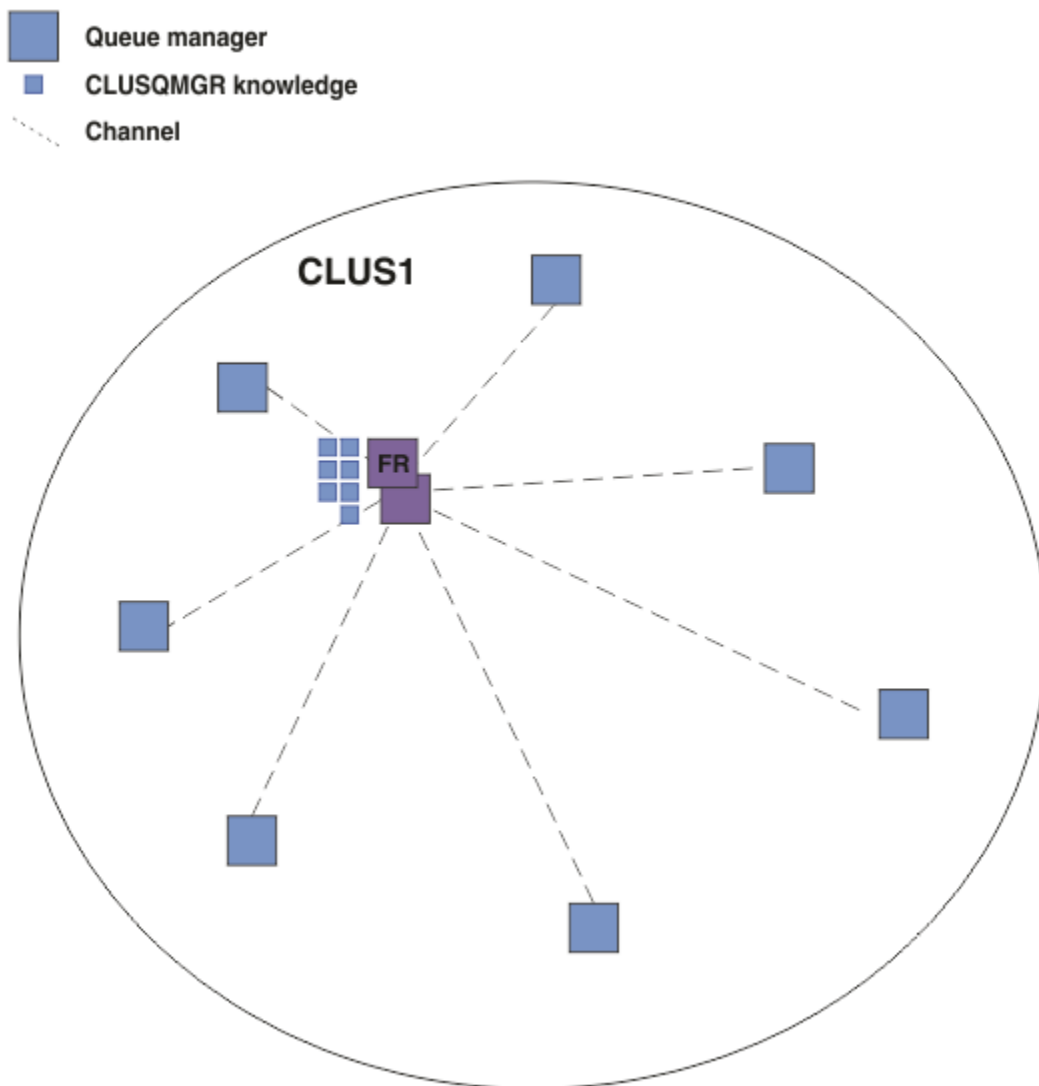


Figura 16. Um cluster de gerenciador de filas

Para que as publicações fluam entre os gerenciadores de filas em um cluster encaminhado diretamente, agrupe uma ramificação da árvore de tópicos como descrito em [Configurando um cluster de publicação/assinatura](#) e especifique *roteamento direto* (o padrão).

Em um cluster de publicação/assinatura encaminhado diretamente, você define o objeto do tópico em qualquer gerenciador de filas no cluster. Ao fazer isso, o conhecimento do objeto e o conhecimento de todos os outros gerenciadores de filas no cluster será automaticamente enviado para todos os gerenciadores de filas no cluster pelos gerenciadores de filas de repositório completo. Isso acontece antes de qualquer gerenciador de filas fazer referência ao tópico:

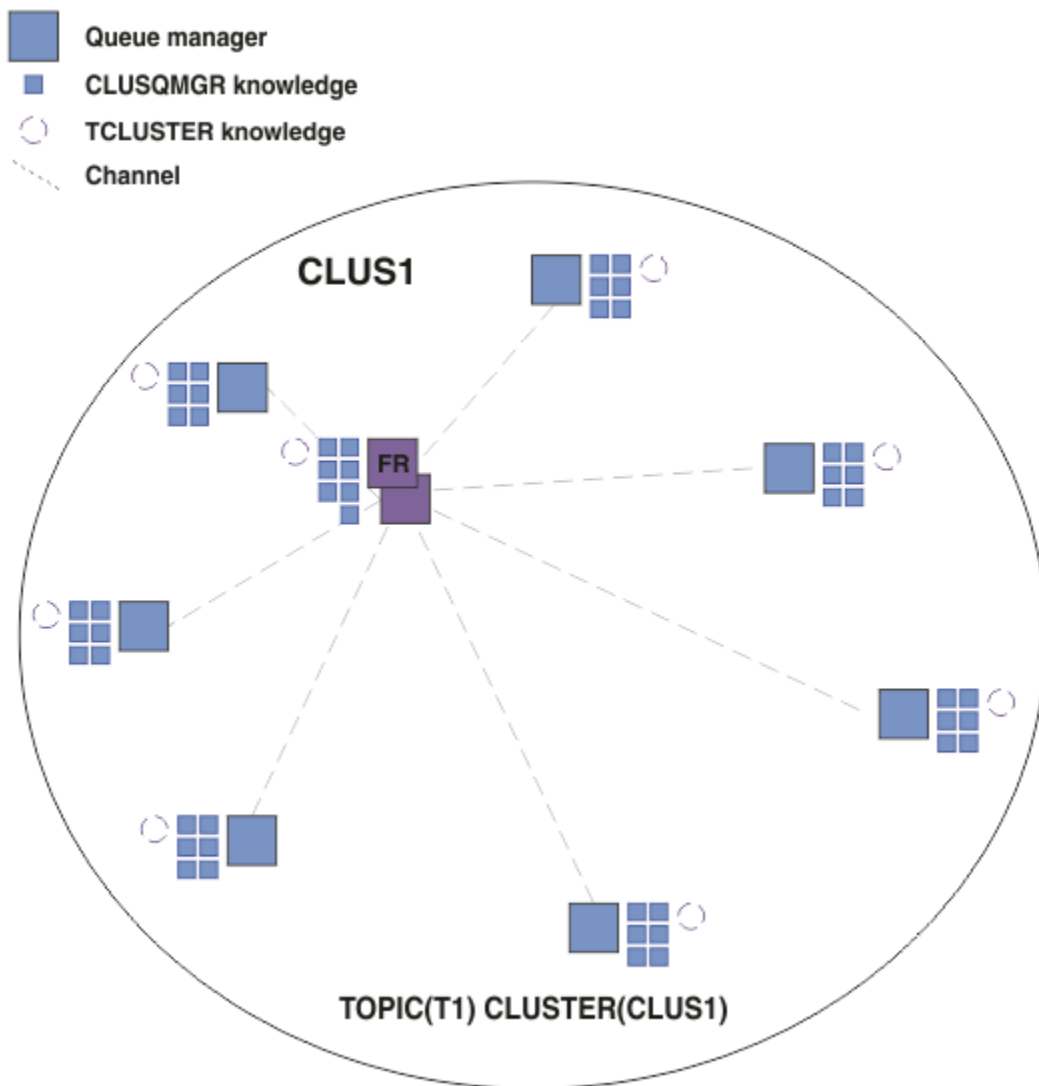


Figura 17. Um cluster de publicação/assinatura encaminhado diretamente

Quando uma assinatura é criada, o gerenciador de filas que hospeda a assinatura estabelece um canal para cada gerenciador de filas no cluster e envia os detalhes da assinatura. Esse conhecimento de assinatura distribuído é representado por uma assinatura de proxy em cada gerenciador de filas. Quando uma publicação é produzida em qualquer gerenciador de filas no cluster que corresponda à sequência de tópicos dessa assinatura de proxy, um canal de cluster é estabelecido a partir do gerenciador de filas do publicador para cada gerenciador de filas que hospeda uma assinatura e a mensagem é enviada para cada um deles.

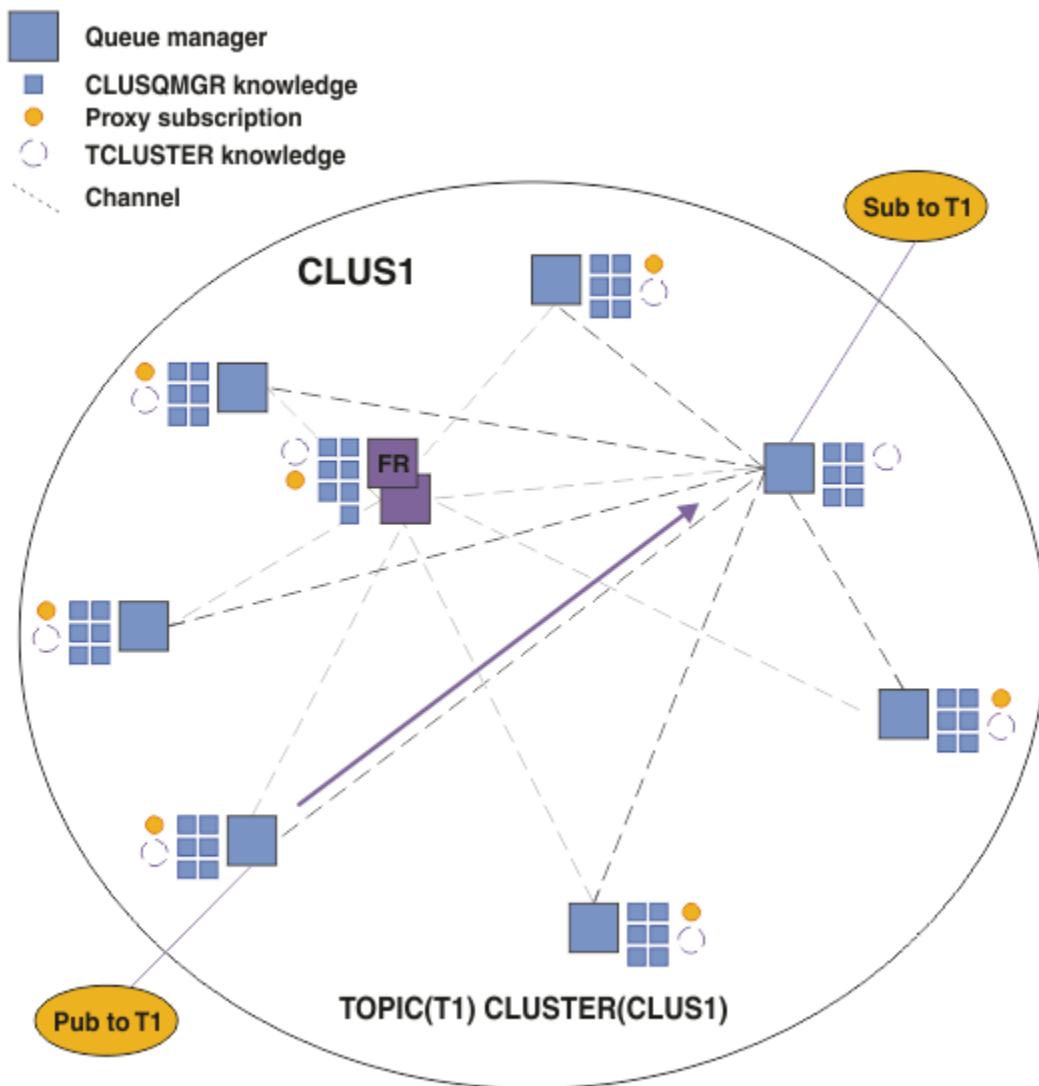


Figura 18. Um cluster de publicação/assinatura encaminhado diretamente com um publicador e um assinante para um tópico em cluster

O roteamento direto de publicações para gerenciadores de filas hospedando assinaturas simplifica a configuração e reduz a latência de entrega de publicações para assinaturas.

No entanto, dependendo do local de assinaturas e os publicadores, o cluster pode rapidamente tornar-se totalmente interconectado, com cada gerenciador de filas que tenha uma conexão direta a cada outro gerenciador de filas. Isso pode ou não ser aceitável em seu ambiente. Da mesma forma, se o conjunto de cadeias do tópico que está sendo inscrito estiver mudando com frequência, a sobrecarga da propagação de informações entre todos os gerenciadores de filas também pode se tornar significativa. Todos os gerenciadores de filas em um cluster de publicação/assinatura encaminhado diretamente deve ser capaz de lidar com essas sobrecargas.

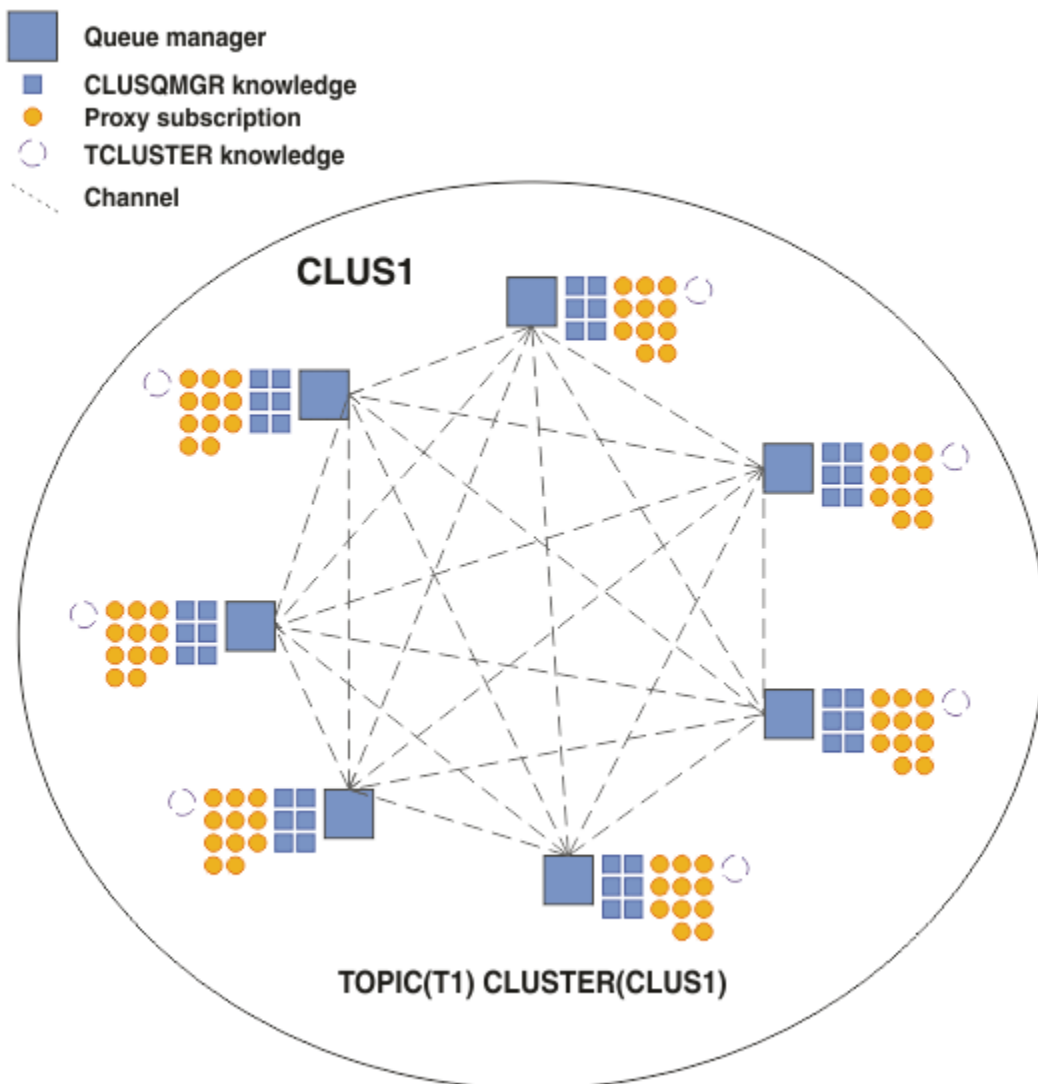


Figura 19. Um cluster de publicação/assinatura encaminhado diretamente totalmente interconectado

Resumo e considerações adicionais

A criação ou administração de um cluster de publicação/assinatura encaminhado diretamente precisa de pouca intervenção manual e fornece roteamento direto entre publicadores e assinantes. Para algumas configurações, geralmente essa é a topologia mais apropriada, principalmente para clusters com poucos gerenciadores de filas ou quando alta conectividade do gerenciador de filas é aceitável e as assinaturas mudam com frequência. No entanto, isso também impõe certas restrições ao seu sistema:

- A carga em cada gerenciador de filas é proporcional ao número total de gerenciadores de filas no cluster. Portanto, em clusters grandes, gerenciadores de filas individuais e o sistema como um todo podem ter problemas de desempenho.
- Por padrão, todas as sequências de tópicos inscritas em cluster são propagadas por todo o cluster, e as publicações são propagadas apenas para gerenciadores de filas remotas que possuem uma assinatura para o tópico associado. Portanto, mudanças rápidas no conjunto de assinaturas podem se tornar um fator limitante. É possível alterar esse comportamento padrão e, em vez disso, ter todas as publicações propagadas para todos os gerenciadores de filas, o que elimina a necessidade de assinaturas de proxy. Isso reduz o tráfego de conhecimento de assinatura, mas é provável que aumente o tráfego de publicações e o número de canais que cada gerenciador de filas estabelece. Consulte [Desempenho de assinatura em redes publicar/assinar](#).

Nota: Uma restrição semelhante também se aplica a hierarquias.

- Devido à natureza interconectada dos gerenciadores de filas de publicação/assinatura, leva tempo para que as assinaturas de proxy se propaguem em todos os nós na rede. Publicações remotas não necessariamente começam a ser assinadas imediatamente, assim, publicações iniciais podem não ser enviadas após uma assinatura de uma nova cadeia de tópicos. É possível remover os problemas causados pelo atraso de assinatura propagando todas as publicações para todos os gerenciadores de filas, o que elimina a necessidade de assinaturas de proxy. Consulte [Desempenho de assinatura em redes publicar/assinar](#).

Nota: Essa restrição também se aplica a hierarquias.

Antes de usar o roteamento direto, explore as abordagens alternativas detalhadas em [“Roteamento de host do tópico em clusters de publicação/assinatura”](#) na página 84 e [“Roteamento em hierarquias de publicação/assinatura”](#) na página 109.

Roteamento de host do tópico em clusters de publicação/assinatura

Publicações de gerenciadores de filas não de hosting no cluster são roteadas através do gerenciador de filas de hosting para qualquer gerenciador de filas no cluster com uma assinatura correspondente.

Para uma introdução de como as mensagens são encaminhadas entre os gerenciadores de filas em hierarquias de publicação/subscrição e clusters, consulte [Redes de publicação/assinatura distribuídas](#).

Para entender o comportamento e os benefícios do roteamento de host do tópico, é melhor primeiro entender [“Roteamento direto em clusters de publicar/assinar”](#) na página 79.

Um cluster de publicação/assinatura encaminhado para o host do tópico comporta-se da maneira a seguir:

- Objetos de tópico administrados em cluster são definidos manualmente em gerenciadores de filas individuais no cluster. Eles são chamados de *gerenciadores de filas de host do tópico*.
- Quando uma assinatura é feita em um gerenciador de filas do cluster, os canais criados a partir do gerenciador de filas do host de assinatura para os gerenciadores de filas do host do tópico e assinaturas de proxy são criadas apenas nos gerenciadores de filas que hospedam o tópico.
- Quando um aplicativo publica informações para um tópico, o gerenciador de filas conectado sempre redireciona a publicação para um gerenciador de filas que hospeda o tópico, que a transmite para todos os gerenciadores de filas no cluster que têm assinaturas correspondentes para o tópico.

Esse processo é explicado com mais detalhes nos exemplos a seguir.

Roteamento do host de tópico usando um host de tópico único

Para que as publicações fluam entre os gerenciadores de filas em um cluster roteado para host de tópico, você agrupa uma ramificação da árvore de tópicos, conforme descrito em [Configurando um cluster de publicação/assinatura](#) e especifica *roteamento de host de tópico*.

Há várias razões para definir um objeto de tópico roteado por host em diversos gerenciadores de filas em um cluster. No entanto, para simplificar vamos começar com um host de tópico único.

O diagrama a seguir mostra um cluster gerenciador de filas que não é usado atualmente para atividades de publicação/assinatura ou ponto a ponto. Observe que cada gerenciador de filas no cluster se conecta apenas para e dos gerenciadores de filas de repositório completo.

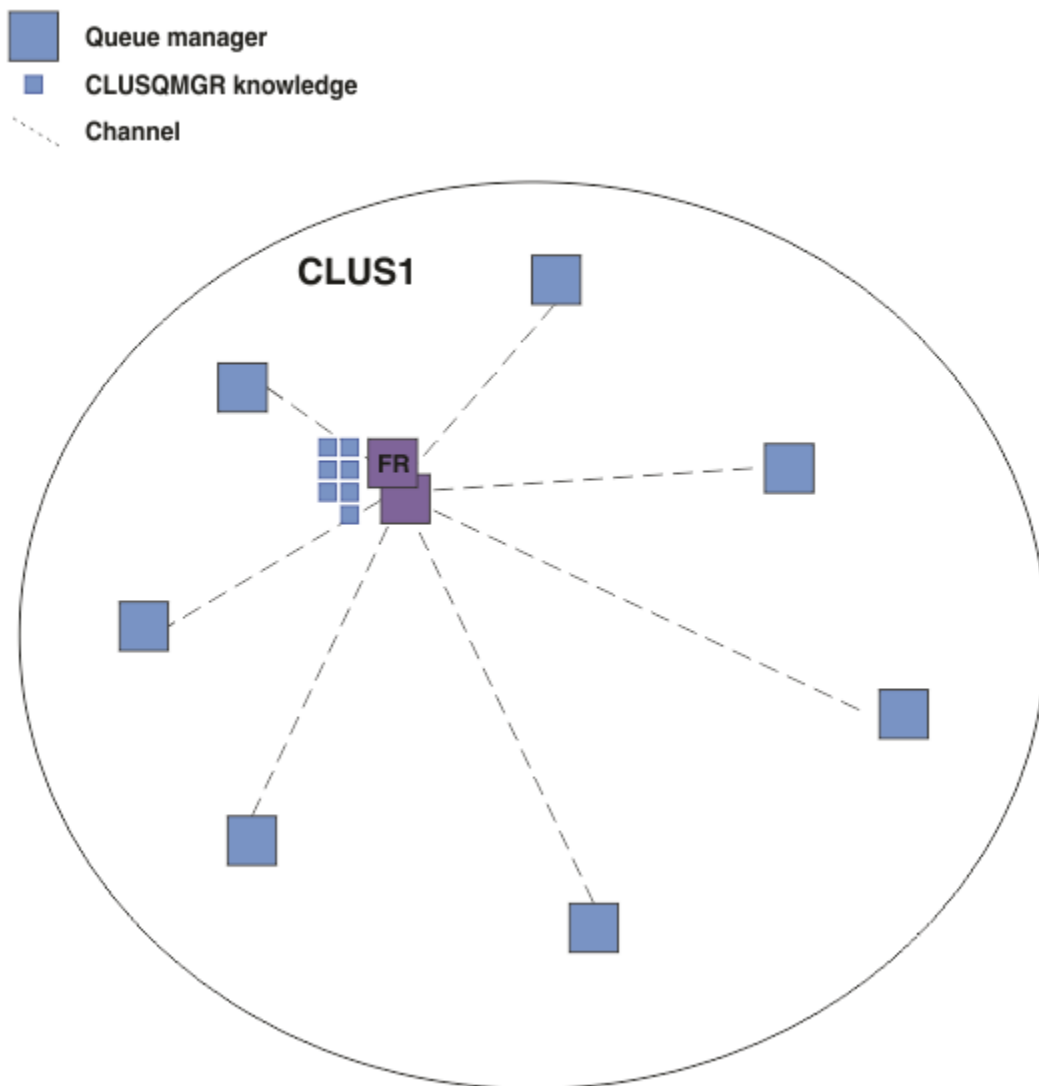


Figura 20. Um cluster de gerenciador de filas

Em um cluster de publicação/assinatura roteado por host, você define o objeto do tópico em um gerenciador de filas específico no cluster. O tráfego de publicação/assinatura então flui através desse gerenciador de filas, tornando-o um gerenciador de filas crítico no cluster e aumentando a sua carga de trabalho. Por essas razões, não é recomendado utilizar um gerenciador de filas de repositório completo, mas usar outro gerenciador de filas no cluster. Ao definir o objeto do tópico no gerenciador de filas do host, o conhecimento do objeto e seu host é automaticamente enviado, pelos gerenciadores de filas de repositório completo, a todos os outros gerenciadores de filas no cluster. Observe que, diferentemente do *roteamento direto*, cada gerenciador de filas não é informado sobre cada outro gerenciador de filas no cluster.

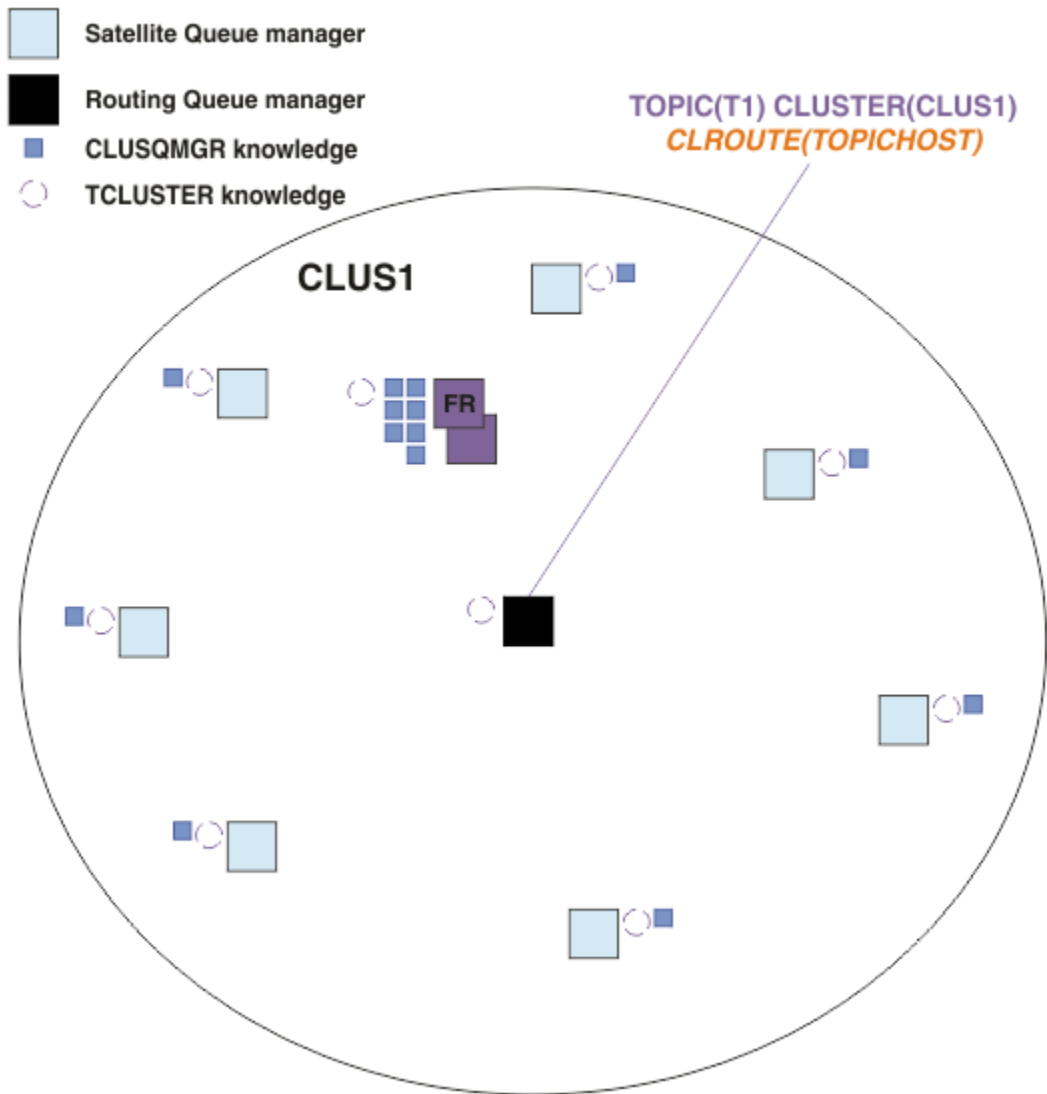


Figura 21. Um publicar/assinar cluster roteado por host de tópico com um tópico definido em um tópico do host

Quando uma assinatura é criada em um gerenciador de filas, um canal é criado entre o gerenciador de filas de assinatura e o gerenciador de filas do host do tópico. O gerenciador de filas de assinatura se conecta somente ao gerenciador de filas do host de tópico e envia detalhes da assinatura (na forma de uma *assinatura de proxy*). O gerenciador de filas do host de tópico não encaminha essas informações de assinatura para mais nenhum gerenciador de fila no cluster.

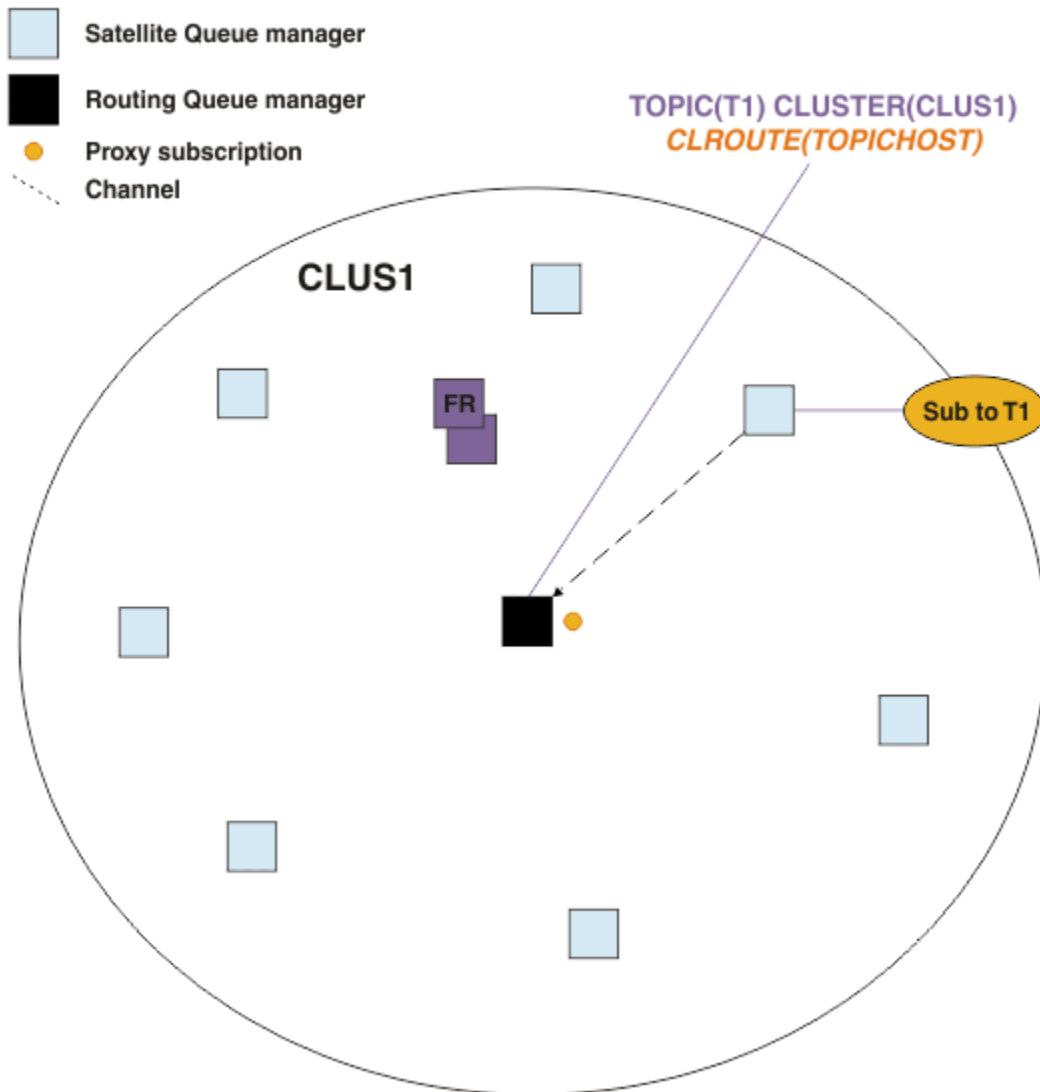


Figura 22. Um cluster de publicação/assinatura roteado para host de tópico com um tópico definido em um tópico do host e um assinante

Quando um aplicativo de publicação se conecta a outro gerenciador de filas e uma mensagem é publicada, um canal é criado entre o gerenciador de filas de publicação e o gerenciador de filas de host do tópico, e a mensagem é redirecionada para esse gerenciador de filas. O gerenciador de filas de publicação não tem conhecimento de quaisquer assinaturas em outros gerenciadores de filas no cluster, portanto, a mensagem é redirecionada para o gerenciador de filas de host de tópico, mesmo se não houver assinantes para esse tópico no cluster. O gerenciador de filas de publicação se conecta somente ao gerenciador de filas do host de tópico. As publicações são roteadas por meio do host do tópico para os gerenciadores de filas de assinatura, se existir algum.

As assinaturas no mesmo gerenciador de filas que o publicador são atendidas diretamente, sem primeiro enviar as mensagens a um gerenciador de filas do host do tópico.

Observe que, devido ao papel fundamental desempenhado por cada gerenciador de filas do host de tópico, é preciso escolher gerenciadores de filas que possam lidar com os requisitos de carga, disponibilidade e conectividade do tópico de hosting.

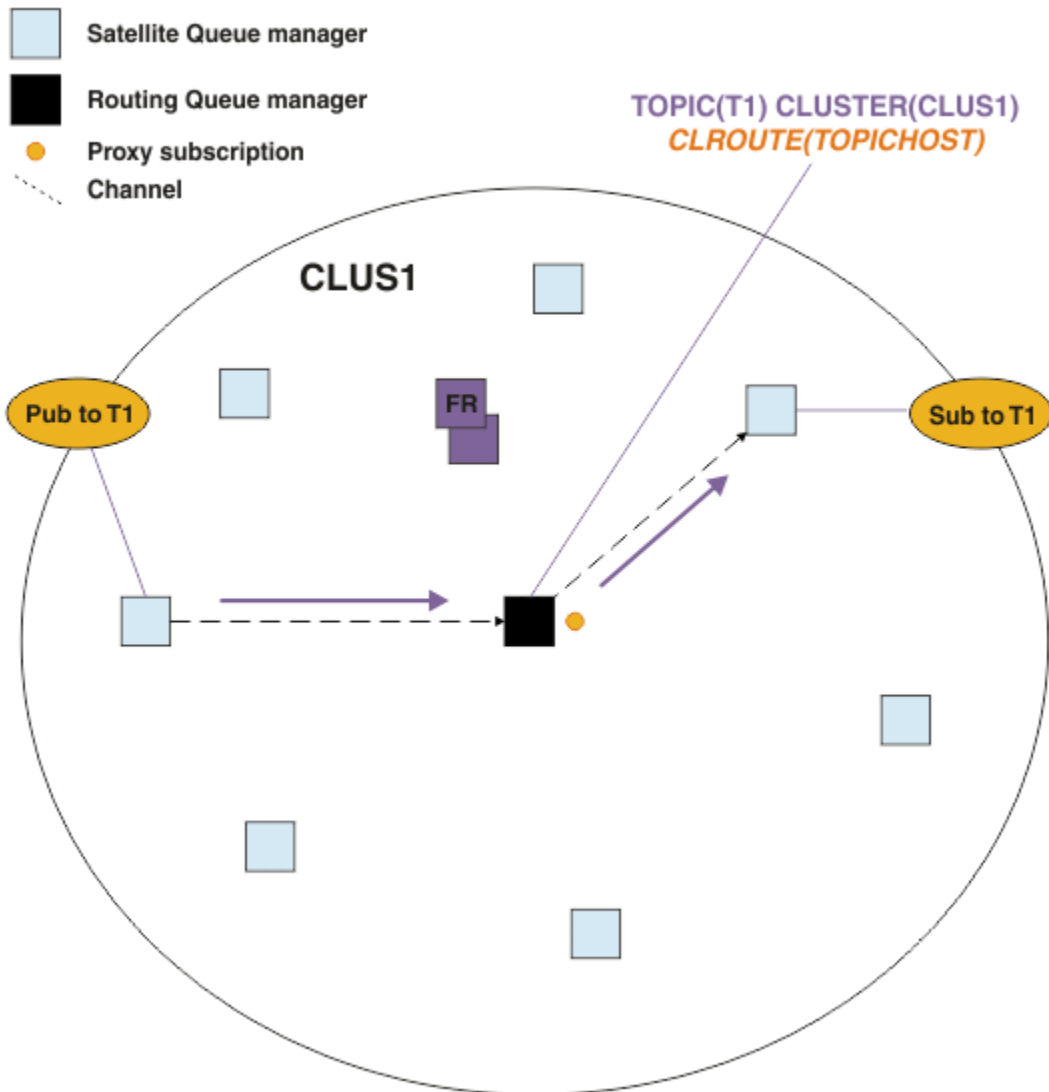


Figura 23. Um cluster de publicação/assinatura roteado por host de tópico com um tópico, um assinante e um publicador

Dividindo a árvore de tópicos entre vários gerenciadores de filas

Um gerenciador de fila de hosting de tópico roteado é responsável apenas pelas mensagens de publicação e conhecimento de assinatura relacionadas à ramificação da árvore de tópicos para a qual o objeto de tópico administrado está configurado. Se tópicos diferentes forem usados por diferentes aplicativos de publicar/assinar no cluster, será possível configurar gerenciadores de filas diferentes para hospedar diferentes ramificações de cluster da árvore de tópicos. Isso permite escalar reduzindo o tráfego de publicação, conhecimento de assinatura e canais em cada gerenciador de filas de host de tópico no cluster. É necessário usar esse método para diferentes ramificações de alto volume da árvore de tópicos:

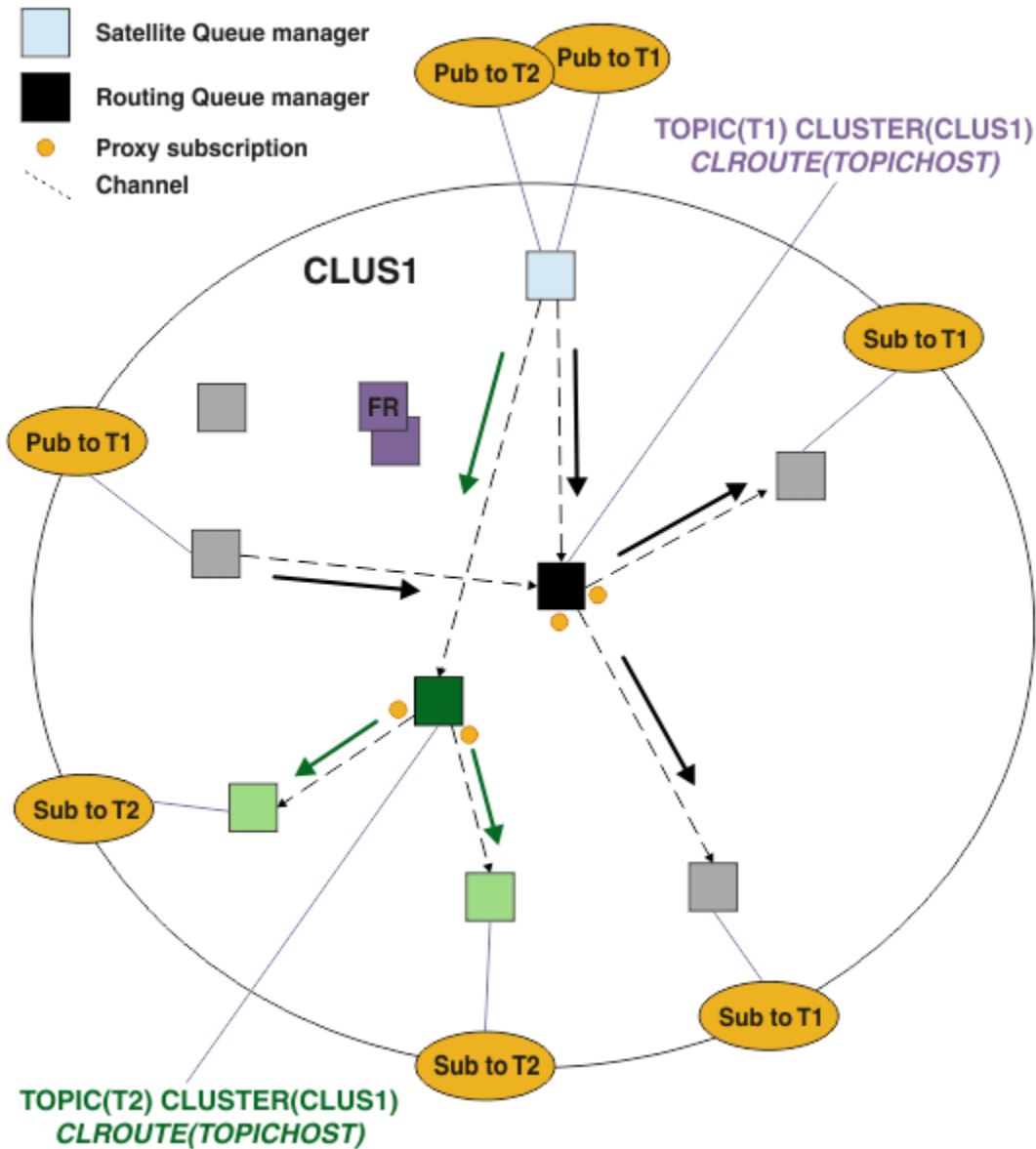


Figura 24. Um cluster de publicação/assinatura roteado por host de tópico com dois tópicos, cada um definido em um host de tópico

Por exemplo, usando os tópicos descritos em [Árvores de tópico](#), se o tópico T1 foi configurado com uma sequência de tópico /USA/Alabama e o tópico T2 foi configurado com uma sequência de tópico /USA/Alaska, então, uma mensagem publicada para /USA/Alabama/Mobile seria roteada através da hospedagem do gerenciador de filas T1 e uma mensagem publicada para /USA/Alaska/Juneau seria roteada através da hospedagem do gerenciador de filas T2.

Nota: Não é possível fazer uma única assinatura abranger várias ramificações de cluster da árvore de tópicos usando um curinga mais alto na árvore de tópicos do que os pontos que estão armazenados em cluster. Consulte [Assinaturas de curinga](#).

Roteamento de host do tópico usando vários hosts de tópico para um único tópico

Se um único gerenciador de filas for responsável pelo roteamento de um tópico, e esse gerenciador de filas ficar indisponível ou não puder processar a carga de trabalho, as publicações não fluirão imediatamente para as assinaturas.

Se você precisar de mais resiliência, escalabilidade e balanceamento de carga do que obtém ao definir um tópico em apenas um gerenciador de filas, pode definir um tópico em mais de um gerenciador de filas.

Cada mensagem individual publicada é roteada através de um único host de tópico. Quando existirem várias definições de host de tópico correspondentes, um dos hosts de tópico será escolhido. A escolha será feita da mesma maneira que para filas em cluster. Isso permite que as mensagens sejam roteadas para hosts de tópico disponíveis, evitando qualquer um que esteja indisponível e permitindo balancear a carga de mensagem entre vários gerenciadores de filas de host de tópico e canais. No entanto, a ordenação entre várias mensagens não é mantida quando se usam vários hosts de tópico para o mesmo tópico no cluster.

O diagrama a seguir mostra um cluster roteado do host do tópico no qual o mesmo tópico foi definido em dois gerenciadores de filas. Neste exemplo, os gerenciadores de filas de assinatura enviam informações sobre o tópico inscrito para ambos os gerenciadores de filas de host de tópico na forma de uma assinatura de proxy:

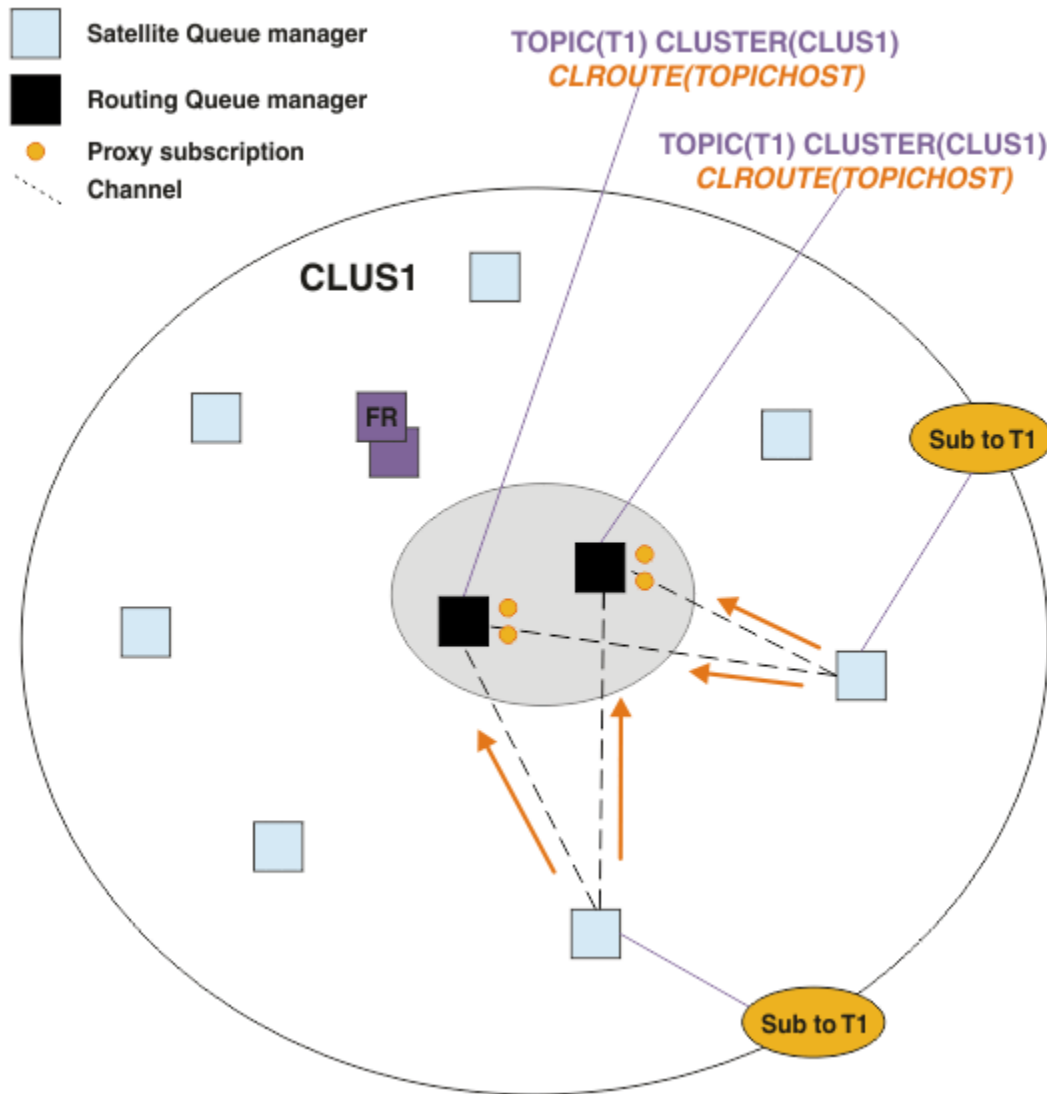


Figura 25. Criando assinaturas de proxy em um cluster de publicação/assinatura de vários hosts de tópico

Quando uma publicação é feita a partir de um gerenciador de filas não hosting, o gerenciador de filas envia uma cópia da publicação para *um* dos gerenciadores de filas de host de tópico para aquele tópico. O sistema escolhe o host com base no comportamento padrão do algoritmo de gerenciamento de carga de trabalho do cluster. Em um sistema típico, isso se aproxima de uma distribuição round-robin em cada gerenciador de filas de host de tópico. Não há afinidade entre as mensagens do mesmo aplicativo de publicação; isso equivale a usar um tipo de ligação do cluster de NOTFIXED.

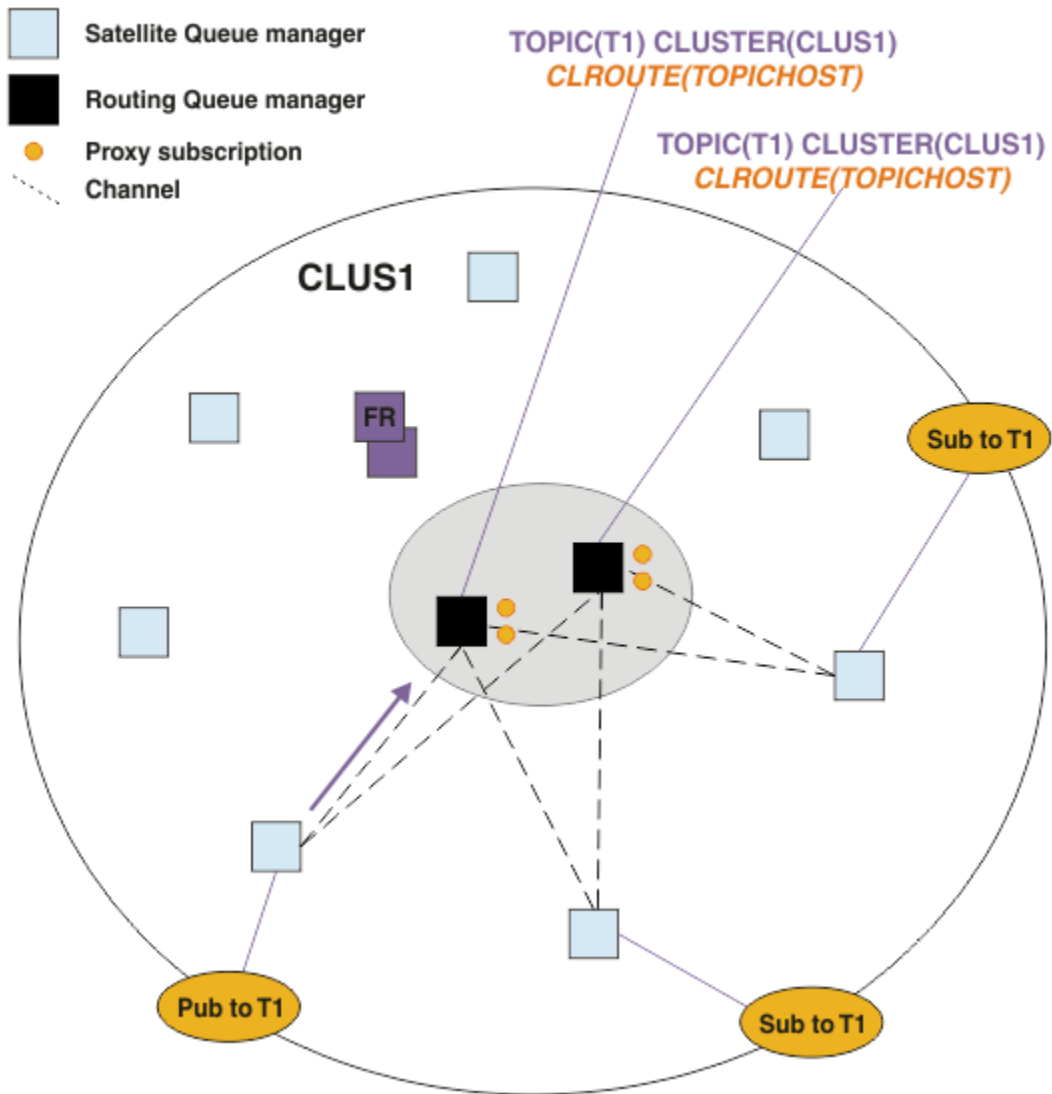


Figura 26. Recebendo publicações em um cluster de publicação/assinatura de hosts de vários tópicos

As publicações de entrada para o gerenciador de filas de host de tópico escolhido então são encaminhada para todos os gerenciadores de filas que registraram uma assinatura de proxy correspondente:

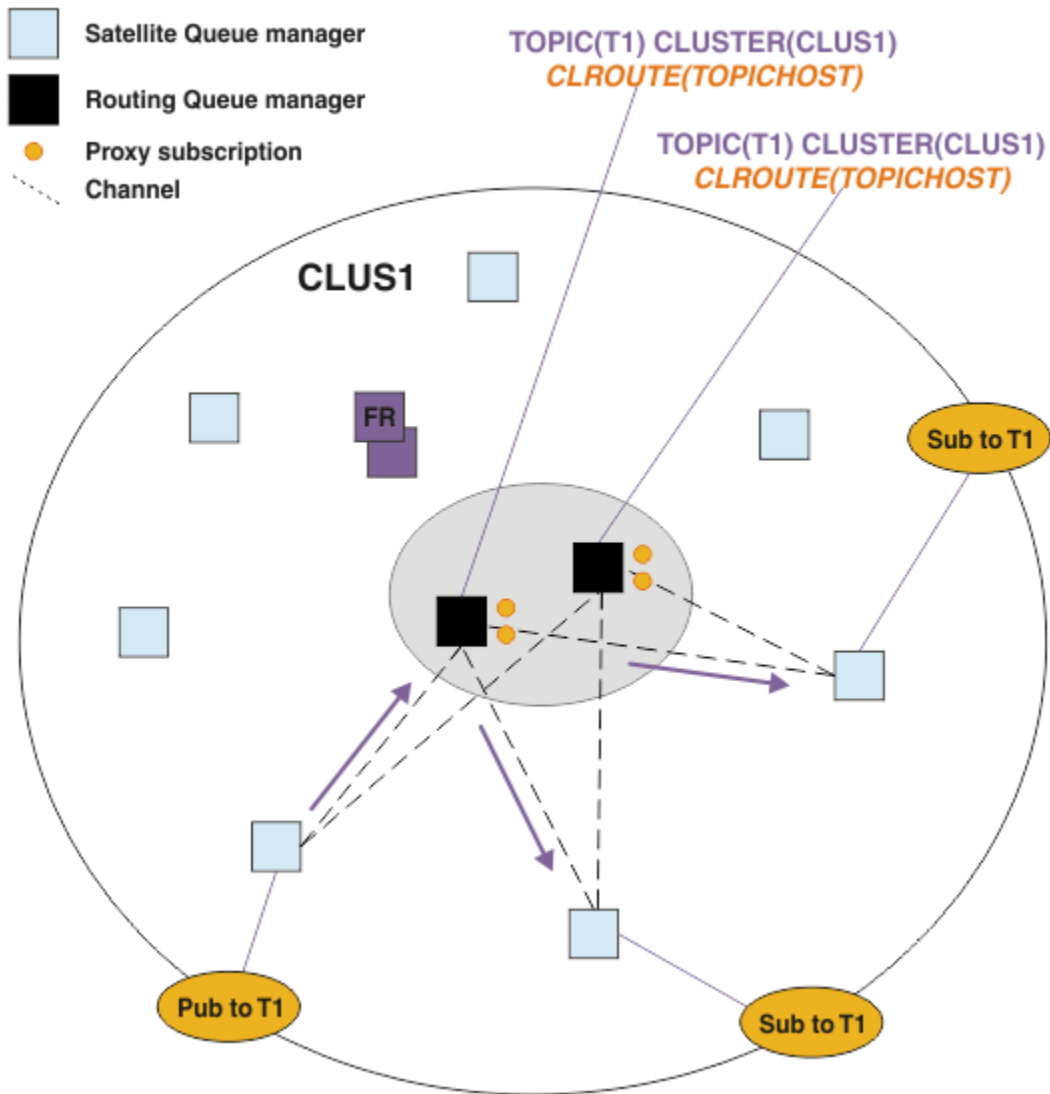


Figura 27. Roteando publicações para assinantes em um cluster de publicação/assinatura de hosts de vários tópicos

Tornando assinaturas e publicadores locais para um gerenciador de filas do host do tópico

Os exemplos acima mostram o roteamento entre os publicadores e os assinantes em gerenciadores de filas que não hospedam objetos do tópico roteados administrados. Nessas topologias, mensagens requerem diversos *hops* para atingir as assinaturas.

Quando o salto adicional não é desejável, pode ser apropriado conectar a publicadores de chave a tópicos hospedando gerenciadores de filas. No entanto, se houver vários hosts de tópico para um tópico e apenas um publicador, todo o tráfego de publicação será roteado através do gerenciador de filas de host de tópico ao qual o publicador está conectado.

Da mesma forma, se houver assinaturas de chave, elas poderão estar localizadas em um gerenciador de filas do host do tópico. No entanto, se houver múltiplos hosts do tópico roteado, apenas uma parte das publicações evitará a *hop* adicional, com o restante sendo roteado através dos outros gerenciadores de filas de host de tópico primeiro.

Topologias como essas são descritas melhor aqui: [Roteamento de host de tópico usando publicadores ou assinantes centralizados.](#)

Nota: Planejamento especial é necessário ao alterar a configuração de tópico roteado ao colocar publicadores ou assinaturas com hosts de tópico roteados. Por exemplo, consulte [Incluindo hosts de tópico adicionais a um cluster roteado para host de tópico](#).

Resumo e considerações adicionais

Um cluster de publicação/assinatura roteado por host de tópico dá controle preciso sobre quais gerenciadores de filas hospedam cada tópico, e esses gerenciadores de filas se tornam os gerenciadores de filas de *roteamento* para essa ramificação da árvore de tópicos. Além disso, os gerenciadores de filas sem assinaturas ou publicadores não precisam se conectar aos gerenciadores de filas de host de tópico, e gerenciadores de filas com assinaturas não precisam se conectar a gerenciadores de filas que não hospedem um tópico. Essa configuração pode reduzir significativamente o número de conexões entre os gerenciadores de fila no cluster e a quantidade de informações que está sendo enviada entre os gerenciadores de fila. Isso é especialmente verdadeiro em clusters grandes em que apenas um subconjunto de gerenciadores de filas está executando trabalho de publicação/assinatura. Essa configuração também dá a você algum controle sobre a carga nos gerenciadores de filas individuais no cluster, portanto (por exemplo) você pode escolher hospedar tópicos altamente ativos em sistemas mais potentes e mais resistentes. Para algumas configurações, especialmente clusters maiores, geralmente é uma topologia mais apropriada do que *roteamento direto*.

No entanto, roteamento de host de tópico também impõe algumas restrições em seu sistema:

- Configuração e manutenção do sistema requerem mais planejamento do que roteamento direto. Você precisa decidir quais pontos para o cluster na árvore de tópicos e o local das definições de tópico no cluster.
- Assim como para direcionar os tópicos roteados, quando um novo tópico roteado por host de tópico é definido, as informações são enviadas para os gerenciadores de filas de repositório completo e a partir daí direto para todos os membros do cluster. Este evento faz com que canais sejam iniciados para cada membro do cluster a partir dos repositórios completos se ainda não estiver iniciado.
- Publicações são sempre enviadas para um gerenciador de filas do host a partir de um gerenciador de filas que não seja de host, mesmo se não houver assinaturas no cluster. Portanto, você deve utilizar os tópicos roteados quando as assinaturas são normalmente esperadas, ou quando o código extra da conectividade global e conhecimento é maior do que o risco do tráfego de publicação extra.

Nota: Conforme descrito anteriormente, tornar publicadores locais para um host de tópico pode minimizar esse risco.

- As mensagens que são publicadas nos gerenciadores de filas que não são de host não vão direto para o gerenciador de filas que hospeda a assinatura, elas são sempre roteadas por um gerenciador de filas do host de tópicos. Essa abordagem pode aumentar a sobrecarga total para o cluster, e aumentar a latência da mensagem e reduzir o desempenho.

Nota: Conforme descrito anteriormente, tornar assinaturas ou publicadores locais para um host de tópico pode minimizar esse risco.

- Usar um gerenciador de filas de host de tópico único apresenta um único ponto de falha para todas as mensagens que são publicadas para um tópico. É possível remover este ponto único de falha, definindo vários hosts de tópico. No entanto, ter vários hosts afeta a ordem de mensagens publicadas como recebidas por assinaturas.
- Carregamento de mensagens extras é suportado pelos gerenciadores de filas de host de tópico, porque o tráfego de publicação a partir de vários gerenciadores de filas precisa ser processado por eles. Este carregamento pode ser reduzido: use vários hosts de tópico para um único tópico (no qual a ordem da mensagem de caso não é mantida), ou utilize gerenciadores de filas diferentes para hospedar tópicos roteados para diferentes ramificações da árvore de tópicos.

Antes de utilizar o roteamento de host de tópico, explore as abordagens alternativas detalhadas em [“Roteamento direto em clusters de publicar/assinar”](#) na página 79 e [“Roteamento em hierarquias de publicação/assinatura”](#) na página 109.

Publicar/assinar cluster: melhores práticas

Usar tópicos em cluster torna simples a extensão do domínio de publicar/assinar entre gerenciadores de fila, mas pode levar a problemas se a mecânica e as implicações não forem totalmente compreendidas. Há dois modelos para compartilhar informações e rotear publicações. Implemente o modelo que melhor atenda às suas necessidades de negócios e tenha o melhor desempenho no seu cluster escolhido.

As informações de melhor prática nas seções a seguir não fornecem uma solução padronizada, mas, em vez disso, compartilha abordagens comuns à resolução de problemas comuns. Ele assume que você tem um entendimento básico de clusters do IBM MQ e de mensagens de publicação/assinatura, e que você está familiarizado com as informações em [Redes de publicar/assinar distribuídas](#) e [“Projetando clusters publicar/assinar”](#) na página 77.

Ao usar um cluster para o sistema de mensagens ponto a ponto, cada gerenciador de filas no cluster funciona conforme a necessidade de conhecimento. Ou seja, apenas toma conhecimento de outros recursos do cluster, como gerenciadores de fila em cluster e filas em cluster, quando os aplicativos que estão se conectando a eles solicitam o seu uso. Quando você inclui um sistema de mensagens de publicar/assinar a um cluster, um nível maior de compartilhamento de informações e conectividade entre os gerenciadores de fila do cluster é introduzido. Para poder seguir melhores práticas para clusters de publicar/assinar, você precisa para compreender totalmente as implicações dessa mudança de comportamento.

Para permitir que você construir a melhor arquitetura, com base em suas necessidades específicas, há dois modelos para compartilhar informações e rotear publicação em clusters de publicar/assinar: *roteamento direto* e *roteamento de host de tópico*. Para fazer a escolha certa, é necessário compreender ambos os modelos e os diferentes requisitos a que cada modelo atende. Esses requisitos são abordados nas seções a seguir, em conjunto com [“Planejando sua rede publicar/assinar distribuída”](#) na página 73:

- [“Razões para limitar o número de gerenciadores de filas do cluster envolvido na atividade de publicar/assinar”](#) na página 94
- [“Como decidir quais tópicos no cluster”](#) na página 95
- [“Como dimensionar seu sistema”](#) na página 95
- [“Local do publicador e de assinatura”](#) na página 96
- [“Tráfego de publicação”](#) na página 97
- [“Mudança da assinatura e sequências de tópicos dinâmicas”](#) na página 97

Razões para limitar o número de gerenciadores de filas do cluster envolvido na atividade de publicar/assinar

Há considerações de capacidade e desempenho ao utilizar o sistema de mensagens de publicar/assinar em um cluster. Portanto, é melhor prática ponderar cuidadosamente a necessidade da atividade de publicar/assinar entre gerenciadores de filas e para limitar-se a somente o número de gerenciadores de fila que a requerem. Após o conjunto mínimo de gerenciadores de filas que precisa publicar e assinar tópicos ser identificado, eles podem se tornar membros de um cluster que contenha somente eles e nenhum outro gerenciador de filas.

Essa abordagem é especialmente útil se você tiver um cluster já estabelecido que funciona bem para o sistema de mensagens ponto a ponto. Quando você estiver ativando um cluster grande existente para um cluster de publicação/assinatura, é melhor prática criar inicialmente um cluster separado para o trabalho de publicar/assinar no qual seja possível testar os aplicativos, em vez de usar o cluster atual. É possível utilizar um subconjunto de gerenciadores de filas existentes que já estão em um ou mais clusters ponto a ponto e tornar esse subconjunto membro do novo cluster de publicação/assinatura. No entanto, os gerenciadores de filas de repositório completos para seu novo cluster não deverão ser membros de nenhum outro cluster; isso isola a carga adicional dos repositórios completos de cluster existentes.

Se não for possível criar um novo cluster, e precisar transformar um cluster grande existente em um cluster de publicação/assinatura, não use um modelo encaminhado direto. O modelo roteado de host de tópico geralmente é mais bem executado em clusters maiores, pois ele geralmente restringe o compartilhamento de informações de publicar/assinar e a conectividade com o conjunto de

gerenciadores de filas que estão executando ativamente trabalhos de publicar/assinar, concentrando-nos gerenciadores de filas que hospedam os tópicos. A exceção a isso é se uma atualização manual das informações de assinatura for chamado em um gerenciador de filas que hospede uma definição de tópico, ponto em que o gerenciador de filas de host de tópico irá se conectar a cada gerenciador de filas no cluster. Veja [Ressincronização de assinaturas de proxy](#).

Se você estabelecer que um cluster não pode ser usado para publicação/assinatura devido ao seu tamanho ou à sua carga atual, é boa prática evitar que esse cluster inesperadamente seja transformado em um cluster de publicação/assinatura. Use a propriedade **PSCLUS** para impedir que qualquer pessoa inclua um tópico em cluster em qualquer gerenciador de filas no cluster. Consulte o ["Inibindo publicação/assinatura em cluster"](#) na página 104.

Como decidir quais tópicos no cluster

É importante escolher atentamente quais tópicos são incluídos no cluster: quanto mais alta for essa árvore de tópicos, mais disseminado seu uso será. Isso pode resultar em propagar mais informações e publicações de assinatura do que o necessário. Se houver várias ramificações distintas da árvore de tópicos, em que algumas precisem ser armazenadas em cluster e outras não, crie objetos de tópico administrados na raiz de cada ramificação que precise de clusterização e inclua-os no cluster. Por exemplo, se as ramificações /A, /B e /C precisarem de armazenamento em cluster, defina objetos de tópico em cluster separados para cada ramificação.

Nota: O sistema impede que você aninhe definições de tópico em cluster na árvore de tópicos. Você tem permissão apenas para clusterizar tópicos em um ponto na árvore de tópicos para cada sub-ramificação. Por exemplo, não é possível definir objetos de tópicos em cluster para /A e para /A/B. O aninhamento de tópicos armazenados em cluster pode causar confusão sobre qual objeto armazenado em cluster se aplica a qual assinatura, especialmente quando as assinaturas usam caracteres curingas. Isso é ainda mais importante ao usar o roteamento de host de tópico, em que as decisões de roteamento são precisamente definidas pela sua alocação dos hosts de tópico.

Se os tópicos em cluster precisarem ser incluídos no topo da árvore de tópicos, mas algumas ramificações da árvore abaixo do ponto em cluster não exigirem o comportamento clusterizado, é possível usar os atributos escopo de assinatura e publicação para reduzir o nível de compartilhamento de assinatura e publicação para mais tópicos.

Não se deve colocar o nó raiz de tópico no cluster sem considerar o comportamento que é visto. Torne os tópicos globais óbvios quando possível, por exemplo, usando um qualificador de alto nível na sequência de tópicos: /global ou /cluster.

Há mais uma razão para não querer colocar o nó do tópico raiz em cluster. Essa razão é que cada gerenciador de filas possui uma definição local para o nó raiz, o objeto do tópico SYSTEM.BASE.TOPIC. Quando esse objeto é armazenado em cluster em um gerenciador de filas no cluster, todos os outros gerenciadores de filas são informados disso. No entanto, quando existe uma definição local do mesmo objeto, suas propriedades substituem o objeto de cluster. Isso resulta em esses gerenciadores de filas atuarem como se o tópico não estivesse em cluster. Para resolver isso, seria necessário colocar todas as definições de SYSTEM.BASE.TOPIC em cluster. Seria possível fazer isso para as definições roteadas diretamente, mas não para as definições roteadas por host de tópico, porque elas fazem com que cada gerenciador de filas se torne um host de tópico.

Como dimensionar seu sistema

Clusters de publicação/assinatura geralmente resultam em um padrão diferente de canais de cluster para o sistema de mensagens ponto a ponto em um cluster. O modelo ponto a ponto é de "adesão", mas clusters de publicar/assinar têm um caráter mais indiscriminado com fan-out de assinatura, especialmente ao usar tópicos roteados diretos. Portanto, é importante identificar quais gerenciadores de filas em um cluster de publicação/assinatura usarão os canais de cluster para conexão a outros gerenciadores de filas e sob quais circunstâncias.

A tabela a seguir lista o conjunto típico de canais de emissor e receptor de clusters esperados para cada gerenciador de filas em um cluster de publicação/assinatura sob execução normal, dependendo da função do gerenciador de filas no cluster de publicação/assinatura.

Tabela 5. Canais do emissor e do receptor do cluster para cada método de roteamento.

Função de gerenciador de filas	Receptores de cluster diretos	Emissores de cluster diretos	Receptores de cluster de tópico	Emissores de cluster de tópico
Repositório completo	AllQmgrs	AllQmgrs	AllQmgrs	AllQMGRs
Host de definição de tópico	n/a	n/a	AllSubs+AllPubs (1)	AllSubs (1)
Assinaturas criadas	AllPubs (1)	AllQMGRs	AllHosts	AllHosts
Publicadores conectados	AllSubs (1)	AllSubs (1)	AllHosts	AllHosts
Nenhum publicador ou assinante	AllSubs (1)	None (1)	Nenhum (2)	Nenhum (2)

Tecla:

AllQmgrs

Um canal para e de cada gerenciador de filas no cluster.

AllSubs

Um canal para e de cada gerenciador de filas no qual uma assinatura foi criada.

AllPubs

Um canal para e de cada gerenciador de filas no qual um aplicativo de publicação foi conectado.

AllHosts

Um canal para e de cada gerenciador de filas no qual uma definição do objeto do tópico em cluster foi configurada.

Nenhum

Nenhum canal para ou de outros gerenciadores de filas no cluster para o propósito exclusivo de mensagens de publicar/assinar.

Notas:

1. Se uma atualização do gerenciador de filas de assinaturas de proxy for feita a partir desse gerenciador de filas, um canal para e de todos os outros gerenciadores de filas no cluster poderá ser criado automaticamente.
2. Se uma atualização do gerenciador de filas de assinaturas de proxy for feita a partir desse gerenciador de filas, um canal para e de quaisquer outros gerenciadores de filas no cluster que hospedam uma definição de um tópico em cluster poderá ser criado automaticamente.

A tabela anterior mostra que o roteamento de host de tópico geralmente usa significativamente menos canais emissores e receptores de cluster e que o roteamento direto. Se a conectividade do canal for uma preocupação para determinados gerenciadores de filas em um cluster, por razões de capacidade ou habilidade de estabelecer determinados canais (por exemplo, através de firewalls), o roteamento de host de tópico é, portanto, uma solução preferencial.

Local do publicador e de assinatura

cluster de publicação/assinatura permite que mensagens publicadas em um gerenciador de filas sejam entregues às assinaturas em qualquer outro gerenciador de filas no cluster. Como para o sistema de mensagens ponto a ponto, o custo de transmissão de mensagens entre os gerenciadores de filas pode ser prejudicial para o desempenho. Portanto, deve-se considerar a criação de assinaturas para tópicos nos mesmos gerenciadores de filas em que as mensagens estão sendo publicadas.

Ao usar o roteamento de host de tópico dentro de um cluster, é importante também considerar o local das assinaturas e dos publicadores com relação aos gerenciadores de fila de hosting do tópico. Quando o publicador não está conectado a um gerenciador de filas que é um host do tópico em cluster, as mensagens publicadas são sempre enviadas para um gerenciador de filas de hosting do tópico. Da

mesma forma, quando uma assinatura é criada em um gerenciador de filas que não é um host de tópico para um tópico de cluster, as mensagens publicadas a partir de outros gerenciadores de filas no cluster sempre são enviadas primeiro para um tópico de hosting do gerenciador de filas. Mais especificamente, se a assinatura estiver localizada em um gerenciador de filas de hosting do tópico, mas houver um ou mais outros gerenciadores de filas que também sejam host do mesmo tópico, uma parte das publicações de outros gerenciadores de filas será roteada por meio desses outros gerenciadores de filas de hosting do tópico. Veja [Roteamento de host de tópico usando publicadores ou assinantes centralizados](#) para obter mais informações sobre como projetar um cluster roteado para host de tópico de publicar/assinar para minimizar a distância entre publicadores e assinaturas.

Tráfego de publicação

Mensagens publicadas por um aplicativo conectado a um gerenciador de filas em um cluster são transmitidas para assinaturas em outros gerenciadores usando canais emissores de cluster.

Quando você usa roteamento direto, as mensagens publicadas seguem o caminho mais curto entre gerenciadores de filas. Ou seja, elas vão direto do gerenciador de filas de publicação para cada um dos gerenciadores de filas com assinaturas. As mensagens não são transmitidas para gerenciadores de filas que não tenham assinaturas para o tópico. Veja [Assinaturas de proxy em uma rede de publicação/assinatura](#).

Se a taxa de mensagens de publicação entre qualquer gerenciador de filas um e outro no cluster for alta, a infraestrutura de canal de cluster entre esses dois pontos deverá ser capaz de manter a taxa. Isso pode envolver ajustar os canais e as filas de transmissão que estão em uso.

Ao usar o roteamento de host de tópico, cada mensagem publicada em um gerenciador de filas que não seja um host de tópico é transmitida para um gerenciador de filas do host do tópico. Isso independe de haver uma ou mais assinaturas em qualquer outro lugar no cluster. Isso apresenta outros fatores a considerar no planejamento:

- A latência adicional de primeiro enviar cada publicação para um gerenciador de filas de host de tópico é aceitável?
- Cada gerenciador de filas de host de tópico pode sustentar a taxa de publicação de entrada e de saída? Considere um sistema com muitos publicadores em diferentes gerenciadores de filas. Se todos eles enviarem suas mensagens para um conjunto muito pequeno de gerenciadores de filas que hospedam tópicos, esses hosts de tópico poderão se tornar um gargalo no processamento de essas mensagens e no seu roteamento para gerenciadores de filas de assinatura.
- É esperado que uma parte significativa das mensagens publicadas não tenha um assinante correspondentes? Se sim, e a taxa de publicação dessas mensagens for alta, pode ser melhor tornar o gerenciador de filas do publicador um host de tópico. Nessa situação, qualquer mensagem publicada em que não haja assinaturas no cluster não será transmitida para nenhum outro gerenciador de filas.

Esses problemas também podem ser atenuados introduzindo vários hosts de tópico para distribuir a carga de publicação entre eles:

- Quando houver vários tópicos diferentes, cada um com uma parte do tráfego de publicação, considere hospedá-los em gerenciadores de filas diferentes.
- Se os tópicos não puderem ser separados em hosts de tópico diferentes, considere definir o mesmo objeto do tópico em vários gerenciadores de filas. Isso resulta no balanceamento de carga das publicações em cada um deles para o roteamento. No entanto, isso é apropriado apenas quando não houver necessidade de ordenação da mensagem de publicação.

Mudança da assinatura e sequências de tópicos dinâmicas

Outra consideração é o efeito sobre o desempenho do sistema para propagar as assinaturas de proxy. Geralmente, um gerenciador de filas envia uma mensagem de assinatura de proxy para determinados outros gerenciadores de filas no cluster quando a primeira assinatura para uma sequência de tópico de cluster específica (não apenas um objeto do tópico configurado) é criada em tal gerenciador de filas. Da mesma forma, uma mensagem de exclusão assinatura de proxy é enviada quando a última assinatura para uma sequência de tópico de cluster específica é excluída.

Para roteamento direto, cada gerenciador de filas com assinaturas envia essas assinaturas de proxy para todo os outros gerenciadores de filas no cluster. Para roteamento de host do tópico, cada gerenciador de filas com assinaturas envia apenas as assinaturas de proxy para cada gerenciador de filas que hospeda uma definição para aquele tópico em cluster. Portanto, com roteamento direto, quanto mais gerenciadores de filas há no cluster, maior a sobrecarga de manutenção de assinaturas de proxy entre eles. Considerando isso, com roteamento de host de tópico, o número de gerenciadores de filas no cluster não é um fator.

Em ambos os modelos de roteamento, se uma solução de publicar/assinar consistir em muitas sequências de tópicos exclusivas serem assinados, ou os tópicos em um gerenciador de filas no cluster serem frequentemente sendo assinados e não assinados, uma sobrecarga significativa será observada nesse gerenciador de filas, causada pela constante geração de mensagens distribuindo e excluindo assinaturas de proxy. Com o roteamento direto, isso é agravado ainda pela necessidade de enviar essas mensagens para cada gerenciador de filas no cluster.

Se a taxa de mudança de assinaturas for muito alta para acomodar, mesmo dentro de um sistema roteado por host de tópico, consulte [Desempenho de assinatura em redes publicar/assinar](#) para obter informações sobre formas de reduzir a sobrecarga de assinatura de proxy.

Definindo Tópicos de Cluster

Os tópicos de cluster são tópicos administrativos com o atributo de **cluster** definido. Informações sobre tópicos de cluster são enviadas por push para todos os membros de um cluster, e combinadas com tópicos locais para criar partes de um espaço de tópico que se estende por vários gerenciadores de filas. Isso permite que mensagens publicadas em um tópico em um gerenciador de filas sejam entregues às assinaturas de outros gerenciadores de filas no cluster.

Ao definir um tópico de cluster em um gerenciador de filas, a definição de tópico de cluster é enviada para os gerenciadores de filas do repositório completo. Os repositórios completos, em seguida, propagam a definição de tópico de cluster para todos os gerenciadores de filas no cluster, tornando o mesmo tópico de cluster disponível para publicadores e assinantes em qualquer gerenciador de filas no cluster. O gerenciador de filas no qual você cria um tópico de cluster é conhecido como um host de tópico de cluster. O tópico de cluster pode ser usado por qualquer gerenciador de filas no cluster, mas quaisquer modificações em um tópico de cluster deverão ser feitas no gerenciador de filas no qual esse tópico está definido (o host), nesse ponto a modificação é propagada para todos os membros do cluster por meio dos repositórios completos.

Ao usar o roteamento direto, o local da definição de tópico em cluster não afeta diretamente o comportamento do sistema, porque todos os gerenciadores de filas no cluster usam a definição de tópico da mesma maneira. Portanto, deve-se definir o tópico em qualquer gerenciador de filas que será um membro do cluster pelo tempo que o tópico for necessário, e que esteja em um sistema confiável suficiente para entrar em contato regularmente com gerenciadores de filas de repositório completo.

Ao usar roteamento de host de tópico, o local da definição de tópico em cluster é muito importante, uma vez que outros gerenciadores de fila no cluster criam canais para esse gerenciador de cluster e envia informações de assinatura e publicações a ele. Para escolher o melhor gerenciador de filas para hospedar a definição de tópico, é preciso entender roteamento de host de tópico. Consulte o [“Roteamento de host do tópico em clusters de publicação/assinatura”](#) na página 84.

Se você tiver um tópico em cluster e um objeto do tópico local, o tópico local terá precedência. Consulte o [“Diversas definições de tópico de cluster com o mesmo nome”](#) na página 101.

Para obter informações sobre os comandos a serem usados para exibir tópicos de cluster, consulte as informações relacionadas.

Herança de tópico em cluster

Geralmente, publicação e assinatura de aplicativos em um cluster de publicação/assinatura esperam que a topologia funcione da mesma forma, não importa a qual gerenciador de filas no cluster estão conectados. É por isso que os objetos de tópico em cluster administrado são propagados para cada gerenciador de filas no cluster.

Um objeto do tópico administrado herda seu comportamento de outros objetos de tópico administrados mais alto na árvore de tópicos. Essa herança ocorre quando um valor explícito não foi configurado para um parâmetro de tópico.

No caso de publicação em cluster, é importante considerar herança, porque ela introduz a possibilidade de que publicadores e assinantes comportem-se de forma diferente, dependendo de a qual gerenciador de filas eles se conectam. Se um objeto de tópico em cluster deixar quaisquer parâmetros para herdar de objetos de tópico mais elevados, o tópico poderá se comportar de forma diferente em diferentes gerenciadores de filas no cluster. Da mesma forma, objetos de tópico definidos localmente abaixo um objeto do tópico em cluster na árvore de tópicos ainda estarão em cluster, mas o objeto local poderá mudar seu comportamento de alguma forma que difira de outros gerenciadores de filas no cluster.

Assinaturas Curinga

Assinaturas de proxy são criadas quando as assinaturas locais são feitas para uma cadeia de tópico que é resolvida em, ou abaixo, um objeto de tópico em cluster. Se uma assinatura curinga for feita acima na hierarquia de tópicos de qualquer tópico de cluster, ela não tem assinaturas de proxy enviadas ao redor do cluster para o tópico de cluster correspondente, portanto, não recebe publicações de outros membros do cluster. Porém, ela recebe publicações do gerenciador de filas locais.

No entanto, se outro aplicativo assinar uma sequência de tópicos que resolva para o tópico de cluster ou abaixo dele, as assinaturas de proxy serão geradas e publicações serão propagadas para esse gerenciador de filas. Na chegada, a assinatura curinga original mais elevada é considerada um destinatário legítimo dessas publicações e recebe uma cópia. Se esse comportamento não for necessário, configure **WILDCARD (BLOCK)** no tópico em cluster. Isso faz o curinga original não ser considerado uma assinatura legítima, e ele para de receber quaisquer publicações (locais ou de outro lugar no cluster) sobre o tópico de cluster ou seus subtópicos.

Conceitos relacionados

[Trabalhando com tópicos administrativos](#)

[Trabalhando com assinaturas](#)

Referências relacionadas

[DISPLAYTOPIC](#)

[DISPLAYTPSTATUS](#)

[DISPLAYSUB](#)

Atributos do tópico de cluster

Quando um objeto do tópico possui o conjunto de atributos de nome do cluster, a definição de tópico é propagada entre todos os gerenciadores de filas no cluster. Cada gerenciador de filas usa os atributos do tópico propagados para controlar o comportamento de aplicativos de publicação/assinatura.

Um objeto do tópico tem vários atributos que se aplicam a clusters de publicação/assinatura. Alguns controlam o comportamento geral dos aplicativos de publicação e assinatura e alguns controlam como o tópico é usado em todo o cluster.

Uma definição de objeto de tópico em cluster deve ser configurada de maneira que todos os gerenciadores de filas no cluster possam usá-la corretamente.

Por exemplo, se o modelo a ser usado para assinaturas gerenciadas (MDURMDL e MNDURMDL) estiver configurados para um nome não padrão da fila, essa fila modelo nomeada deverá ser definida em todos os gerenciadores de filas nos quais as assinaturas gerenciadas serão criadas.

Da mesma forma, se algum atributo for configurado como ASPARENT, o comportamento do tópico será dependente dos nós superiores na árvore de tópicos (ver [Objetos de tópicos administrativos](#)) em cada gerenciador de filas individual no cluster. Isso pode resultar em um comportamento diferente ao publicar ou assinar de gerenciadores de fila diferentes.

Os principais atributos diretamente relacionados a comportamento de publicação/assinatura em todo o cluster são os seguintes:

CLROUTE

Esse parâmetro controla o roteamento de mensagens entre os gerenciadores de filas nos quais publicadores estão conectados e gerenciadores de filas nos quais existem assinaturas correspondentes.

- Você configura a rota para ser direta entre esses gerenciadores de filas ou através de um gerenciador de filas que hospeda uma definição do tópico em cluster. Veja [Clusters de publicação/assinatura](#) para obter mais detalhes.
- Não é possível mudar o **CLROUTE** enquanto o parâmetro **CLUSTER** está configurado. Para mudar o **CLROUTE**, primeiro configure a propriedade **CLUSTER** para estar em branco. Isso impede que os aplicativos que usam o tópico comportem-se de modo clusterizado. Isso, por sua vez, resulta em uma interrupção em publicações sendo entregues para assinaturas, portanto, também é preciso fazer quiesce de sistema de mensagens de publicação/assinatura enquanto faz a alteração.

PROXYSUB

Esse parâmetro controla quando as assinaturas de proxy são feitas.

- **FIRSTUSE** é o valor padrão e faz com que as assinaturas de proxy sejam enviadas em resposta a assinaturas locais em um gerenciador de filas em uma topologia de publicação/assinatura distribuída e canceladas quando não são mais necessárias. Para obter detalhes sobre porque você pode querer alterar o valor padrão **FIRSTUSE** desse atributo, veja [Encaminhamento de assinatura de proxy individual e publicar em todos os lugares](#).
- Para ativar *publicar em todos os lugares*, configure o parâmetro **PROXYSUB** como **FORCE** para um objeto do tópico de alto nível. Isso resulta em uma assinatura de proxy curinga única que corresponde a todos os tópicos abaixo deste objeto do tópico na árvore de tópicos.

Nota: Configurar o atributo **PROXYSUB (FORCE)** em um cluster de publicação/assinatura grande ou ocupado pode resultar em carga excessiva sobre os recursos do sistema. O atributo **PROXYSUB (FORCE)** é propagado para cada gerenciador de filas, não apenas para o gerenciador de filas em que o tópico foi definido. Isso faz com que cada gerenciador de filas no cluster crie uma assinatura de proxy curinga.

Uma cópia de uma mensagem para esse tópico, publicada em qualquer gerenciador de filas no cluster, é enviada para cada gerenciador de filas no cluster – diretamente ou por meio de um gerenciador de filas do host de tópico, dependendo da configuração de **CLROUTE**.

Quando o tópico é encaminhado diretamente, cada gerenciador de fila cria canais do emissor de cluster para gerenciadores de filas alternados. Quando o tópico é encaminhado para o host de tópico, os canais para cada gerenciador de filas de host de tópico são criados a partir de cada gerenciador de filas no cluster.

Para obter mais informações sobre o parâmetro **PROXYSUB** quando usado em clusters, veja [Desempenho de publicação/assinatura encaminhado diretamente](#).

PUBSCOPE and SUBSCOPE

Esses parâmetros determinam se o gerenciador de filas propaga publicações para gerenciadores de filas na topologia (cluster ou hierarquia de publicação/assinatura) ou restringe o escopo para apenas seu gerenciador de filas locais. É possível fazer o trabalho equivalente de maneira programática usando **MQPMO_SCOPE_QMGR** e **MQSO_SCOPE_QMGR**.

PUBSCOPE

Se um objeto do tópico de cluster estiver definido com **PUBSCOPE (QMGR)**, a definição será compartilhada com o cluster, mas o escopo de publicações que são baseadas nesse tópico é apenas local e elas não são enviadas para outros gerenciadores de filas no cluster.

SUBSCOPE

Se um objeto do tópico de cluster estiver definido com **SUBSCOPE (QMGR)**, a definição será compartilhada com o cluster, mas o escopo de assinaturas baseadas nesse tópico é apenas local, portanto, nenhuma das assinaturas de proxy será enviada para outros gerenciadores de filas no cluster.

Esses dois atributos normalmente são usados em conjunto para isolar um gerenciador de filas da interação com outros membros do cluster em tópicos específicos. O gerenciador de filas não publica

nem recebe publicações sobre esses tópicos para e de outros membros do cluster. Essa situação não impede a publicação ou a assinatura se os objetos de tópico estiverem definidos nos subtópicos.

A configuração de **SUBSCOPE** para QMGR em uma definição local de um tópico não impede que outros gerenciadores de filas no cluster propaguem suas assinaturas de proxy para o gerenciador de filas se eles estiverem usando uma versão em cluster do tópico com **SUBSCOPE(ALL)**. Porém, se as definições locais também definirem **PUBSCOPE** para QMGR, essas assinaturas de proxy não receberão publicações desse gerenciador de filas.

Conceitos relacionados

[Escopo da publicação](#)

[Escopo da assinatura](#)



Diversas definições de tópico de cluster com o mesmo nome

É possível definir o mesmo objeto de tópico de cluster nomeado em mais de um gerenciador de filas no cluster, e, em determinados cenários, isso permite um comportamento específico. Quando há várias definições de tópico de cluster para o mesmo nome, a maior parte das propriedades deve corresponder. Se não corresponderem, erros ou avisos serão relatados dependendo da importância da incompatibilidade.

Em geral, se houver uma incompatibilidade nas propriedades de várias definições de tópico de cluster, serão emitidos avisos e uma das definições de objeto de tópico será usada por cada gerenciador de filas no cluster. Qual definição é usada por cada gerenciador de filas não é determinista ou consistente entre os gerenciadores de filas no cluster. Essas incompatibilidades devem ser resolvidas o mais rapidamente possível.

Durante a configuração ou manutenção do cluster, às vezes é necessário criar várias definições de tópico de cluster que não sejam idênticas. No entanto, isso só é útil como uma medida temporária e, portanto, é tratado como uma condição de erro em potencial.



Quando forem detectadas incompatibilidades, as seguintes mensagens de aviso serão gravadas no log de erro de cada gerenciador de filas:

-  No [Multiplataformas](#), [AMQ9465](#) e [AMQ9466](#).
-  No z/OS, [CSQX465I](#) e [CSQX466I](#).

As propriedades escolhidas para qualquer sequência de tópicos em cada gerenciador de filas podem ser determinadas visualizando o status do tópico em vez de as definições de objeto do tópico, por exemplo, usando **DISPLAY TPSTATUS**

Em algumas situações, um conflito nas propriedades de configuração é grave o suficiente para interromper o objeto do tópico que está sendo criado ou causar a incompatibilidade de objetos a serem marcados como inválidos e não propagados em todo o cluster (veja **CLSTATE** em [DISPLAY TOPIC](#)). Essas situações ocorrem quando há um conflito no cluster de roteamento da propriedade (**CLROUTE**) das definições de tópico. Além disso, devido à importância de consistência entre definições roteadas por host de tópico, inconsistências adicionais serão rejeitadas, conforme detalhado nas seções subsequentes deste artigo.

Se o conflito for detectado no momento da definição do objeto, a mudança da configuração será rejeitada. Se detectada posteriormente pelo gerenciadores de filas de repositório completo, as seguintes mensagens de aviso são gravadas nos logs de erro dos gerenciadores de filas:

-  No [Multiplataformas](#): [AMQ9879](#)
-  No z/OS: [CSQX879E](#).

Quando diversas definições do mesmo objeto do tópico forem especificadas no cluster, uma definição especificada localmente terá precedência sobre qualquer uma especificada remotamente. Portanto, se existir alguma diferença nas definições, os gerenciadores de filas que hospedam várias definições se comportarão de maneira diferente um do outro.

O efeito de definir um tópico sem cluster com o mesmo nome que um tópico de cluster de outro gerenciador de filas

É possível definir um objeto do tópico administrado que não esteja em cluster em um gerenciador de filas que esteja em um cluster e, ao mesmo tempo, definir o mesmo objeto do tópico nomeado que uma definição de tópico em cluster em um gerenciador de filas diferente. Nesse caso, o objeto do tópico definido localmente tem precedência sobre todas as definições remotas com o mesmo nome.

Isso tem o efeito de evitar o comportamento de armazenamento em cluster do tópico quando usado a partir deste gerenciador de filas. Ou seja, as assinaturas podem não receber publicações de publicadores remotos, e mensagens de publicadores podem não ser propagadas para assinaturas remotas no cluster.

Consideração cuidadosa deve ser feita antes de configurar um sistema assim, pois isso pode levar a um comportamento confuso.

Nota: Se um gerenciador de filas individual precisar impedir que publicações e assinaturas propagem pelo cluster, mesmo quando o tópico tiver sido agrupado em outro lugar, uma abordagem alternativa é configurar os escopos de publicação e assinatura para apenas o gerenciador de filas local. Consulte o [“Atributos do tópico de cluster”](#) na página 99.

Definições de tópico em vários clusters em um cluster roteado diretamente

Para roteamento direto, você não costuma definir o mesmo tópico de cluster em mais de um gerenciador de filas de cluster. Isso ocorre porque o roteamento direto torna o tópico disponível em todos os gerenciadores de filas no cluster, não importa qual gerenciador de filas foi definido. Além disso, incluir várias definições de tópico de cluster aumenta significativamente a atividade do sistema e a complexidade administrativa, e mais complexidade acarreta uma chance maior de erro humano:

- Cada definição resulta em no envio por push de um objeto de tópico de cluster adicional para os outros gerenciadores de fila no cluster, incluindo os outros gerenciadores de fila de host do tópico de cluster.
- Todas as definições para um tópico específico em um cluster devem ser idênticas, caso contrário, será difícil determinar qual definição de tópico está sendo usada por um gerenciador de filas.

Também não é essencial que o gerenciador de filas do host único esteja continuamente disponível para o tópico funcionar corretamente no cluster, pois a definição de tópico de cluster é armazenada em cache pelos gerenciadores de filas de repositório completo e todos os outros gerenciadores de filas em seus repositórios de cluster parciais. Para obter mais informações, consulte [Disponibilidade de gerenciadores de filas do host de tópico que usam o roteamento direto](#).

Para uma situação em que você possa precisar definir temporariamente um tópico de cluster em um segundo gerenciador de filas, por exemplo, quando o host existente do tópico precisar ser removido do cluster, consulte [Movendo uma definição de tópico de cluster para um outro gerenciador de filas](#).

Se você precisar alterar uma definição de tópico de cluster, tenha cuidado para modificá-lo no mesmo gerenciador de filas que foi definido. Tentar modificá-la de outro gerenciador de filas pode acidentalmente criar uma segunda definição do tópico com atributos de tópico conflitantes.

Definições de tópico em vários clusters em um cluster roteado para host de tópico

Quando um tópico de cluster for definido com uma rota de cluster de *host de tópico*, o tópico será propagado para todos os gerenciadores de filas no cluster da mesma forma que para tópicos roteados *diretos*. Além disso, todas as mensagens de publicar/assinar para esse tópico serão roteadas por meio dos gerenciadores de filas no qual esse tópico está definido. Portanto, o local e o número de definições do tópico no cluster se tornam importantes (consulte [“Roteamento de host do tópico em clusters de publicação/assinatura”](#) na página 84).

Para assegurar a disponibilidade e escalabilidade adequadas, é útil, se possível, ter várias definições de tópico. Veja [Disponibilidade de gerenciadores de filas do host de tópico que usam roteamento de host de tópico](#).

Ao incluir ou remover definições adicionais de um tópico roteados do *host de tópico* em um cluster, é necessário considerar o fluxo de mensagens no momento da mudança na configuração. Se as

mensagens estiverem sendo publicadas no cluster para o tópico no momento da mudança, um processo do intermediário será necessário para incluir ou remover uma definição de tópico. Veja [Movendo uma definição de tópico de cluster para um outro gerenciador de filas](#) e [Incluindo hosts de tópico adicionais para um cluster roteado por host de tópico](#).

Como já explicado, as propriedades das várias definições devem corresponder, com a possível exceção do parâmetro **PUB**, como descrito na próxima seção. Quando publicações são roteadas por meio de gerenciadores de filas de host de tópico, é ainda mais importante que as várias definições sejam consistentes. Portanto, uma inconsistência detectada na sequência de tópicos ou no nome do cluster é rejeitada se uma ou mais das definições de tópico tiver sido configurada para roteamento de cluster do host de tópico.

Nota: Definições do tópico do cluster também são rejeitadas se for feita uma tentativa de configurá-las acima ou abaixo de outro tópico na árvore de tópicos, em que a definição de tópico em cluster existente está configurada para roteamento de host de tópico. Isso impede a ambiguidade no roteamento de publicações com relação a assinaturas curingas.

Manipulação especial para o parâmetro PUB

O parâmetro **PUB** é usado para controlar quando os aplicativos podem publicar em um tópico. No caso de roteamento de host de tópico em um cluster, ele também pode controlar quais gerenciadores de filas de host de tópico são usados para rotear as publicações. Por esse motivo, é permitido ter diversas definições do mesmo objeto do tópico no cluster com configurações diferentes para o parâmetro PUB.

Se várias definições de cluster remoto de um tópico tiverem configurações diferentes para esse parâmetro, o tópico permitirá que publicações sejam enviadas e entregues a assinaturas se as seguintes condições forem atendidas:

- Não há um objeto do tópico correspondente definido no gerenciador de filas ao qual o publicador esteja conectado que esteja configurado como PUB (DISABLED).
- Uma ou mais das várias definições de tópico no cluster é definida para PUB (ENABLED) ou uma ou mais das várias definições de tópico é definida para PUB (ASPARENT) e os gerenciadores de fila locais em que o publicador está conectado e a assinatura definida estão definidos para PUB (ENABLED) em um ponto mais alto na árvore de tópicos.

Para roteamento de host de tópico, quando as mensagens são publicadas por aplicativos conectados a gerenciadores de filas que não são hosts de tópico, as mensagens são roteadas apenas para o tópico que hospeda gerenciadores de filas em que o parâmetro **PUB** não foi explicitamente configurado como DISABLED. Portanto, é possível usar a configuração PUB (DISABLED) para executar o quiesce do tráfego de mensagem por meio de determinados hosts de tópico. Isso pode ser feito para preparar a manutenção ou a remoção de um gerenciador de filas ou pelas razões descritas em [Incluindo hosts de tópico adicionais a um cluster roteado para host de tópico](#).

Disponibilidade de gerenciadores de filas do host de tópico de cluster

Projete seu cluster de publicação/assinatura para minimizar o risco de, no caso de um gerenciador de filas de host de tópico ficar indisponível, o cluster não poder mais processar tráfego para o tópico. O efeito de um gerenciador de filas do host do tópico ficar indisponível depende de se o cluster está usando o roteamento do host do tópico ou roteamento direto.

Disponibilidade de gerenciadores de filas do host de tópico que usam roteamento direto

Para roteamento direto, você não costuma definir o mesmo tópico de cluster em mais de um gerenciador de filas de cluster. Isso ocorre porque o roteamento direto torna o tópico disponível em todos os gerenciadores de filas no cluster, não importa qual gerenciador de filas foi definido. Veja [Diversas definições de tópico de cluster em um cluster encaminhado diretamente](#).

Em um cluster, sempre que o host de um objeto em cluster (por exemplo, uma fila em cluster ou um tópico em cluster) fica indisponível por um longo período, os outros membros do cluster acabarão expirando o conhecimento desses objetos. No caso de um tópico em cluster, se o gerenciador de filas

de host de tópico de cluster ficar indisponível, os outros gerenciadores continuam a processar pedidos de publicação/assinatura para o tópico em um modo de cluster direto (ou seja, enviando publicações para assinaturas em gerenciadores de filas remotas) durante pelo menos 60 dias a partir de quando o tópico que hospeda o gerenciador de filas comunicou-se pela última vez com o gerenciadores de filas de repositório completo. Se o gerenciador de filas no qual você definiu o objeto do tópico de cluster nunca for disponibilizado novamente, os objetos do tópico em cache nos outros gerenciadores de filas acabarão sendo excluídos e o tópico será revertido para um tópico local, e, nesse caso, as assinaturas deixarão de receber publicações de aplicativos conectados aos gerenciadores de filas remotas.

Com o período de 60 dias para recuperar o gerenciador de filas no qual você define um objeto do tópico de cluster, há pouca necessidade de tomar medidas especiais para garantir que um host de tópico de cluster permaneça disponível (entretanto, observe que nenhuma assinatura definida no host do tópico de cluster indisponível permanecerá disponível). O período de 60 dias é suficiente para resolver problemas técnicos, e é provável que seja ultrapassado apenas devido a erros administrativos. Para diminuir essa possibilidade, se o host de tópico de cluster estiver indisponível, todos os membros do cluster gravarão mensagens de log de erro por hora, informando que o cache de tópico de cluster objeto não foi atualizado. Responda a essas mensagens garantindo que o gerenciador de fila em que o objeto do tópico do cluster está definido esteja em execução. Se não for possível tornar o gerenciador de filas de host de tópico de cluster disponível novamente, determine a mesma definição de tópico em cluster, com exatamente os mesmos atributos, em outro gerenciador de filas no cluster.

Disponibilidade de gerenciadores de filas do host de tópico que usam o roteamento de host de tópico

Para roteamento de host de tópico, todas as mensagens de publicação/assinatura para um tópico são roteadas por meio dos gerenciadores de filas no qual o tópico está definido. Por esse motivo, é muito importante considerar a disponibilidade contínua desses gerenciadores de filas no cluster. Se um host de tópico ficar indisponível e não existir nenhum outro host para o tópico, o tráfego dos publicadores para os assinantes em diferentes gerenciadores de fila no cluster imediatamente para o tópico. Se hosts de tópico adicionais estiverem disponíveis, os gerenciadores de filas do cluster encaminham novos tráfegos de publicação através desses hosts de tópico, fornecendo disponibilidade contínua de rotas de mensagens.

Como para tópicos direto, após 60 dias, se o primeiro host de tópico ainda estiver indisponível, o conhecimento do tópico desse host de tópico será removido do cluster. Se essa for a última definição para este tópico restante no cluster, todos os outros gerenciadores de filas param de encaminhar publicações para qualquer host de tópico para roteamento.

Para assegurar disponibilidade e escalabilidade adequadas é, portanto, útil, se possível, definir cada tópico em pelo menos dois gerenciadores de filas do cluster. Isso fornece proteção contra qualquer gerenciador de filas do host do tópico especificado ficar indisponível. Veja também [Várias definições de tópico de cluster em um cluster encaminhado de host de tópico](#).

Se não for possível configurar vários hosts de tópico (por exemplo, porque é preciso preservar a ordem das mensagens) nem configurar apenas um host de tópico (porque a disponibilidade de um gerenciador de filas único não deve afetar o fluxo de publicações para assinaturas entre todos os gerenciadores de filas no cluster), considere configurar o tópico como um tópico encaminhado diretamente. Isso evita depender de um único gerenciador de filas para o cluster inteiro, mas ainda requer que cada gerenciador de filas individual esteja disponível para processar assinaturas e publicadores hospedados localmente.

Inibindo publicação/assinatura em cluster

Introduzir o primeiro tópico em cluster encaminhado diretamente em um cluster força todos os gerenciadores de filas no cluster a tomarem conhecimento de todos os outros gerenciadores de filas, e potencialmente, faz com que eles criem canais uns para os outros. Se isso não for o desejado, configure publicação/assinatura encaminhada por host de tópico. Se a existência de um tópico em cluster encaminhado diretamente puder prejudicar a estabilidade do cluster, devido a preocupações de escala de cada gerenciador de filas, é possível desativar completamente a funcionalidade do cluster de publicação/assinatura configurando **PSCLUS** para **DISABLED** em cada gerenciador de filas no cluster.

Conforme descrito em [“Roteamento direto em clusters de publicar/assinar”](#) na página 79, ao introduzir um tópico em cluster encaminhado diretamente em um cluster, todos os repositórios parciais são

notificados automaticamente sobre todos os outros membros do cluster. O tópico em cluster também pode criar assinaturas em todos os outros nós (por exemplo, onde **PROXYSUB (FORCE)** está especificado) e causar grandes números de canais a serem iniciados a partir de um gerenciador de filas, mesmo quando não há assinaturas locais. Isto coloca uma carga adicional imediata em cada gerenciador de filas no cluster. Para um cluster que contém vários gerenciadores de filas, isso pode causar uma redução significativa no desempenho. Portanto, a introdução de publicação/assinatura encaminhado diretamente para um cluster deve ser cuidadosamente planejada.

Quando você souber que um cluster não pode acomodar as sobrecargas de publicação/assinatura encaminhada diretamente, é possível usar publicação/assinatura encaminhada por host de tópico. Para obter uma visão geral das diferenças, veja [“Projetando clusters publicar/assinar” na página 77](#).

Se preferir desativar completamente a funcionalidade de publicação/assinatura para o cluster, configure o atributo do gerenciador de filas **PSCLUS** para DISABLED em cada gerenciador de filas no cluster. Essa configuração desativa publicação/assinatura encaminhada diretamente e encaminhada por host de tópico no cluster, modificando três aspectos de funcionalidade do gerenciador de filas:

- Um administrador deste gerenciador de filas não consegue mais definir um objeto Topic como em cluster.
- As definições de tópico ou assinaturas de proxy que chegam de outros gerenciadores de filas são rejeitadas, e uma mensagem de aviso é registrada para informar o administrador sobre a configuração incorreta.
- Repositórios completos não compartilham automaticamente informações sobre cada gerenciador de filas com todos os outros repositórios parciais ao receberem uma definição de tópico.

Embora **PSCLUS** seja um parâmetro de cada gerenciador de filas individual em um cluster, ele não se destina a desativar seletivamente a publicação/assinatura em um subconjunto de gerenciadores de filas no cluster. Se você desativar seletivamente dessa maneira, verá mensagens de erro frequentes. Isso ocorre porque assinaturas de proxy e as definições de tópico são constantemente vistas e rejeitadas se um tópico for armazenado em cluster em um gerenciador de filas no qual **PSCLUS** esteja ativado.

Portanto, é preciso ter como objetivo configurar **PSCLUS** para DISABLED em cada gerenciador de filas no cluster. Contudo, na prática, esse estado pode ser difícil de atingir e manter, por exemplo, gerenciadores de filas podem se entrar e sair do cluster a qualquer momento. Como um mínimo, é preciso assegurar que **PSCLUS** seja configurado como DISABLED em todos os gerenciadores de filas de repositório completo. Se você fizer isso, e um tópico em cluster depois for definido em um gerenciador de filas ENABLED no cluster, isso não fará os repositórios completos informarem cada gerenciador de filas sobre cada outro gerenciador de filas, assim, seu cluster está protegido contra problemas de ajuste de escala em potencial em todos os gerenciadores de filas. Nesse cenário, a origem do tópico em cluster é relatada nos logs de erros dos gerenciadores de filas de repositório completo.

Se um gerenciador de filas participar de um ou mais clusters de publicação/assinatura, e também de um ou mais clusters ponto a ponto, configure **PSCLUS** para ENABLED nesse gerenciador de filas. Por essa razão, ao sobrepor um cluster ponto a ponto a um cluster de publicação/assinatura/, use um conjunto separado de repositórios completos em cada cluster. Essa abordagem permite que as definições de tópico e as informações sobre cada gerenciador de filas fluam somente no cluster de publicar/assinar.

Para evitar configurações inconsistentes ao mudar **PSCLUS** de ENABLED para DISABLED, nenhum objeto de tópico em cluster pode existir em qualquer cluster do qual esse gerenciador de filas seja um membro. Os tópicos, mesmo aqueles definidos remotamente, devem ser excluídos antes de alterar **PSCLUS** para DISABLED.

Para obter mais informações sobre **PSCLUS**, consulte [ALTER QMGR \(PSCLUS\)](#).

Conceitos relacionados

[Desempenho do cluster de publicação/assinatura encaminhado diretamente](#)

Clusters de publicação/assinatura e múltiplos

Um único gerenciador de filas pode ser membro de mais de um cluster. Essa disposição é, às vezes, conhecida como *clusters sobrepostos*. Através dessa sobreposição, filas em cluster podem se tornar acessíveis a partir de vários clusters, e o tráfego de mensagens ponto a ponto pode ser encaminhado de

gerenciadores de filas em um cluster para os gerenciadores de filas em outro cluster. Tópicos em cluster em clusters de publicação/assinatura não fornecem a mesma capacidade. Portanto, seu comportamento deve ser bem compreendido ao usar vários clusters.

Diferente de uma fila, não é possível associar uma definição de tópico a mais de um cluster. O escopo de um tópico em cluster está delimitado aos gerenciadores de filas no mesmo cluster para o qual o tópico está definido. Isso permite que as publicações sejam propagadas para assinaturas apenas nos gerenciadores de filas no mesmo cluster.

Uma árvore de tópicos do gerenciador de filas

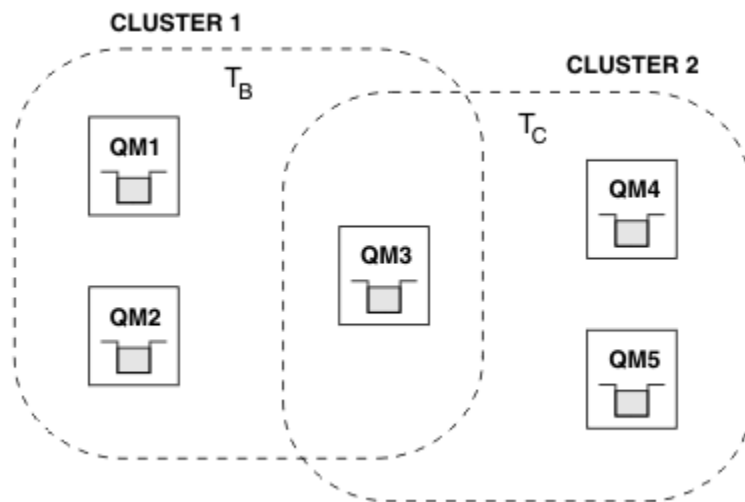


Figura 28. Clusters sobrepostos: dois clusters cada assinando tópicos diferentes

Quando um gerenciador de filas é membro de vários clusters, ele é informado de todos os tópicos em cluster definidos em cada um desses clusters. Por exemplo, na figura anterior, QM3 está ciente tanto dos objetos de tópicos armazenados em cluster administrados por T_B quanto por T_C , enquanto que QM1 só tem conhecimento de T_B . QM3 aplica ambas as definições de tópico ao seu tópico local e, portanto, tem um comportamento diferente para QM1 para determinados tópicos. Por esse motivo, é importante que os tópicos em cluster de clusters diferentes não interfiram uns nos outros. Interferências podem ocorrer quando um tópico em cluster é definido acima ou abaixo de outro tópico em cluster em um cluster diferente (por exemplo, possuem cadeias de tópico /Sport e /Sport/Football) ou inclusive para a mesma sequência de tópicos em ambos. Outra forma de interferência é quando os objetos administrados do tópico em cluster são definidos com o nome de objeto igual em clusters diferentes, mas para sequências de tópicos diferentes.

Se essa configuração for feita, a entrega de publicações para assinaturas correspondentes se tornará muito dependente dos locais relativos dos publicadores e assinantes com relação ao cluster. Por esse motivo, não se pode confiar em tal configuração, e é preciso mudá-la para remover tópicos que causam interferência.

Ao planejar uma topologia de cluster sobreposto com o sistema de mensagens de publicação/assinatura, é possível evitar qualquer interferência tratando a árvore de tópicos e os nomes de objeto do tópico em cluster como se eles abrangessem todos os clusters sobrepostos na topologia.

Integração de diversos clusters de publicação/assinatura

Se houver necessidade de mensagens de publicação/assinatura abrangerem gerenciadores de filas em diferentes clusters, há duas opções disponíveis:

- Conectar os clusters entre si usando uma configuração de hierarquia de publicação/assinatura. Consulte [Combinando os espaços de tópico de diversos clusters](#).

- Criar um cluster adicional que sobreponha os clusters existentes e inclua todos os gerenciadores de filas que precisam publicar ou assinar um tópico específico.

Com a última opção, é preciso considerar cuidadosamente o tamanho do cluster e o mecanismo de roteamento de cluster mais eficiente. Consulte o [“Projetando clusters publicar/assinar”](#) na página 77.

Considerações de design para publicações retidas em clusters de publicação/assinatura

Há poucas restrições a serem consideradas ao projetar um cluster de publicação/assinatura para funcionar com as publicações retidas.

Considerações

Consideração 1: os seguintes gerenciadores de filas do cluster sempre armazenam a versão mais recente de uma publicação retida:

- O gerenciador de filas do publicador
- Em um cluster roteado de host de tópico, o host de tópico (desde que haja apenas um host de tópico para o tópico, conforme explicado na seção seguinte deste artigo)
- Todos os gerenciadores de filas com assinaturas correspondentes à sequência de tópicos da publicação retida

Consideração 2: os gerenciadores de filas não recebem publicações retidas enquanto eles não tiverem assinaturas atualizadas. Portanto, qualquer publicação retida armazenada em um gerenciador de filas que não assina mais o tópico será considerada antiga.

Consideração 3: ao criar qualquer assinatura, se houver uma cópia local de uma publicação retida para a sequência de tópicos, a cópia local será entregues à assinatura. Se você for o primeiro assinante para qualquer sequência de tópicos determinada, uma publicação retida correspondente também será entregue a partir de um dos seguintes membros de cluster:

- Em um cluster encaminhado diretamente, o gerenciador de filas do publicador
- Em cluster encaminhado de host de tópico, os hosts de tópico para o tópico determinado

A entrega de uma publicação retida de um host de tópico ou gerenciador de filas de publicação para o gerenciador de filas de assinatura é assíncrona com as chamadas do `MQSUB`. Portanto, se você usar a chamada do `MQSUBRQ`, a publicação retida mais recente poderá ser perdida até que uma chamada subsequente para o `MQSUBRQ`.

Implicações

Para qualquer cluster de publicação/assinatura, quando uma assinatura é feita pela primeira vez, o gerenciador de filas locais pode estar armazenando uma cópia antiga de uma publicação retida, e esta é a cópia que será entregue à nova assinatura. A existência de uma assinatura no gerenciador de filas local significa que isso se resolverá na próxima vez que a publicação retida for atualizada.

Para um cluster de publicação/assinatura roteado do host do tópico, se você configurar mais de um host de tópico para um determinado tópico, novos assinantes poderão receber a publicação retida mais recente de um host de tópico, ou receber uma publicação retida caducada de outro host de tópico (com o mais recente sendo perdido). Para roteamento de host de tópico, é habitual configurar vários hosts de tópico para um determinado tópico. No entanto, se você esperar que os aplicativos usem publicações retidas, configure apenas um host de tópico para cada tópico.

Para qualquer sequência de tópicos específica, use apenas um único publicador e assegure que o publicador sempre use o mesmo gerenciador de filas. Se você não fizer isso, diferentes publicações retidas poderão estar ativas em gerenciadores de filas diferentes para o mesmo tópico, levando a um comportamento inesperado. Como múltiplas assinaturas de proxy são distribuídas, várias publicações retidas podem ser recebidas.

Se você ainda estiver preocupado com os assinantes usarem publicações caducadas, considere configurar uma expiração de mensagem ao criar cada publicação retida.

É possível usar o comando **CLEAR TOPICSTR** para remover uma publicação retida de um cluster de publicação/assinatura. Em determinadas circunstâncias, pode ser necessário emitir o comando em vários membros do cluster de publicação/assinatura, conforme descrito em [CLEAR TOPICSTR](#).

Assinaturas curinga e publicações retidas

Se você estiver usando assinaturas curinga, as assinaturas de proxy correspondentes entregues para outros membros do cluster de publicação/assinatura serão usadas como curinga para o separador tópico imediatamente antes do primeiro caractere curinga. Veja [Caracteres curinga e tópicos de cluster](#).

Portanto, o caractere curinga usado pode corresponder a mais sequência de tópicos, e a mais publicações retidas, do que corresponderão ao aplicativo de assinatura.

Isso aumenta a quantidade de armazenamento necessário para as publicações retidas e, portanto, é preciso garantir que os gerenciadores de filas de hosting tenham capacidade de armazenamento suficiente.

Conceitos relacionados

[Publicações Retidas](#)

[Encaminhamento de assinatura de proxy individual e publicar em todos os lugares](#)

Contraprestações de REFRESH CLUSTER para clusters de publicação/assinatura

A emissão do comando **REFRESH CLUSTER** resulta em o gerenciador de filas temporariamente descartar informações mantidas localmente sobre um cluster, incluindo quaisquer tópicos de cluster e suas assinaturas de proxy associadas.

O tempo levado da emissão do comando **REFRESH CLUSTER** para o ponto em que o gerenciador de filas recupera um conhecimento total das informações necessárias para o cluster de publicação/assinatura depende do tamanho do cluster, da disponibilidade e da responsividade dos gerenciadores de filas de repositório completo.

Durante o processamento da atualização, há interrupção para tráfego de publicação/assinatura em um cluster de publicação/assinatura. Para clusters grandes, o uso do comando **REFRESH CLUSTER** pode interromper o cluster enquanto ele estiver em andamento e novamente em intervalos de 27 dias depois disso, quando os objetos do cluster enviarem automaticamente atualizações de status para todos os gerenciadores de filas interessados. Consulte [Atualizando em um grande cluster pode afetar o desempenho e disponibilidade do cluster](#). Por esses motivos, o comando **REFRESH CLUSTER** deve ser usado em um cluster de publicação/assinatura apenas sob a orientação do centro de suporte do IBM.

A interrupção para o cluster pode aparecer externamente com os seguintes sintomas:

- Assinaturas para tópicos de cluster nesse gerenciador de filas não estão recebendo publicações de publicadores que estão conectados a outros gerenciadores de filas no cluster.
- As mensagens que são publicadas para tópicos de cluster nesse gerenciador de filas não estão sendo propagadas para assinaturas em outros gerenciadores de filas.
- Assinaturas para tópicos de cluster nesse gerenciador de filas criado durante este período não estão consistentemente enviando assinaturas de proxy para outros membros do cluster.
- As assinaturas para tópicos de cluster nesse gerenciador de filas excluídas durante este período não estão consistentemente removendo as assinaturas de proxy de outros membros do cluster.
- Pausa de 10 segundos ou mais na entrega das mensagens.
- **MQPUT** falha, por exemplo, **MQRC_PUBLICATION_FAILURE**.
- Publicações colocadas na fila de devoluções com motivo de **MQRC_UNKNOWN_REMOTE_Q_MGR**

Por esses motivos, aplicativos de publicação/assinatura precisam ser colocados no modo quiesce antes da emissão do comando **REFRESH CLUSTER**.

Depois de um comando **REFRESH CLUSTER** ser emitido em um gerenciador de filas em um cluster de publicação/assinatura, aguarde até que todos os gerenciadores de filas do cluster e tópicos de cluster sejam atualizados com êxito, então ressincronizar as assinaturas de proxy conforme descrito

em Ressincronização de assinaturas de proxy. Quando todas as assinaturas de proxy tiverem sido corretamente ressincronizadas, reinicie os aplicativos de publicação/assinatura.

Se um comando **REFRESH CLUSTER** estiver levando muito tempo para ser concluído, monitore-o consultando CURDEPTH de SYSTEM . CLUSTER . COMMAND . QUEUE.

Conceitos relacionados

“Armazenamento em Cluster: Usando Melhores Práticas de REFRESH CLUSTER” na página 70

Use o comando **REFRESH CLUSTER** para descartar todas as informações sobre um cluster mantidas localmente e reconstruir essas informações do repositórios completos no cluster. Não deve ser necessário usar esse comando, exceto em circunstâncias excepcionais. Se você precisar usá-lo, existem considerações especiais sobre como fazer isso. Essas informações são um guia baseado em teste e feedback de clientes.

Referências relacionadas

[Problemas do aplicativo vistos ao executar REFRESH CLUSTER](#)

[Referência de Comandos MQSC: REFRESH CLUSTER](#)

Roteamento em hierarquias de publicação/assinatura

Se a topologia do gerenciador de filas distribuída for uma hierarquia de publicação/assinatura, e uma assinatura for feita em um gerenciador de filas, por padrão, uma assinatura de proxy será criada em cada gerenciador de filas na hierarquia. Publicações recebidas em qualquer gerenciador de filas então serão roteadas através da hierarquia para cada gerenciador de filas que hospede uma assinatura correspondente.

Para uma introdução de como as mensagens são encaminhadas entre os gerenciadores de filas em hierarquias de publicação/subscrição e clusters, consulte [Redes de publicação/assinatura distribuídas](#).

Quando uma assinatura para um tópico for feita em um gerenciador de filas em uma hierarquia de publicar/assinar distribuída, o gerenciador de filas gerenciará o processo pelo qual a assinatura é propagada para gerenciadores de filas conectados. *Assinaturas de Proxy* fluem para todos os gerenciadores de filas na rede. Uma assinatura de proxy fornece a um gerenciador de filas as informações de que ele precisa para encaminhar uma publicação para os gerenciadores de filas que hospedam assinaturas para esse tópico. Cada gerenciador de filas em uma hierarquia de publicar/assinar está apenas ciente das suas relações diretas. Publicações colocadas em um gerenciador de filas são enviadas, por meio de suas relações diretas, para os gerenciadores de filas com assinaturas. Isso é ilustrado na figura a seguir, em que o *Assinante 1* registra uma assinatura para um tópico específico no gerenciador de filas *Ásia* (1). As assinaturas de proxy para essa assinatura no gerenciador de filas *Ásia* são encaminhadas para todos os outros gerenciadores de filas na rede (2,3,4).

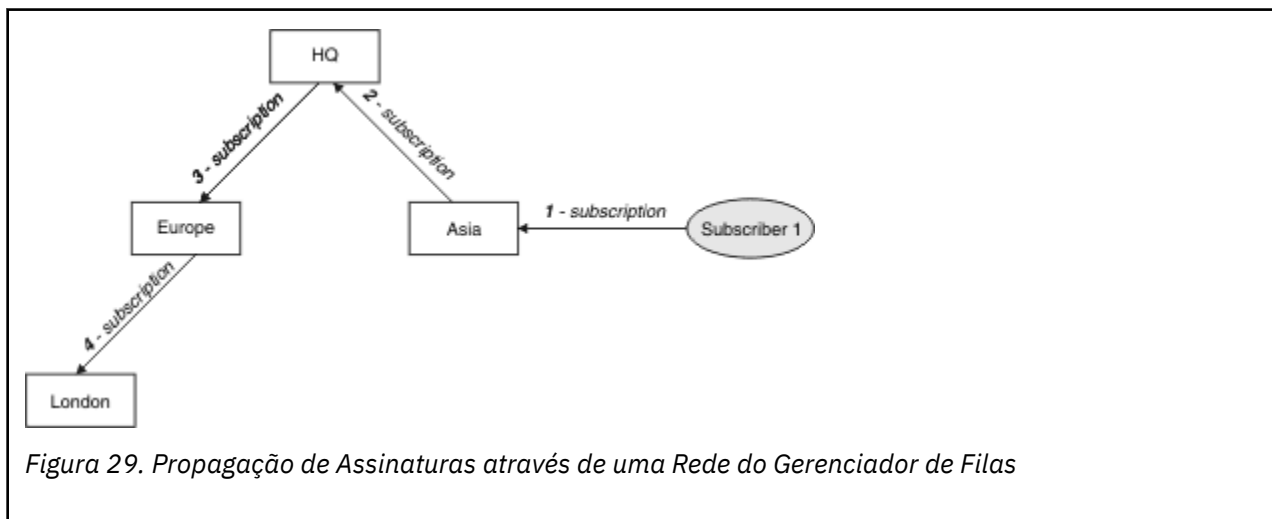
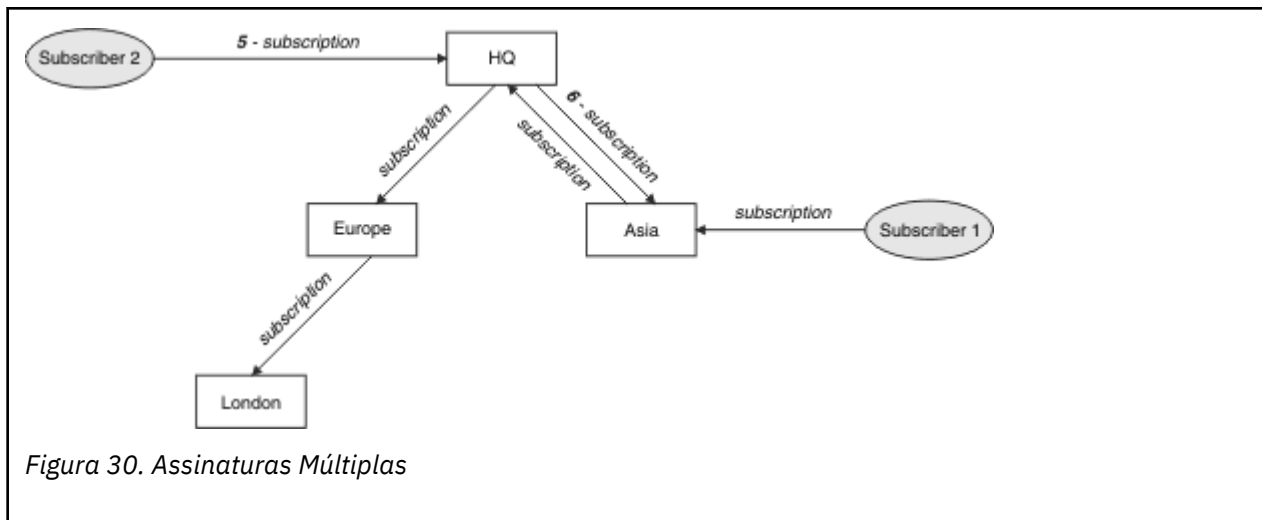


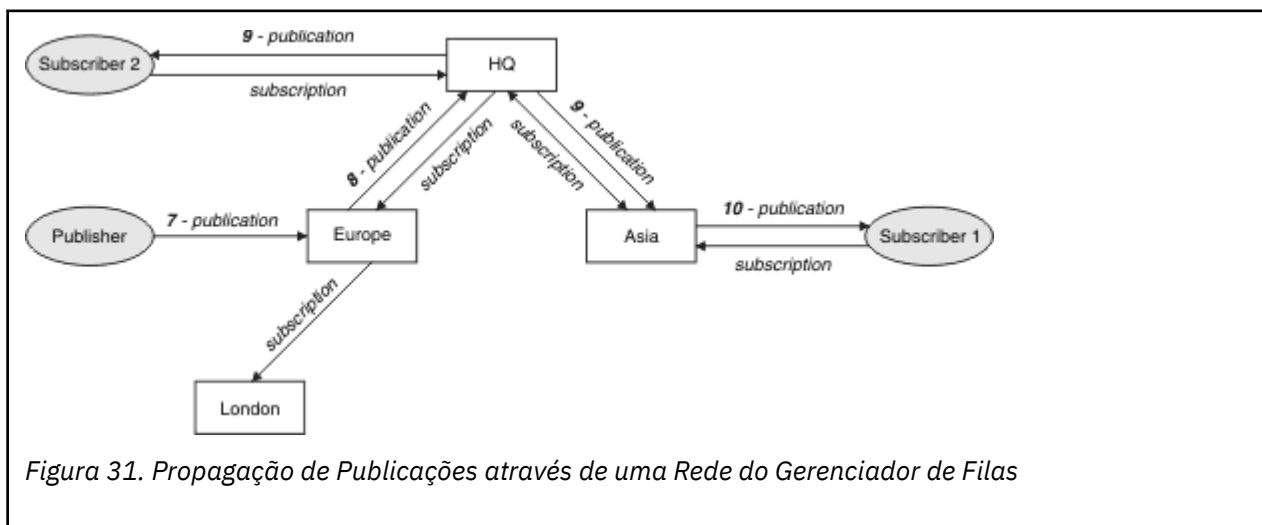
Figura 29. Propagação de Assinaturas através de uma Rede do Gerenciador de Filas

Um gerenciador de filas consolida todas as assinaturas que são criadas nele, se a partir de aplicativos locais ou a partir de gerenciadores de filas remotas. Ele cria assinaturas de proxy para os tópicos das assinaturas com seus vizinhos, a menos que uma assinatura de proxy já exista. Isso é ilustrado na figura

a seguir, em que o *Assinante 2* registra uma assinatura para o mesmo tópico que em [Figura 29](#) na página 109, no gerenciador de filas *HQ* (5). A assinatura para esse tópico é redirecionada para o gerenciador de filas *Ásia*, de modo que ele fica ciente de que as assinaturas existem em qualquer lugar na rede (6). A assinatura não será enviada para o gerenciador de filas *Europe* porque uma assinatura para esse tópico já foi registrada; veja a etapa 3 em [Figura 29](#) na página 109.



Quando um aplicativo publica informações para um tópico, por padrão, o gerenciador de filas receptor as redireciona para todos os gerenciadores de filas que possuem assinaturas válidas para o tópico. Ele pode encaminhá-las através de um ou mais gerenciadores de filas intermediários. Isso é ilustrado na figura a seguir, em que um publicador envia uma publicação, sobre o mesmo tópico que em [Figura 30](#) na página 110, para o gerenciador de filas *Europa* (7). Uma assinatura para esse tópico existe de *HQ* para *Europa*, portanto, a publicação será encaminhada para o gerenciador de filas *HQ* (8). No entanto, não existe nenhuma assinatura de *Londres* para *a* (apenas de *Europa* para *Londres*), portanto, a publicação não será enviada para o gerenciador de filas *Londres*. O gerenciador de filas *HQ* envia a publicação diretamente para o *Assinante 2* e para o gerenciador de filas *Ásia* (9). A publicação será encaminhada para o *Assinante 1* da *Ásia* (10).



Quando um gerenciador de filas envia quaisquer publicações ou assinaturas para outro gerenciador de filas, ele define seu próprio ID do usuário na mensagem. Se você estiver usando uma hierarquia de publicar/assinar e o canal de entrada estiver configurado para colocar mensagens com a autoridade do ID do usuário na mensagem, você deverá autorizar o ID do usuário do gerenciador de filas remetente. Veja [Usando os IDs de usuário padrão com uma hierarquia do gerenciador de filas](#).

Nota: Se, em vez disso, você usar clusters de publicar/assinar, a autorização será manipulada pelo cluster.

Resumo e considerações adicionais

Uma hierarquia publicar/assinar dá controle preciso sobre o relacionamento entre os gerenciadores de filas. Após ter sido criada, sua administração requer pouca intervenção manual. No entanto, isso também impõe certas restrições ao seu sistema:

- Quanto mais altos forem os nós na hierarquia, especialmente o nó raiz deverá ser hospedado em equipamento robusto, altamente disponível e de bom desempenho. Isso é porque espera-se que mais tráfego de publicação flua através desses nós.
- A disponibilidade de cada gerenciador de filas não folha na hierarquia afeta a habilidade da rede de enviar mensagens dos publicadores para os assinantes em outros gerenciador de filas.
- Por padrão, todas as sequências de tópicos assinadas são propagadas através da hierarquia, e as publicações são propagadas apenas para gerenciadores de filas remotas que possuem uma assinatura para o tópico associado. Portanto, mudanças rápidas no conjunto de assinaturas podem se tornar um fator limitante. É possível alterar esse comportamento padrão e, em vez disso, ter todas as publicações propagadas para todos os gerenciadores de filas, o que elimina a necessidade de assinaturas de proxy. Consulte [Desempenho de assinatura em redes publicar/assinar](#).

Nota: Uma restrição similar também se aplica a clusters encaminhados diretos.

- Devido à natureza interconectada dos gerenciadores de filas de publicação/assinatura, leva tempo para que as assinaturas de proxy se propaguem em todos os nós na rede. Publicações remotas não necessariamente começam a ser assinadas imediatamente, assim, publicações iniciais podem não ser enviadas após uma assinatura de uma nova cadeia de tópicos. É possível remover os problemas causados pelo atraso de assinatura propagando todas as publicações para todos os gerenciadores de filas, o que elimina a necessidade de assinaturas de proxy. Consulte [Desempenho de assinatura em redes publicar/assinar](#).

Nota: Essa restrição também se aplica a clusters encaminhados diretos.

- Para uma hierarquia publicar/assinar, incluir ou remover gerenciadores de filas requer configuração manual na hierarquia, com consideração cuidadosa quanto ao local desses gerenciadores de filas e suas dependência de outros gerenciadores de filas. A menos que você esteja incluindo ou removendo gerenciador de filas que estejam na parte inferior da hierarquia, e, portanto, não tenham mais ramificações abaixo deles, também será necessário configurar outros gerenciador de filas na hierarquia.

Antes de usar uma hierarquia publicar/assinar como mecanismo de roteamento, explore as abordagens alternativas detalhadas em [“Roteamento direto em clusters de publicar/assinar”](#) na página 79 e [“Roteamento de host do tópico em clusters de publicação/assinatura”](#) na página 84.

Filas do Sistema de Publicação/Assinatura Distribuído

Quatro filas do sistema são usadas pelos gerenciadores de filas para mensagens de publicação/assinatura. Você precisa estar ciente de sua existência apenas para fins de determinação de problemas e planejamento de capacidade.

Veja [Balanceando produtores e consumidores em redes de publicar/assinar](#) para obter orientação sobre como monitorar essas filas.

Fila do sistema	Propósito
SYSTEM.INTER.QMGR.CONTROL	Fila de controle de publicar/assinar distribuída do IBM MQ
SYSTEM.INTER.QMGR.FANREQ	Fila de entrada de processo de fan-out de assinatura de proxy interna de assinar/publicar distribuída do IBM MQ
SYSTEM.INTER.QMGR.PUBS	Publicações de publicar/assinar distribuídas do IBM MQ
SYSTEM.HIERARCHY.STATE	Estado de relacionamento de hierarquia de publicar/assinar distribuída do IBM MQ

z/OS No z/OS, você configura os objetos do sistema necessários ao criar o gerenciador de filas incluindo as amostras CSQ4INSX, CSQ4INSR e CSQ4INSG no conjunto de dados de entrada de inicialização CSQINP2. Para obter mais informações, consulte [Tarefa 13: customizar os conjuntos de dados de entrada de inicialização](#).

Os atributos das filas do sistema de publicar/assinar são mostrados em [Tabela 7](#) na página 112.

<i>Tabela 7. Os atributos filas do sistema de publicação/assinatura</i>	
Atributo	Valor padrão
DEFPSIST	Sim
DEFSOPT	SHARED
MAXMSGL	<p>Multi Em Multiplataformas: o valor do parâmetro MAXMSGL do comando ALTER QMGR</p> <p>z/OS Em z/OS: 4194304 (ou seja, 4 MB)</p>
MAXDEPTH	999999999
SHARE	N/D
<p>z/OS</p> <p>z/OS</p> STGCLASS	Esse atributo é usado apenas em plataformas z/OS

Nota: A única fila que contém mensagens inseridas por aplicativos é SYSTEM . INTER . QMGR . PUBS. O **MAXDEPTH** é configurado o seu valor máximo para esta fila para permitir a criação temporária de mensagens publicadas durante as períodos de indisponibilidade ou de carga excessiva. Se o gerenciador de filas estiver em execução em um sistema em que a profundidade da fila não possa ser contida, esse deverá ser ajustado.

Tarefas relacionadas

[Resolução de problemas de publicação / assinatura distribuída](#)

Erros de fila do sistema de publicação/assinatura distribuído

Os erros podem ocorrer quando filas do gerenciador de filas de publicação/assinatura distribuídas estiverem indisponíveis. Isso afeta a propagação de conhecimento de assinatura em toda a rede de publicação/assinatura e publicação para assinaturas em gerenciadores de filas remotas.

Se a fila de solicitação de fan out do SYSTEM . INTER . QMGR . FANREQ estiver indisponível, a criação de uma assinatura pode gerar um erro e mensagens de erro serão gravadas no log de erros do gerenciador de filas quando as assinaturas de proxy precisarem ser entregues para gerenciadores de filas diretamente conectados.

Se a hierarquia de relacionamento de estado da fila SYSTEM . HIERARCHY . STATE não estiver disponível, uma mensagem de erro será gravada no log de erros do gerenciador de filas e o mecanismo de publicação/assinatura será colocado no modo COMPAT. Para visualizar o modo de publicação/assinatura, use o comando DISPLAY QMGR PSMODE.

Se qualquer outra das filas SYSTEM . INTER . QMGR estiver indisponível, uma mensagem de erro será gravada no log de erros do gerenciador de filas e, embora a função não seja desativada, é provável que as mensagens de publicação/assinatura constem em filas neste ou nos gerenciadores de filas remotas.

Se a fila do sistema de publicação/assinatura ou a fila de transmissão necessária para um gerenciador de filas de cluster pai, filho ou de publicação/assinatura estiver indisponível, ocorrem os seguintes resultados:

- As publicações não são entregues e um aplicativo de publicação pode receber um erro. Para obter detalhes sobre quando o aplicativo de publicação recebe um erro, consulte os seguintes parâmetros do comando **DEFINE TOPIC : PMSGDLV , NPMSGDLV e USEDLO**.
- Publicações entre gerenciadores de filas recebidas são restauradas para a fila de entrada e depois é feita uma nova tentativa. Se o limite de restauração for atingido, as publicações não entregues serão colocadas na fila de mensagens não entregues. O log de erros do gerenciador de filas conterá detalhes do problema.
- Uma assinatura de proxy não entregue é restaurada para a fila de solicitação fan-out e depois é feita uma nova tentativa. Se o limite de recuperação for alcançado, a assinatura de proxy não entregue não será entregue a nenhum gerenciador de filas conectado e será colocada na fila de mensagens não entregues. O log de erros do gerenciador de filas conterá detalhes do problema, incluindo detalhes de qualquer ação administrativa corretiva necessária.
- As mensagens de protocolo de relacionamento de hierarquia falharam e o status de conexão é sinalizado como ERROR. Para visualizar o status da conexão, use o comando **DISPLAY PUBSUB**

Tarefas relacionadas

[Resolução de problemas de publicação / assinatura distribuída](#)





Multi **Planejando seus requisitos de armazenamento e desempenho em multiplataformas**

Deve-se configurar o armazenamento realista e realizável, e objetivos de desempenho para seu sistema IBM MQ. Use os links para descobrir os fatores que afetam o armazenamento e o desempenho na sua plataforma.






Os requisitos variam conforme os sistemas em que você está usando o IBM MQ e quais componentes você deseja usar.

Para obter as informações mais recentes sobre ambientes de hardware e software suportados, veja [Requisitos do sistema para IBM MQ](#).

O IBM MQ armazena dados do gerenciador de filas no sistema de arquivos. Use os seguintes links para descobrir como planejar e configurar estruturas de diretórios para uso com o IBM MQ:

- [“Planejando o suporte ao sistema de arquivos em multiplataformas” na página 116](#)
- [“Requisitos para sistemas de arquivos compartilhados em multiplataformas” na página 117](#)
- [“Compartilhando arquivos do IBM MQ em multiplataformas” na página 127](#)
-   [“Estrutura de Diretório em Sistemas AIX and Linux” na página 129](#)
-  [“Estrutura de Diretório em Sistemas Windows” na página 138](#)
-  [“Estrutura de diretório no IBM i” na página 141](#)

Use os seguintes links para obter informações sobre os recursos do sistema, memória compartilhada e prioridade de processo no AIX and Linux:

-   [“Recursos IPC do IBM MQ e UNIX System V” na página 146](#)
-  [“memória compartilhada no AIX” na página 145](#)
-   [“Prioridade de processo do IBM MQ e do UNIX” na página 146](#)

Use os links a seguir para obter informações sobre arquivos de log:

- [“Escolhendo a criação de log circular ou linear em multiplataformas” na página 145](#)
- [Calculando o tamanho do log](#)

Conceitos relacionados

[“Planning your IBM MQ environment on z/OS” na página 146](#)

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.

Tarefas relacionadas

“Planejando uma arquitetura do IBM MQ” na página 5

Ao planejar seu ambiente do IBM MQ, considere o suporte que o IBM MQ fornece para arquiteturas de gerenciador de filas único e múltiplo e para estilos de sistema de mensagem ponto a ponto e publicar/ assinar. Além disso, planeje os requisitos de recursos e sua utilização de recursos de criação de log e backup.

Referências relacionadas

[Requisitos de Hardware e Software no AIX and Linux](#)

[Requisitos de Hardware e Software no Windows](#)

Multi

Requisitos de espaço em disco em multiplataformas

Os requisitos de armazenamento para o IBM MQ dependem de quais componentes você instala e de quanto espaço de trabalho você precisa.

O armazenamento em disco é requerido para os componentes opcionais escolhidos para instalação, incluindo os componentes obrigatórios requeridos. O requisito de armazenamento total também depende do número de filas usadas, do número e tamanho das mensagens nas filas e se as mensagens são persistentes. Também é necessário ter a capacidade de arquivamento em disco, fita ou outra mídia e também espaço para seus próprios programas de aplicativo.

As tabelas a seguir mostram o espaço em disco aproximado necessário quando você instala várias combinações do produto em diferentes plataformas. (Os valores são arredondados para perto de 5 MB, onde um MB é 1.048.576 bytes.)

- ▶ **LTS** [“Requisitos de Espaço em Disco para Long Term Support” na página 114](#)
- ▶ **CD** [“Requisitos de Espaço em Disco para Continuous Delivery” na página 115](#)


Requisitos de Espaço em Disco para Long Term Support

LTS ▶ V 9.4.0


Tabela 8. Requisitos de espaço em disco do IBM MQ para Multiplataformas para Long Term Support

Plataforma	Instalação do cliente “1” na página 115	Instalação de servidor “2” na página 115	Instalação completa “3” na página 115
▶ AIX AIX	335 MB	375 MB	1810 MB
▶ IBM i IBM i (consulte Notas adicionais para o 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

Tabela 8. Requisitos de espaço em disco do IBM MQ para Multiplataformas para Long Term Support (continuação)

Plataforma	Instalação do cliente "1" na página 115	Instalação de servidor "2" na página 115	Instalação completa "3" na página 115
 Windows (instalação de 64 bits) "4" na página 115	295 MB	425 MB	2310 MB

Notas:

- Uma instalação de cliente inclui os seguintes componentes:
 - Tempo de execução
 - Client
- Uma instalação do servidor inclui os seguintes componentes:
 - Tempo de execução
 - Servidor
- Uma instalação completa inclui todos os componentes disponíveis.
-  Nem todos os componentes listados aqui são recursos instaláveis nos sistemas Windows; sua funcionalidade às vezes é incluída em outros recursos. Consulte [Recursos do IBM MQ para sistemas do Windows](#).

Notas Adicionais para o IBM i: 

- No IBM i, não é possível separar o cliente nativo do servidor. A figura do servidor na tabela é para 5724H72*BASE sem Java, juntamente com o Carregamento do idioma inglês (2924). Há 22 carregamentos de idioma exclusivos possíveis.
- A figura na tabela é para o cliente nativo 5725A49 *BASE sem Java.
- As classes Java e JMS podem ser incluídas em ambas as ligações do servidor e do cliente. Se desejar incluir esses recursos, inclua 110 MB.
- A inclusão de origem de amostras para o cliente ou servidor inclui um extra de 10 MB.
- A inclusão de amostras para as classes Java e JMS inclui um extra de 5 MB.

Requisitos de Espaço em Disco para Continuous Delivery

Tabela 9. Requisitos de espaço em disco do IBM MQ para Multiplataformas para Continuous Delivery












Liberação de plataforma/CD	Instalação do cliente "1" na página 116	Instalação de servidor "2" na página 116	Instalação completa "3" na página 116
 AIX			
 IBM MQ 9.4.0	355 MB	390 MB	1440 MB
 Linux for x86-64 (64 bits)			
 IBM MQ 9.4.0	280 MB	295 MB	1.195 MB
 Linux on Power Systems - Little Endian			

Tabela 9. Requisitos de espaço em disco do IBM MQ para Multiplataformas para Continuous Delivery (continuação)

Liberação de plataforma/CD	Instalação do cliente ^{"1"} na página 116	Instalação de servidor ^{"2"} na página 116	Instalação completa ^{"3"} na página 116
 IBM MQ 9.4.0	170 MB	195 MB	1075 MB
 Linux for IBM Z			
 IBM MQ 9.4.0	260 MB	290 MB	1.160 MB
 Windows (instalação de 64 bits) ^{"4"} na página 116			
 IBM MQ 9.4.0	300 MB	425 MB	1785 MB

Notas:

- Uma instalação de cliente inclui os seguintes componentes:
 - Tempo de execução
 - Client
- Uma instalação do servidor inclui os seguintes componentes:
 - Tempo de execução
 - Servidor
- Uma instalação completa inclui todos os componentes disponíveis.
-  Nem todos os componentes listados aqui são recursos instaláveis nos sistemas Windows; sua funcionalidade às vezes é incluída em outros recursos. Consulte [Recursos do IBM MQ para sistemas do Windows](#).

Conceitos relacionados

[Componentes e recursos do IBM MQ](#)

Planejando o suporte ao sistema de arquivos em multiplataformas

Os dados do gerenciador de filas são armazenados no sistema de arquivos. Um gerenciador de filas faz uso do bloqueio do sistema de arquivos para evitar que diversas instâncias de um gerenciador de filas de várias instâncias fique ativo ao mesmo tempo.

Sistemas de arquivos compartilhados

Os sistemas de arquivo compartilhado permitem que diversos sistemas acessem o mesmo dispositivo de armazenamento físico simultaneamente. Ocorreriam danos se diversos sistemas acessarem o mesmo dispositivo de armazenamento físico diretamente sem algum meio de forçar o controle de simultaneidade e bloqueio. Os sistemas operacionais fornecem sistemas de arquivos locais com controle de simultaneidade e bloqueio para processos locais; os sistemas de arquivos de rede fornecem controle de simultaneidade e bloqueio para sistemas distribuídos.

Historicamente, os sistemas de arquivo em rede não foram executados rápido o suficiente ou forneceram bloqueio suficiente e controle de simultaneidade, para atender os requisitos para as mensagens de log. Hoje, os sistemas de arquivo em rede podem fornecer bom desempenho e implementações de protocolos confiáveis do sistema de arquivo de rede como *Protocolo RFC 3530*, *Network File System (NFS) versão 4*, para atender os requisitos para as mensagens de log confiavelmente.

Sistemas de arquivo compartilhado e IBM MQ

Os dados do gerenciador de filas para um gerenciador de filas de várias instâncias são armazenados em um Network File System compartilhado. Em sistemas AIX, Linux, and Windows, arquivos de dados e arquivos de log do gerenciador de filas devem ser colocados no sistema de arquivos de rede

compartilhada. **IBM i** No IBM i, os diários são usados em vez dos arquivos de log, e diários não podem ser compartilhados. Gerenciadores de filas de várias instâncias no IBM i usam a replicação de diário ou os diários alternáveis para disponibilizar os diários entre diferentes instâncias do gerenciador de filas.

O IBM MQ usa o bloqueio para evitar que várias instâncias do mesmo gerenciador de filas de várias instâncias estejam ativas ao mesmo tempo O mesmo bloqueio também assegura que dois gerenciadores de filas separados não possam usar inadvertidamente o mesmo conjunto de arquivos de dados do gerenciador de filas. Apenas uma instância por vez de um gerenciador de filas pode ter seu bloqueio. Consequentemente, o IBM MQ suporta dados do gerenciador de filas armazenados no armazenamento de rede acessados como um sistema de arquivo compartilhado.

Como nem todos os protocolos que bloqueio dos sistemas de arquivo de rede são robustos e como um sistema de arquivo pode ser configurado para desempenho em vez da integridade de dados, você deve executar o comando **amqmfscck** para testar se um sistema de arquivo de rede controlará o acesso para os dados do gerenciador de filas e logs corretamente. Este comando se aplica somente ao sistemas UNIX, Linux e IBM i. No Windows, há apenas um sistema de arquivos de rede suportado e o comando **amqmfscck** não é necessário.

Tarefas relacionadas

[“Verificando o comportamento do sistema de arquivo compartilhado em multiplataformas” na página 119](#)
Execute **amqmfscck** para verificar se um sistema de arquivos compartilhado no AIX, Linux ou IBM i atende aos requisitos para armazenar os dados de um gerenciador de filas de várias instâncias. (O único requisito para uma configuração do Windows é que ele use SMB 3 para provisão de armazenamento compartilhado.)

Multi Requisitos para sistemas de arquivos compartilhados em multiplataformas

sistemas de arquivos compartilhados devem fornecer integridade de gravação de dados, garantir acesso exclusivo aos arquivos e liberar bloqueios em falha para funcionarem confiavelmente com o IBM MQ.

Requisitos que um sistema de arquivo compartilhado deve atender

Há três requisitos fundamentais que um sistema de arquivo compartilhado deve atender para trabalhar de forma confiável com o IBM MQ:

1. Integridade de gravação de dados

A integridade de gravação de dados é, às vezes, chamada de *Gravação através de disco em liberação*. O gerenciador de filas deve ser capaz de sincronizar-se com os dados sendo confirmados com êxito no dispositivo físico. Em um sistema transacional, é preciso ter certeza de que algumas gravações foram confirmadas com segurança antes de continuar outro processamento.

Mais especificamente, as plataformas IBM MQ for AIX or Linux usam a opção de abertura *O_SYNC* e a chamada de sistema *fsync()* para forçar explicitamente as gravações em mídia recuperável, além de a operação de gravação ser dependente da operação correta dessas opções.



Atenção: **Linux** Deve-se montar o sistema de arquivo com a opção assíncrono, que ainda suporta a opção de gravações síncronas e fornece melhor desempenho do que a opção sincronização.

Observe, no entanto, que, se o sistema de arquivos tiver sido exportado do Linux, ainda será preciso exportar o sistema de arquivos usando a opção sincronização.

2. Acesso exclusivo garantido a arquivos

Para sincronizar os vários gerenciadores de filas, é necessário haver um mecanismo para um gerenciador de filas obter um bloqueio restrito em um arquivo.

3. Liberar bloqueios na falha

Se um gerenciador de filas falhar, ou se houver uma falha na comunicação com o sistema de arquivos, arquivos travados pelo gerenciador de filas precisam ser desbloqueados e disponibilizados para outros processos sem esperar que o gerenciador de filas seja reconectado ao sistema de arquivos.

Um sistema de arquivo compartilhado deve atender a esses requisitos para que o IBM MQ opere confiavelmente. Caso contrário, os dados do gerenciador de filas e logs serão corrompidos ao usar o sistema de arquivo compartilhado em uma configuração do gerenciador de filas de várias instâncias.

Para gerenciadores de filas de várias instâncias no Microsoft Windows, o armazenamento em rede deve ser acessado pelo protocolo Server Message Block (SMB) usado por redes do Microsoft Windows. O cliente Server Message Block (SMB) não atende aos requisitos do IBM MQ para a semântica de bloqueio em plataformas diferentes do Microsoft Windows, portanto, os gerenciadores de filas de várias instâncias em execução em plataformas diferentes do Microsoft Windows não devem usar o Server Message Block (SMB) como seu sistema de arquivo compartilhado.

Para gerenciadores de filas de várias instâncias em outras plataformas suportadas, o armazenamento deverá ser acessado por um protocolo de sistema de arquivos de rede em conformidade com Posix e com suporte para bloqueio baseado em lease. O Network File System 4 atende a esse requisito. Sistemas de arquivos mais antigos, como Network File System Versão 3, que não têm um mecanismo confiável para liberar bloqueios após uma falha, não devem ser usados com gerenciadores de filas de várias instâncias.

Verifica se o sistema de arquivo compartilhado atende aos requisitos

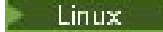


É preciso verificar se o sistema de arquivo compartilhado que você planeja usar atende a esses requisitos. Também se deve verificar se o sistema de arquivos está corretamente configurado para confiabilidade. Sistemas de arquivos compartilhados às vezes fornecem opções de configuração para melhorar o desempenho à custa da confiabilidade.

Para obter informações adicionais, consulte [Instrução de teste para sistemas de arquivos do gerenciador de filas de várias instâncias do IBM MQ](#).

Sob circunstâncias normais, o IBM MQ funciona corretamente com o armazenamento em cache de atributos e não é necessário desativar o armazenamento em cache, por exemplo, configurando NOAC em uma montagem NFS. O cache de atributos pode causar problemas quando vários clientes do sistema de arquivos estiverem competindo por acesso de gravação para o mesmo arquivo no servidor do sistema de arquivos, uma vez que os atributos em cache usados por cada cliente podem não ser iguais a esses atributos no servidor. Um exemplo de arquivos acessados dessa forma são logs de erros do gerenciador de filas para um gerenciador de filas de várias instâncias. Os logs de erros do gerenciador de filas podem ser gravado por uma instância de gerenciador de filas ativa e em espera, e os atributos do arquivo em cache podem fazer os logs de erros ficarem maiores do que o esperado antes de a rolagem dos arquivos ocorrer.

Para ajudar a verificar o sistema de arquivos, execute a tarefa [Verificando o comportamento do sistema de arquivo compartilhado](#). Esta tarefa verifica se o seu sistema de arquivos compartilhado atende aos requisitos [2](#) e [3](#). É necessário verificar o requisito [1](#) na documentação do sistema de arquivo compartilhado ou experimentar criar log de dados no disco.

Falhas de disco podem causar erros ao gravar em disco, que o IBM MQ relata como erros de primeira captura de dados com falha. É possível executar o verificador de sistema de arquivos de seu sistema operacional para verificar o sistema de arquivo compartilhado quanto a quaisquer falhas de disco. Por exemplo:

-   No AIX and Linux, o verificador do sistema de arquivos é chamado de fsck.
-  Em plataformas Windows, o verificador de sistema de arquivos se chama CHKDSK ou SCANDISK.

Segurança do servidor NFS

Notas:

- Não é possível usar as opções **nosuid** ou **noexec** para um ponto de montagem usado para manter o diretório de instalação IBM MQ . Isso ocorre porque o IBM MQ inclui programas executáveis `setuid / setgid` e eles não devem ser impedidos de executar corretamente.
- Ao colocar dados do gerenciador de filas somente em um servidor Network File System (NFS), é possível usar as três opções a seguir com o comando `mount` para tornar o sistema seguro, sem impacto prejudicial para a execução do gerenciador de filas:

noexec

Usando essa opção, você interrompe a execução de arquivos binários no NFS, o que evita que um usuário remoto execute um código indesejado no sistema.

nosuid

Usando essa opção, você evita o uso dos bits `set-user-identifier` e `set-group-identifier`, o que evita que um usuário remoto obtenha privilégios mais altos.

nodev

Usando essa opção, você para o uso ou a definição de dispositivos especiais de caractere e de bloco, o que evita que um usuário remoto saia de um `chroot jail`.

Verificando o comportamento do sistema de arquivo compartilhado em multiplataformas

Execute **amqmfscck** para verificar se um sistema de arquivos compartilhado no AIX, Linux ou IBM i atende aos requisitos para armazenar os dados de um gerenciador de filas de várias instâncias. (O único requisito para uma configuração do Windows é que ele use SMB 3 para provisão de armazenamento compartilhado.)

Antes de começar

É necessário ter um servidor com armazenamento em rede e dois outros servidores conectados a ele que possuam IBM MQ instalado. Você deve ter autoridade de administrador (raiz) para configurar o sistema de arquivos e ser um Administrador IBM MQ para executar **amqmfscck**.

Sobre esta tarefa

O “Requisitos para sistemas de arquivos compartilhados em multiplataformas” na página 117 descreve os requisitos do sistema de arquivo para usar um sistema de arquivo compartilhado com gerenciadores de filas de várias instâncias. A nota técnica do IBM MQ [Instrução de teste para sistemas de arquivos do gerenciador de filas de várias instâncias do IBM MQ](#) lista os sistemas de arquivos compartilhados com os quais o IBM já testou. O procedimento nesta tarefa descreve como testar um sistema de arquivos para ajudá-lo a avaliar se um sistema de arquivos mantém a integridade dos dados.

Failover de um gerenciador de filas de várias instâncias pode ser acionado por falhas de hardware ou software, incluindo problemas de rede que impeçam o gerenciador de filas de gravar em seus dados ou arquivos de log. Principalmente, você está interessado em causar falhas no servidor de arquivos. Mas também se deve fazer os servidores IBM MQ falharem para testar se quaisquer bloqueios são liberados com êxito. Para estar confiante em um sistema de arquivo compartilhado, teste todas as seguintes falhas e quaisquer outras falhas específicas do seu ambiente:

1. Encerrar o sistema operacional no servidor de arquivos, incluindo a sincronização os discos.
2. Descontinuar o sistema operacional no servidor de arquivos sem sincronizar os discos.
3. Pressionar o botão de reconfiguração em cada um dos servidores.
4. Puxar o cabo de rede para fora de cada um dos servidores.
5. Puxar o cabo de energia para fora de cada um dos servidores.
6. Desativar cada um dos servidores.

Crie o diretório no armazenamento em rede que será usado para compartilhar dados e logs do gerenciador de filas. O proprietário do diretório deve ser um Administrador do IBM MQ, ou em outras palavras, um membro do grupo mqm no AIX and Linux. O usuário que executa os testes deve ter autoridade de administrador do IBM MQ.

Use o exemplo de exportação e montagem de um sistema de arquivos em [Criando um gerenciador de filas de várias instâncias em Linux](#) ou [Criando um gerenciador de filas de várias instâncias usando espelhamento de diário e NetServer em IBM i](#) para ajudá-lo a configurar o sistema de arquivos. Diferentes sistemas de arquivos requerem diferentes etapas de configuração. Leia a documentação do sistema de arquivos.

Nota: Execute o IBM MQ MQI client programa de amostra **amqsfhac** em paralelo com **amqmfscck** para demonstrar que um gerenciador de filas mantém a integridade da mensagem durante uma falha..

Procedimento

Em cada um dos controles, cause todas as falhas na lista anterior durante a execução do verificador do sistema de arquivos. Se você pretender executar **amqsfhac** ao mesmo tempo que o **amqmfscck**, execute a tarefa [“Executando amqsfhac para testar a integridade da mensagem”](#) na página 124 em paralelo com essa tarefa.

1. Monte o diretório exportado nos dois servidores do IBM MQ.

No servidor do sistema de arquivos, crie um diretório compartilhado `share` de um subdiretório para salvar os dados para os gerenciadores de filas de várias instâncias, `qmdata`. Para obter um exemplo de configuração de um diretório compartilhado para gerenciadores de filas de várias instâncias no Linux, consulte [Criando um gerenciador de filas de várias instâncias no Linux](#).

2. Verifique o comportamento do sistema de arquivos básico.

Em um servidor IBM MQ, execute o verificador de sistema de arquivo sem parâmetros.

No servidor IBM MQ 1:

```
amqmfscck /shared/qmdata
```

3. Verifique a gravação simultaneamente para o mesmo diretório de ambos os servidores IBM MQ.

Em ambos os servidores IBM MQ, execute o verificador de sistema de arquivos ao mesmo tempo com a opção `-c`.

No servidor IBM MQ 1:

```
amqmfscck -c /shared/qmdata
```

No servidor IBM MQ 2:

```
amqmfscck -c /shared/qmdata
```

4. Verifique a espera e a liberação de bloqueios em ambos os servidores do IBM MQ.

Em ambos os servidores do IBM MQ, execute o verificador de sistema de arquivos ao mesmo tempo com a opção `-w`.

No servidor IBM MQ 1:

```
amqmfscck -w /shared/qmdata
```

No servidor IBM MQ 2:

```
amqmfscck -w /shared/qmdata
```

5. Verifique a integridade dos dados.

a) Formate o arquivo de teste.

Crie um arquivo grande no diretório que está sendo testado. O arquivo é formatado de modo que as fases subsequentes possam ser concluídas com êxito. O arquivo deve ser grande o suficiente para que haja tempo o bastante para interromper a segunda fase para simular o failover. Tente o valor padrão de 262144 páginas (1 GB). O programa automaticamente reduz esse padrão nos sistemas de arquivos lentos de modo que a formatação seja concluída em cerca de 60 segundos

No servidor IBM MQ 1:

```
amqmfscck -f /shared/qmdata
```

O servidor responde com as seguintes mensagens:

```
Formatting test file for data integrity test.
```

```
Test file formatted with 262144 pages of data.
```

b) Gravar dados no arquivo de teste usando o verificador do sistema de arquivos enquanto se causa uma falha.

Execute o programa de teste em dois servidores ao mesmo tempo. Inicie o programa de teste no servidor que sofrerá a falha, então inicie o programa de teste no servidor que irá sobreviver à falha. Cause a falha que você está investigando.

O primeiro programa de teste para com uma mensagem de erro. O segundo programa de teste obtém o bloqueio no arquivo de teste e grava dados no arquivo de teste começando onde o primeiro programa de teste parou. Deixe a execução do segundo programa de teste ser concluída.

Tabela 10. Executando a verificação de integridade de dados em dois servidores ao mesmo tempo

Servidor IBM MQ 1	Servidor IBM MQ 2
amqmfscck -a /shared/qmdata	
Please start this program on a second machine with the same parameters.	amqmfscck -a /shared/qmdata
File lock acquired.	Waiting for lock...
Start a second copy of this program with the same parameters on another server.	Waiting for lock...
Writing data into test file.	Waiting for lock...
To increase the effectiveness of the test, interrupt the writing by ending the process, temporarily breaking the network connection to the networked storage, rebooting the server or turning off the power.	Waiting for lock...
Turn the power off here.	Waiting for lock...

<i>Tabela 10. Executando a verificação de integridade de dados em dois servidores ao mesmo tempo (continuação)</i>	
Servidor IBM MQ 1	Servidor IBM MQ 2
	<pre>File lock acquired. Reading test file Checking the integrity of the data read. Appending data into the test file after data already found. The test file is full of data. It is ready to be inspected for data integrity.</pre>

A sincronização do teste depende do comportamento do sistema de arquivos. Por exemplo, geralmente demora de 30 a 90 segundos para um sistema de arquivos liberar os bloqueios de arquivos obtidos pelo primeiro programa após uma falta de energia. Se você tiver tempo suficiente para apresentar a falha antes de o primeiro programa de teste preencher o arquivo, use a opção **-x amqmfscck** para excluir o arquivo de teste. Tente o teste do início com um arquivo de teste maior.

c) Verifique a integridade dos dados no arquivo de teste.

No servidor IBM MQ 2:

```
amqmfscck -i /shared/qmdata
```

O servidor responde com as seguintes mensagens:

```
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. Exclua os arquivos de teste.

No servidor IBM MQ 2:

```
amqmfscck -x /shared/qmdata
Test files deleted.
```

O servidor responde com a mensagem:

```
Test files deleted.
```

Resultados

O programa retornará um código de saída zero se os testes forem concluídos com êxito, caso contrário, retornará um valor diferente de zero.

Exemplos

O primeiro conjunto de três exemplos mostra o comando produzindo saída mínima.

Teste bem-sucedido do bloqueio de arquivo básico em um servidor

```
> amqmfscck /shared/qmdata
The tests on the directory completed successfully.
```

Teste malsucedido de bloqueio de arquivo básico em um servidor

```
> amqmfscck /shared/qmdata
AMQ6245: Error Calling 'write()[2]' on file '/shared/qmdata/amqmfscck.lck' error '2'.
```

Teste bem-sucedido de bloqueio de dois servidores

<i>Tabela 11. Bloqueio bem-sucedido em dois servidores</i>	
Servidor IBM MQ 1	Servidor IBM MQ 2
<pre>> amqmfscck -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.</pre>	
	<pre>> amqmfscck -w /shared/qmdata Waiting for lock...</pre>
<pre>[Return pressed] Lock released.</pre>	
	<pre>Lock acquired. The tests on the directory completed successfully</pre>

O segundo conjunto de três exemplos mostra os mesmos comandos usando o modo detalhado.

Teste bem-sucedido do bloqueio de arquivo básico em um servidor

```
> amqmfscck -v /shared/qmdata
System call: stat("/shared/qmdata")
System call: fd = open("/shared/qmdata/amqmfscck.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/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd1 = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd1, F_SETLK, F_RDLCK)
System call: fd2 = open("/shared/qmdata/amqmfscck.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.
```

Teste malsucedido de bloqueio de arquivo básico em um servidor

```
> amqmfscck -v /shared/qmdata
System call: stat("/shared/qmdata")
System call: fd = open("/shared/qmdata/amqmfscck.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/amqmfsc.lck", O_RDWR, 0666)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfsc.lck", O_RDWR, 0666)
System call: fcntl(fd, F_SETLK, F_RDLCK)
System call: fdSameFile = open("/shared/qmdata/amqmfsc.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/amqmfsc.lck', errno 2
(Permission denied).

```

Teste bem-sucedido de bloqueio de dois servidores

Tabela 12. Bloqueio bem-sucedido em dois servidores - modo detalhado	
Servidor IBM MQ 1	Servidor IBM MQ 2
<pre> > amqmfsc -wv /shared/qmdata Calling 'stat("/shared/qmdata")' Calling 'fd = open("/shared/qmdata/ amqmfsc.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. </pre>	
	<pre> > amqmfsc -wv /shared/qmdata Calling 'stat("/shared/qmdata")' Calling 'fd = open("/shared/qmdata/ amqmfsc.lkw", O_EXCL O_CREAT O_RDWR,0666)' Calling 'fd = open("/shared/qmdata/amqmfsc.lkw, O_RDWR, 0666)' Calling 'fcntl(fd, F_SETLK, F_WRLCK)' 'Waiting for lock... </pre>
<pre> [Return pressed] Calling 'close(fd)' Lock released. </pre>	
	<pre> Calling 'fcntl(fd, F_SETLK, F_WRLCK)' Lock acquired. The tests on the directory completed successfully </pre>

Referências relacionadas

[Programas de amostra de alta disponibilidade](#)

Multi Executando amqsfhac para testar a integridade da mensagem

Execute o programa de amostra IBM MQ MQI client **amqsfhac** em paralelo com o **amqmfsc** para demonstrar que um gerenciador de filas mantém a integridade da mensagem durante uma falha.

Antes de começar

Você requer quatro servidores para esse teste. Dois servidores para o gerenciador de filas de várias instâncias, um para o sistema de arquivos e um para executar **amqsfhac** como um aplicativo IBM MQ MQI client.

Siga a etapa “1” na página 120 em “Verificando o comportamento do sistema de arquivo compartilhado em multiplataformas” na página 119 para configurar o sistema de arquivos para um gerenciador de filas de várias instâncias.

Sobre esta tarefa

O programa de amostra do IBM MQ MQI client **amqsfhac** verifica se um gerenciador de filas que usa o armazenamento em rede mantém a integridade de dados após uma falha. Execute **amqsfhac** em paralelo com **amqmfscck** para demonstrar que um gerenciador de filas mantém a integridade da mensagem durante uma falha..

Procedimento

1. Crie um gerenciador de filas de várias instâncias em outro servidor, QM1, usando o sistema de arquivos criado na etapa “1” na página 120 em Procedimento.

Consulte [Criar um gerenciador de filas de várias instâncias](#).

2. Inicie o gerenciador de filas em ambos os servidores tornando-o altamente disponível.

No servidor 1:

```
strmqm -x QM1
```

On server 2:

```
strmqm -x QM1
```

3. Configure a conexão do cliente para executar **amqsfhac**.
 - a) Use o procedimento em *Verificando uma instalação do IBM MQ* para a plataforma, ou plataformas, que sua empresa usa para configurar uma conexão do cliente ou os scripts de exemplo em [Amostras de cliente reconectáveis](#).
 - b) Modifique o canal do cliente para ter dois endereços IP, correspondendo aos dois servidores executando QM1.

No o script de exemplo, modifique:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +  
CONNAME('LOCALHOST(2345)') QMNAME(QM1) REPLACE
```

Para:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +  
CONNAME('server1(2345),server2(2345)') QMNAME(QM1) REPLACE
```

em que `server1` e `server2` são os nomes de host dos dois servidores da web e 2345 é a porta na qual o ouvinte de canal está atendendo. Geralmente, ela é padronizada para 1414. É possível usar 1414 com configuração do listener padrão.

4. Crie duas filas locais no QM1 para o teste.
Execute o script MQSC:

```
DEFINE QLOCAL(TARGETQ) REPLACE  
DEFINE QLOCAL(SIDEQ) REPLACE
```

5. Teste a configuração com **amqsfhac**

```
amqsfhac QM1 TARGETQ SIDEQ 2 2 2
```

6. Teste a integridade da mensagem enquanto estiver testando a integridade do sistema de arquivos.

Execute **amqsfhac** durante a etapa “5” na página 120 de “[Verificando o comportamento do sistema de arquivo compartilhado em multiplataformas](#)” na página 119..

```
amqsfhac QM1 TARGETQ SIDEQ 10 20 0
```

Se você parar a instância ativa do gerenciador de filas, **amqsfhac** reconecta-se à instância do gerenciador de filas depois de ela tornar-se ativa. Reinicie a instância do gerenciador de filas parada novamente para poder reverter a falha no próximo teste. Provavelmente será necessário aumentar o número de iterações com base em experiências com seu ambiente para que o programa de teste seja executado por tempo suficiente para que o failover ocorra.

Resultados

Um exemplo de execução de **amqsfhac** na etapa “6” na página 126 é mostrado no exemplo a seguir. Neste exemplo, o teste é um sucesso.

```
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
```

Se o teste tiver detectado um problema, a saída irá comunicar a falha. Em algumas execuções de teste, MQRC_CALL_INTERRUPTED pode relatar "Resolving to backed out". Não faz diferença para o resultado. O resultado depende de se a gravação em disco foi confirmada pelo armazenamento de arquivo em rede antes ou depois de a falha ocorrer.

Referências relacionadas

amqmfack ([verificação de sistema de arquivos](#))

[Programas de amostra de alta disponibilidade](#)

Alguns arquivos do IBM MQ são acessado exclusivamente por um gerenciador de filas ativo, outros arquivos são compartilhados.

Os arquivos IBM MQ são divididos em arquivos de programas e arquivos de dados. Arquivos de programas normalmente são instalados localmente em cada servidor em execução no IBM MQ. Gerenciadores de fila compartilham acesso a arquivos e diretórios de dados no diretório de dados padrão. Eles requerem acesso exclusivo às próprias árvores de diretórios de gerenciador de filas contidas em cada um dos diretórios `qmgrs` e `log` mostrados em [Figura 32 na página 127](#).

O [Figura 32 na página 127](#) é uma visão de alto nível da estrutura de diretórios do IBM MQ. Ele mostra os diretórios que podem ser compartilhadas entre os gerenciadores de filas e tornados remotos. Os detalhes variam por plataforma. As linhas pontilhadas indicam caminhos configuráveis.

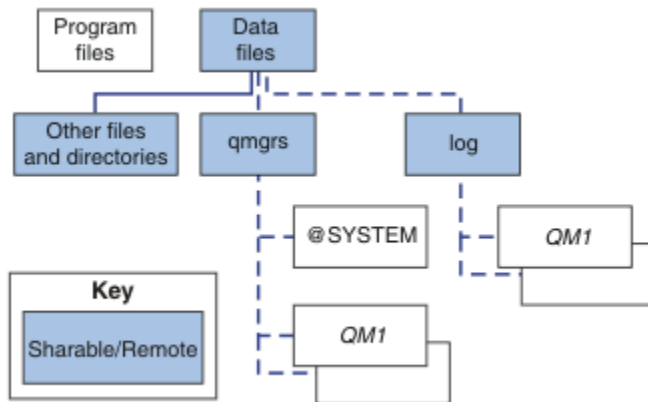


Figura 32. Visão geral da estrutura de diretório do IBM MQ

Arquivos do Programa

O diretório de arquivos do programa geralmente é deixado no local padrão, é local e compartilhado por todos os gerenciadores de filas no servidor.

Arquivos de dados

O diretório de arquivos de dados é tipicamente local no local padrão, `/var/mqm` em sistemas AIX and Linux e configuráveis na instalação no Windows. Ele é compartilhado entre gerenciadores de filas. É possível tornar remoto o local padrão, mas não compartilhá-lo entre diferentes instalações do IBM MQ. O atributo `DefaultPrefix` na configuração do IBM MQ aponta para este caminho.

qmgrs

Há duas maneiras alternativas de especificar o local dos dados do gerenciador de fila.

Usando o atributo Prefix

O atributo **Prefix** especifica o local do diretório `qmgrs`. O IBM MQ constrói o nome do diretório a partir do nome do gerenciador de filas e o cria como um subdiretório do diretório `qmgrs`.

O atributo **Prefix** está localizado na sub-rotina `QueueManager` do arquivo `mqm.ini` e é herdado do valor no atributo **DefaultPrefix** da sub-rotina `All Queue Managers`. Por padrão, para simplicidade administrativa, os gerenciadores de filas geralmente compartilham o mesmo diretório `qmgrs`.

Se você alterar o local do diretório `qmgrs` para qualquer gerenciador de filas, deverá alterar o valor de seu atributo **Prefix**.

O atributo **Prefix** para o diretório `QM1` em [Figura 32 na página 127](#) para uma plataforma AIX and Linux é este:

```
Prefix=/var/mqm
```

Usando o atributo `DataPath`

O atributo `DataPath` especifica o local do diretório de dados do gerenciador de filas.

O atributo `DataPath` especifica o caminho completo, incluindo o nome do diretório de dados do gerenciador de filas. O atributo `DataPath` é diferente do atributo `Prefix`, que especifica um caminho incompleto para o diretório de dados do gerenciador de filas..

O atributo `DataPath`, se for especificado, está localizado na sub-rotina `QueueManager` do arquivo `mqs.ini`. Se tiver sido especificado, ele tem precedência sobre qualquer valor no atributo `Prefix`.

Se você alterar o local do diretório de dados do gerenciador de filas para qualquer gerenciador de filas, é preciso alterar o valor do atributo `DataPath`.

O atributo `DataPath` para o diretório QM1 em [Figura 32 na página 127](#), para uma plataforma Linux ou AIX, é o seguinte:

```
DataPath=/var/mqm/qmgrs/QM1
```

Log

O diretório de log é especificado separadamente para cada gerenciador de filas na [sub-rotina de log](#) na configuração do gerenciador de filas A configuração do gerenciador de filas está em `qm.ini`.

Subdiretórios do `DataPath/QmgrName/@IPCC`

Os subdiretórios do `DataPath/QmgrName/@IPCC` estão no caminho do diretório compartilhado. Eles são usados para construir o caminho de diretório para objetos do sistema de arquivos do IPC. Eles precisam distinguir o namespace de um gerenciador de filas quando um gerenciador de filas é compartilhado entre sistemas.

Os objetos do sistema de arquivos IPC têm de ser distinguidos pelo sistema Um subdiretório, para cada sistema em que o gerenciador de filas é executado, é incluído no caminho do diretório, veja [Figura 33 na página 128](#).

```
DataPath/QmgrName/@IPCC/esem/myHostName/
```

Figura 33. Subdiretório IPC de Exemplo

`myHostName` é até os primeiros 20 caracteres do nome do host retornado pelo sistema operacional. Em alguns sistemas, o nome do host pode ter até 64 caracteres de comprimento antes do truncamento. O valor gerado de `myHostName` pode causar um problema por dois motivos:

1. Os primeiros 20 caracteres não são exclusivos.
2. O nome do host é gerado por um algoritmo DHCP que nem sempre aloca o mesmo nome do host a um sistema.

Nesses casos, configure `myHostName` usando a variável de ambiente `MQS_IPC_HOST`; consulte [Figura 34 na página 128](#).

```
export MQS_IPC_HOST= myHostName
```

Figura 34. Exemplo: configurando `MQS_IPC_HOST`

Outros arquivos e diretórios

Outros arquivos e diretórios, como o diretório que contém arquivos de rastreamento e o log de erro comum, normalmente são compartilhados e mantidos no sistema de arquivos local.

Com suporte de sistemas de arquivos compartilhados, o IBM MQ gerencia acesso exclusivo a esses arquivos usando bloqueios do sistema de arquivos. Um bloqueio de sistema de arquivos permite a apenas uma instância de um gerenciador de filas em particular estar ativa por vez.

Quando você inicia a primeira instância de um gerenciador de filas específico, ele assume a propriedade do diretório do gerenciador de filas. Se você iniciar uma segunda instância, ela poderá assumir propriedade apenas se a primeira instância for interrompida. Se o primeiro gerenciador de fila ainda estiver em execução, a segunda instância falhará em iniciar e relatará que o gerenciador de filas está em execução em outro lugar. Se o primeiro gerenciador de filas tiver parado e o segundo gerenciador de filas assume a propriedade dos arquivos do gerenciador de filas e se torna o gerenciador de filas em execução.

É possível automatizar o procedimento de o segundo gerenciador de filas assumir a propriedade do primeiro. Inicie o primeiro gerenciador de filas com a opção `strmqm -x` que permite que outro gerenciador de filas assuma a sua propriedade. O segundo gerenciador de filas então aguarda até que os arquivos do gerenciador de filas sejam desbloqueado antes de tentar assumir a propriedade dos arquivos do gerenciador de filas, e inicia.

Linux AIX Estrutura de Diretório em Sistemas AIX and Linux

A estrutura de diretório do IBM MQ em sistemas AIX and Linux pode ser mapeada para os sistemas de arquivos diferentes para facilitar o gerenciamento, melhor desempenho e melhor confiabilidade.

Use a estrutura de diretório flexível de IBM MQ para aproveitar os sistemas de arquivos compartilhados para executar gerenciadores de filas de várias instâncias.

Use o Comando `crtmqm QM1` para criar a estrutura de diretório mostrada em [Figura 35 na página 129](#) em que R é a liberação do produto. Ela é uma estrutura de diretório típica para um gerenciador de filas criado em um sistema IBM MQ Alguns diretórios, arquivos e configurações de atributo .ini são omitidas para clareza e outro nome do gerenciador de filas pode ser alterado por desfiguração. Os nomes dos sistemas de arquivos variam em diferentes sistemas.

Em uma instalação típica, todo gerenciador de filas que você cria aponta para diretórios comuns `log` e `qmgrs` no sistema de arquivos local. Em uma configuração de várias instâncias, os diretórios `log` e `qmgrs` estão em um sistema de arquivos de rede compartilhado com outra instalação de IBM MQ.

O [Figura 35 na página 129](#) mostra a configuração padrão para o IBM MQ v7.R no AIX em que R é a liberação do produto. Para obter exemplos de configurações de várias instâncias alternativas, consulte [“Exemplo de configurações do diretório em sistemas AIX and Linux” na página 134](#).

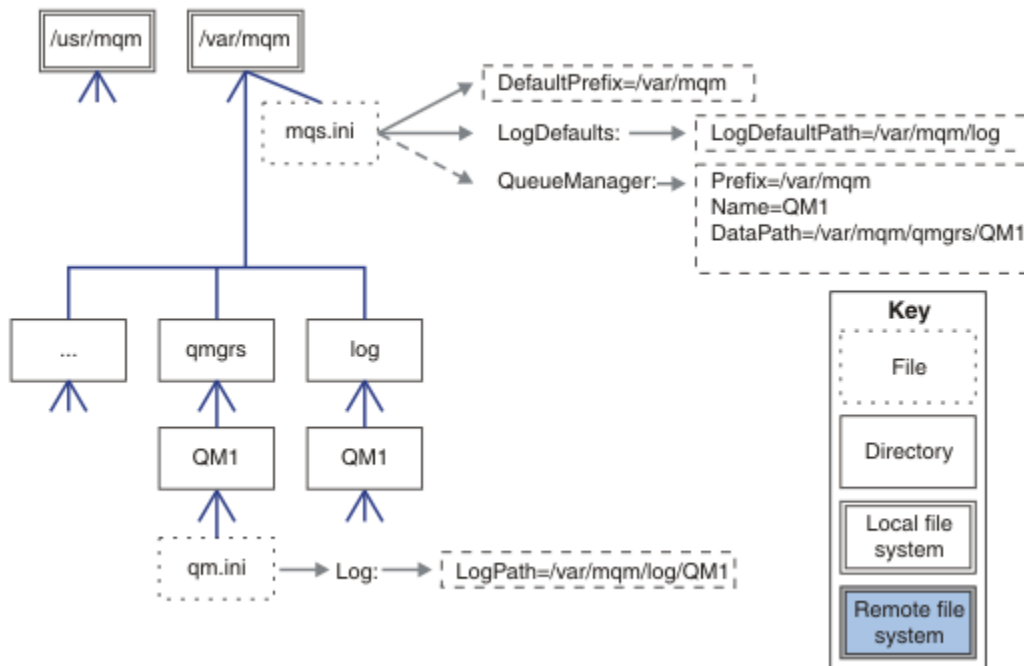


Figura 35. Exemplo de estrutura de diretórios padrão do IBM MQ para os sistemas AIX and Linux

O produto é instalado em `/usr/mqm` em AIX e `/opt/mqm` nos outros sistemas, por padrão. Os diretórios de trabalho são instalados no diretório `/var/mqm`.

Nota: Se você criou o sistema de arquivos `/var/mqm` antes de instalar o IBM MQ, certifique-se de que o usuário `mqm` tenha permissões de diretório completas, por exemplo, modo de arquivo 755.

Nota: O diretório `/var/mqm/errors` deve ser um sistema de arquivos separado para evitar FFDCs produzidos pelo gerenciador de filas a partir do preenchimento do sistema de sistema de arquivos que contém o `/var/mqm`.

Consulte [Criando sistemas de arquivos nos sistemas AIX and Linux](#) para obter mais informações.

Os diretórios `log` e `qmgrs` são mostrados em seus locais padrão conforme definido pelos valores padrão dos atributos `LogDefaultPath` e `DefaultPrefix` no arquivo `mqm.ini`. Quando um gerenciador de filas é criado, por padrão o diretório de dados do gerenciador de filas é criado no `DefaultPrefix/qmgrs` e o diretório de arquivos de log no `LogDefaultPath/log`. `LogDefaultPath` e `DefaultPrefix` afeta apenas onde os gerenciadores de fila e os arquivos de log são criados por padrão. O local real de um diretório gerenciador de filas é salvo no arquivo `mqm.ini` e o local do diretório de arquivos log é salvo no arquivo `qm.ini`.

O diretório de arquivo de log para um gerenciador de filas é definido no arquivo `qm.ini` no atributo `LogPath`. Use a opção `-ld` no comando `crtmqm`, para configurar o atributo `LogPath` para um gerenciador de filas; por exemplo, `crtmqm -ld LogPath QM1`. Se omitir o parâmetro `ld`, o valor de `LogDefaultPath` será usado em substituição.

O diretório de dados do gerenciador de filas é definido no atributo `DataPath` na sub-rotina `QueueManager` no arquivo `mqm.ini`. Use a opção `-md` no comando `crtmqm` para definir o `DataPath` para um gerenciador de filas; por exemplo, `crtmqm -md DataPath QM1`. Se omitir o parâmetro `md`, o valor do atributo `DefaultPrefix` ou `Prefix` será usado em substituição. `Prefix` tem precedência sobre `DefaultPrefix`.

Geralmente, crie `QM1` especificando os diretórios de dados e de log em um único comando.

```
crtmqm  
-md DataPath -ld  
LogPath QM1
```

É possível modificar o local de um log do gerenciador de filas e diretórios de dados de um gerenciador de filas existente, editando os atributos `DataPath` e `LogPath` no arquivo `qm.ini` quando o gerenciador de filas está parado.

O caminho para o diretório `errors`, como os caminhos para todos os outros diretórios no `/var/mqm`, não é modificável. No entanto, os diretórios podem ser montados em diferentes sistemas de arquivos ou vinculados simbolicamente para diferentes diretórios.

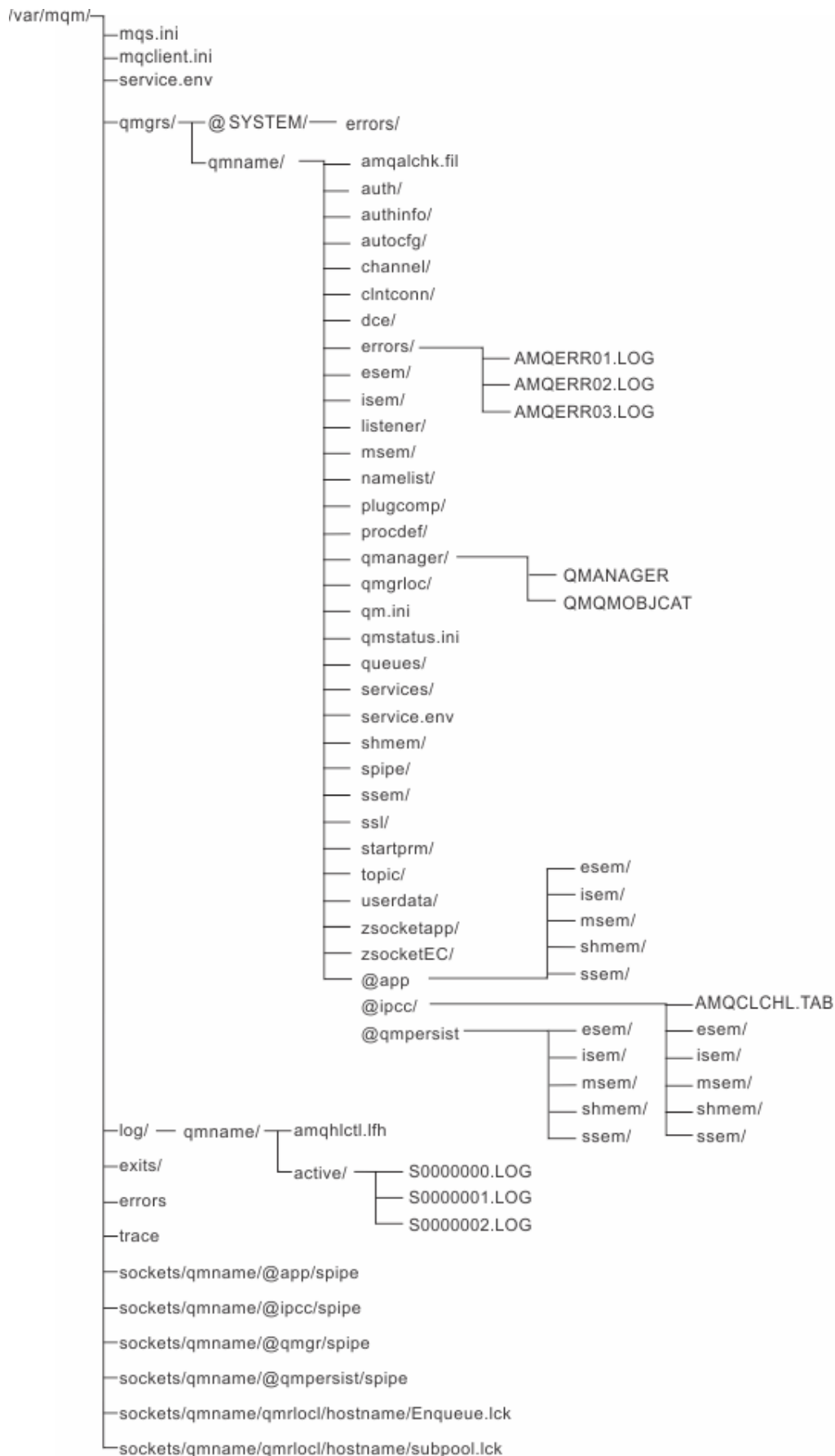
Conteúdo do diretório nos sistemas AIX and Linux

Conteúdo dos diretórios associados a um gerenciador de filas.

Para obter informações sobre o local dos arquivos do produto, consulte [Escolhendo um local de instalação](#)

Para obter informações sobre as configurações alternativas do diretório, consulte [“Planejando o suporte ao sistema de arquivos em multiplataformas”](#) na página 116.

A estrutura de diretórios a seguir é representativa de IBM MQ depois que um gerenciador de filas está em uso por algum tempo. A estrutura real que você tem depende de quais operações ocorreram no gerenciador de filas.



/var/mqm/

O diretório */var/mqm* contém arquivos de configuração e diretórios de saída que se aplicam a uma instalação IBM MQ como um todo, e não a um gerenciador de filas individual.

Tabela 13. Conteúdo documentado do diretório /var/mqm no AIX and Linux

Nome do diretório ou arquivo	Índice
<u>mqs.ini</u>	Arquivo de configuração do IBM MQ de ampla instalação; ler quando um gerenciador de filas é iniciado. Caminho do arquivo modificável usando a variável de ambiente AMQ_MQS_INI_LOCATION . Certifique-se de que isto seja definido e exportado no shell no qual o comando strmqm é executado.
<u>mqclient.ini</u>	Arquivo de configuração do cliente lido pelos programas do IBM MQ MQI client. Caminho do arquivo modificável usando a variável de ambiente MQCLNTCF .
<u>service.env</u>	Contém variáveis de ambiente de escopo de máquina para um processo de serviços. Caminho do arquivo fixo.
<u>errors/</u>	Logs de erro de escopo de máquina e arquivos do FFST. Caminho do diretório fixo. Consulte também FFST: sistemas IBM MQ for UNIX e Linux .
<u>sockets/</u>	Contém informações de cada gerenciador de filas apenas para uso do sistema.
<u>trace/</u>	Arquivos de rastreamento. Caminho do diretório fixo.
<u>web /</u>	Diretório do servidor mqweb.
<u>saídas /</u>	Diretório padrão contendo programas de saída do canal do usuário. Local modificável em sub-rotinas ApiExit no arquivo mqs.ini.
<u>exits64/</u>	

/var/mqm/qmgrs/qmname/

O */var/mqm/qmgrs/qmname/* contém diretórios e arquivos para um gerenciador de filas. O diretório está bloqueado para acesso exclusivo pela instância ativa do gerenciador de filas. O caminho do diretório é diretamente modificável no arquivo *mqs.ini* ou usando a opção **md** do comando **crtmqm**.

Tabela 14. Conteúdo documentado do diretório /var/mqm/qmgrs/qmname no AIX and Linux

Nome do diretório ou arquivo	Índice
<u>qm.ini</u>	Arquivo de configuração do gerenciador de filas; ler quando um gerenciador de filas é iniciado.
<u>errors/</u>	Logs de erro do escopo do gerenciador de filas. <i>qmname</i> = @system contém mensagens relacionadas ao canal para um gerenciador de filas desconhecido ou indisponível.

Tabela 14. Conteúdo documentado do diretório `/var/mqm/qmgrs/qmname` no AIX and Linux (continuação)

Nome do diretório ou arquivo	Índice
<code>@ipcc/AMQCLCHL.TAB</code>	Tabela de controle do canal de cliente padrão, criada pelo servidor IBM MQ e lida pelos programas do IBM MQ MQI client. Caminho do arquivo modificável usando as variáveis de ambiente MQCHLLIB e MQCHLTAB .
<code>qmanager</code>	Arquivo objeto do gerenciador de filas: QMANAGER Catálogo de objetos do gerenciador de filas: QMQMOBJCAT
<code>authinfo/</code>	Cada objeto definido dentro do gerenciador de filas é associado a um arquivo nesses diretórios. O nome do arquivo aproximadamente corresponde ao nome da definição; veja, Entendendo IBM MQ nomes de arquivos .
<code>channel/</code>	
<code>clntconn/</code>	
<code>listener/</code>	
<code>lista de nomes /</code>	
<code>procdef/</code>	
<code>filas /</code>	
<code>services/</code>	
<code>tópicos /</code>	
...	Outros diretórios usados pelo IBM MQ, como <code>@ipcc</code> , para serem modificados apenas pelo IBM MQ.
<code>dados do usuário /</code>	Pode ser usado para armazenar o estado persistente dos aplicativos (pode ser usado pelo RDQM ao mover os gerenciadores de filas para diferentes nós. Consulte Armazenando o status do aplicativo persistente .)
<code>DataPath\autocfg</code>	Usado para configuração automática

`/var/mqm/log/qmname/`

`/var/mqm/log/qmname/` contém os arquivos de log do gerenciador de filas. O diretório está bloqueado para acesso exclusivo pela instância ativa do gerenciador de filas. O caminho do diretório é modificável no arquivo `qm.ini` ou usando a opção **ld** do comando **crtmqm**.

Tabela 15. Conteúdo documentado do diretório `/var/mqm/log/qmname` no AIX and Linux

Nome do diretório ou arquivo	Índice
<code>amqhlctl.lfh</code>	Arquivo de controle de log.
<code>ativo /</code>	Esse diretório contém os arquivos de log numerados S0000000.LOG, S0000001.LOG, S0000002.LOG, e assim por diante.

`/opt/mqm`

`/opt/mqm` é, por padrão, o diretório de instalação na maioria das plataformas. Consulte [“Requisitos de espaço em disco em multiplataformas”](#) na página 114 para obter mais informações sobre a quantidade

de espaço necessária para o diretório de instalação na plataforma, ou plataformas, que a sua empresa usa.

Linux > AIX **Exemplo de configurações do diretório em sistemas AIX and Linux**

Exemplos de configurações do sistema de arquivos alternativo em sistemas AIX and Linux.

É possível customizar a estrutura de diretório do IBM MQ de várias maneiras para atingir vários objetivos diferentes.

- Coloque os diretórios `qmgrs` e `log` em sistemas de arquivos compartilhados remotos para configurar um gerenciador de filas de várias instâncias.
- Use sistemas de arquivos separados para os diretórios de dados e de log e aloque os diretórios para discos diferentes para melhorar o desempenho, reduzindo a contenção de E/S.
- Use dispositivos de armazenamento mais rápidos para os diretórios que têm maior impacto no desempenho. A latência do dispositivo físico é frequentemente o fator mais importante no desempenho de mensagens persistentes do que um dispositivo montado local ou remotamente. A lista a seguir mostra quais diretórios são mais e menos sensíveis ao desempenho.
 1. `log`
 2. `qmgrs`
 3. Outros diretórios, incluindo `/usr/mqm`
- Crie os diretórios `qmgrs` e `log` em sistemas de arquivos que são alocados para armazenamento com boa resiliência, como uma matriz de disco redundante, por exemplo.
- É melhor armazenar os logs de erros comuns em `var/mqm/errors`, localmente, em vez de em um sistema de arquivos de rede, de modo que o erro relacionado ao sistema de arquivos de rede pode ser logado.

Figura 36 na página 135 é um modelo a partir do qual as estruturas de diretório alternativas do IBM MQ são derivadas. No modelo, linhas pontilhadas representam os caminhos que são configuráveis. Nos exemplos, as linhas pontilhadas são substituídas por linhas sólidas que correspondem às informações de configuração armazenadas na variável de ambiente de `AMQ_MQS_INI_LOCATION`, e nos arquivos `mqs.ini` e `qm.ini`.

Nota: As informações do caminho são mostradas como aparece nos arquivos `mqs.ini` ou `qm.ini`. Se você fornecer parâmetros de caminho no comando `crtmqm`, omite o nome do diretório do gerenciador de filas: o nome do gerenciador de filas é adicionado no caminho por IBM MQ.

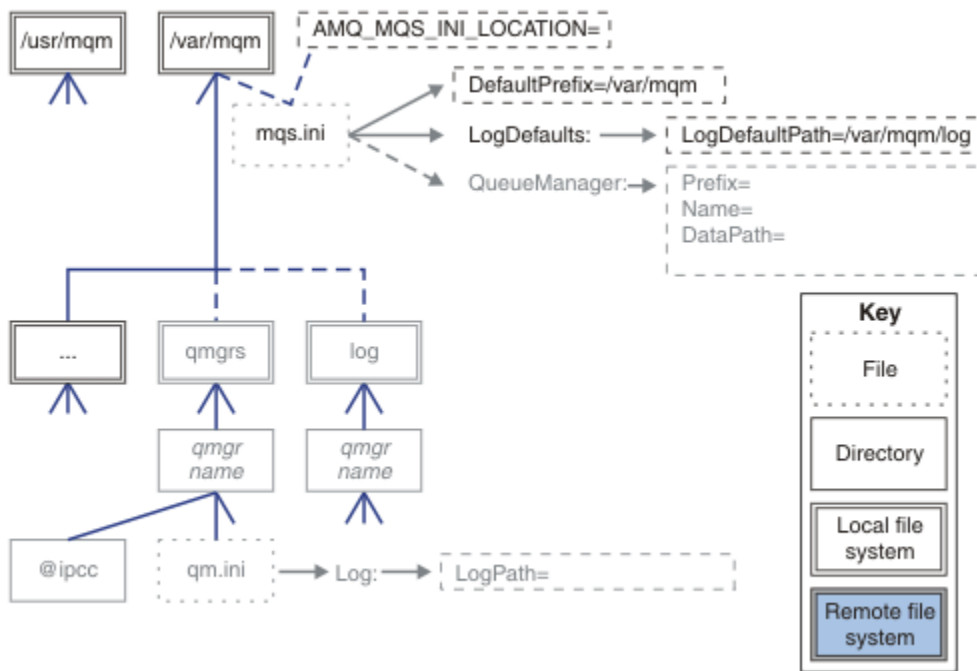


Figura 36. Modelo do padrão da estrutura de diretório

Estrutura de diretório típica para IBM MQ

A Figura 37 na página 135 é a estrutura de diretório padrão criada na IBM MQ, emitindo o comando `crtmqm QM1`.

O arquivo `mqs.ini` tem uma sub-rotina para o gerenciador de filas QM1, criado ao se referir ao valor de `DefaultPrefix`. A sub-rotina `Log` no arquivo `qm.ini` tem um valor para `LogPath`, configurado como referência para `LogDefaultPath` no `mqs.ini`.

Use o parâmetro opcional `crtmqm` para substituir os valores padrão de `DataPath` e `LogPath`.

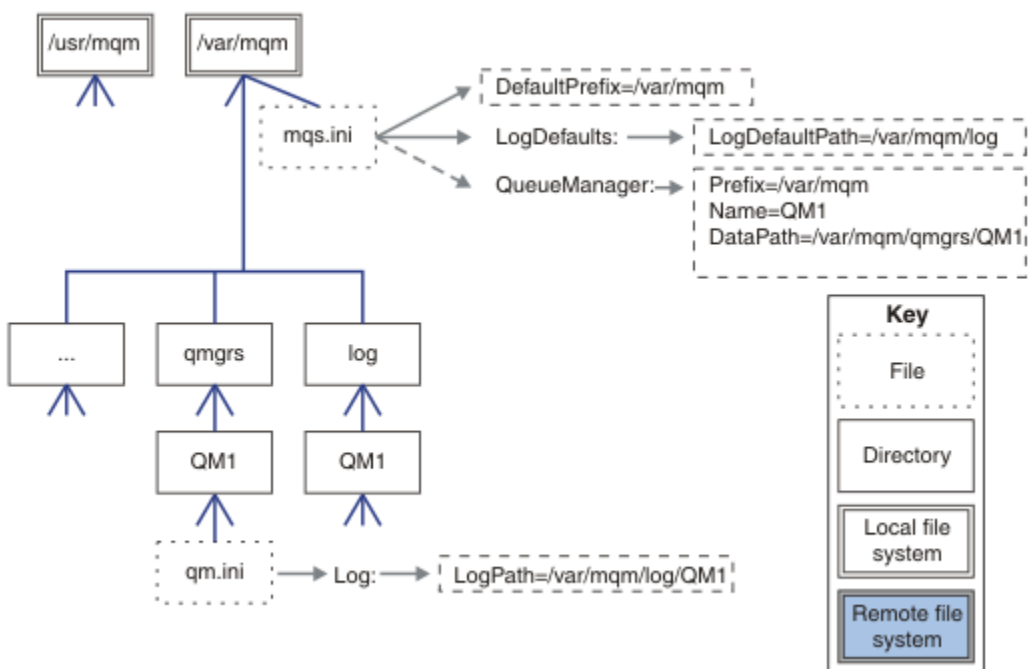


Figura 37. Exemplo de estrutura de diretórios padrão do IBM MQ para os sistemas AIX and Linux

Compartilhar diretórios qmgrs e log padrão

Uma alternativa para “Compartilhar tudo” na página 137 é compartilhar os diretórios qmgrs e log separadamente (Figura 38 na página 136). Nessa configuração, não há necessidade de configurar AMQ_MQS_INI_LOCATION , pois o padrão mqs.ini é armazenado no sistema de arquivos /var/mqm local. Os arquivos e diretórios, como mqclient.ini e mqserver.ini também não são compartilhados.

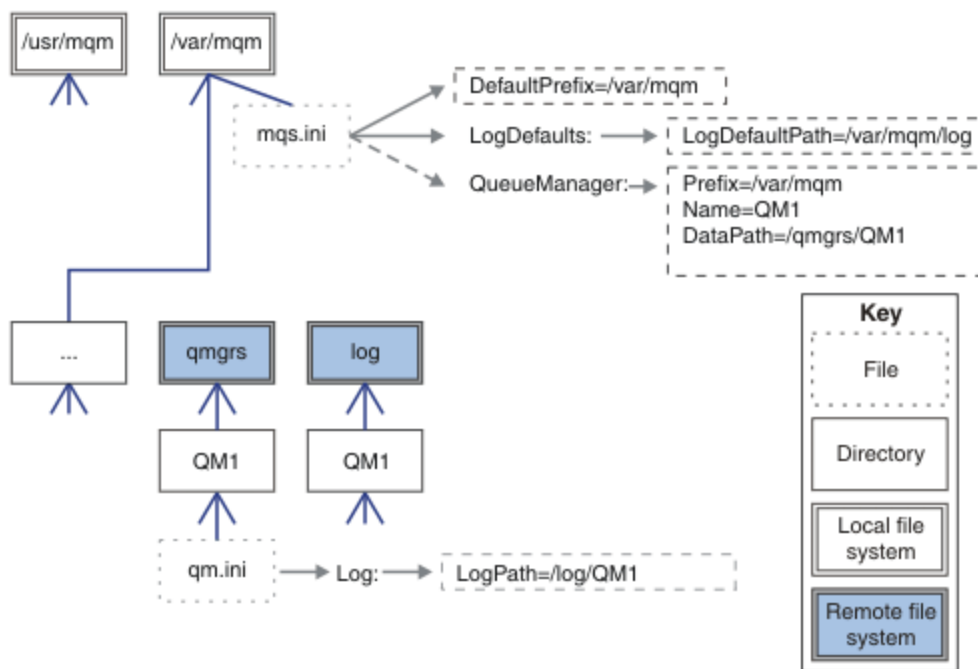


Figura 38. Compartilhar os diretórios qmgrs e log

Compartilhar diretórios denominados qmgrs e log

A configuração no Figura 39 na página 137 coloca log e qmgrs em um sistema remoto de arquivos compartilhados comum chamado /ha. A mesma configuração física pode ser criada de duas formas diferentes.

1. Configure LogDefaultPath=/ha e, em seguida, execute o comando **crtmqm - md /ha/qmgrs QM1**. O resultado é exatamente conforme ilustrado em Figura 39 na página 137.
2. Deixe os caminhos padrão inalterados e, em seguida, execute o comando, **crtmqm - ld /ha/log - md /ha/qmgrs QM1**.

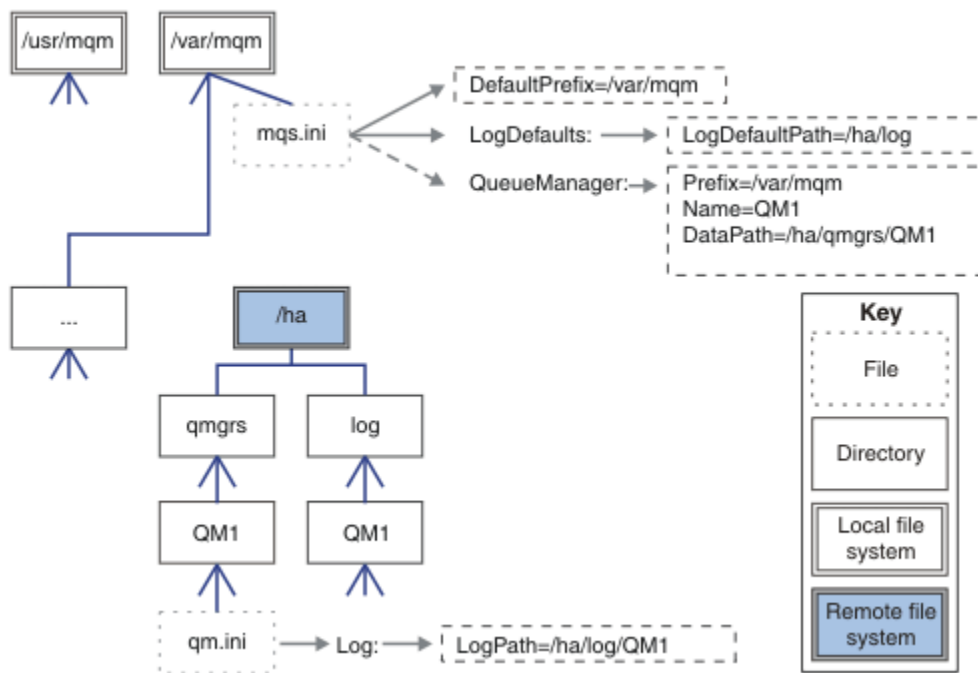


Figura 39. Compartilhar diretórios denominados `qmgrs` e `log`

Compartilhar tudo

Figura 40 na página 138 é uma configuração simples para armazenamento de arquivo de rede rápido.

Monte `/var/mqm` como um sistema de arquivo compartilhado remoto. Por padrão, ao iniciar QM1, ele procura `/var/mqm`, localiza-o no sistema de arquivos compartilhados e lê o arquivo `mqs.ini` em `/var/mqm`. Em vez de usar o arquivo `/var/mqm/mqs.ini` único para gerenciadores de filas em todos os servidores, é possível configurar a variável de ambiente `AMQ_MQS_INI_LOCATION` em cada servidor para apontar para diferentes arquivos `mqs.ini`.

Nota: O conteúdo do arquivo de erro genérico em `/var/mqm/errors/` são compartilhados entre os gerenciadores de filas em servidores diferentes.

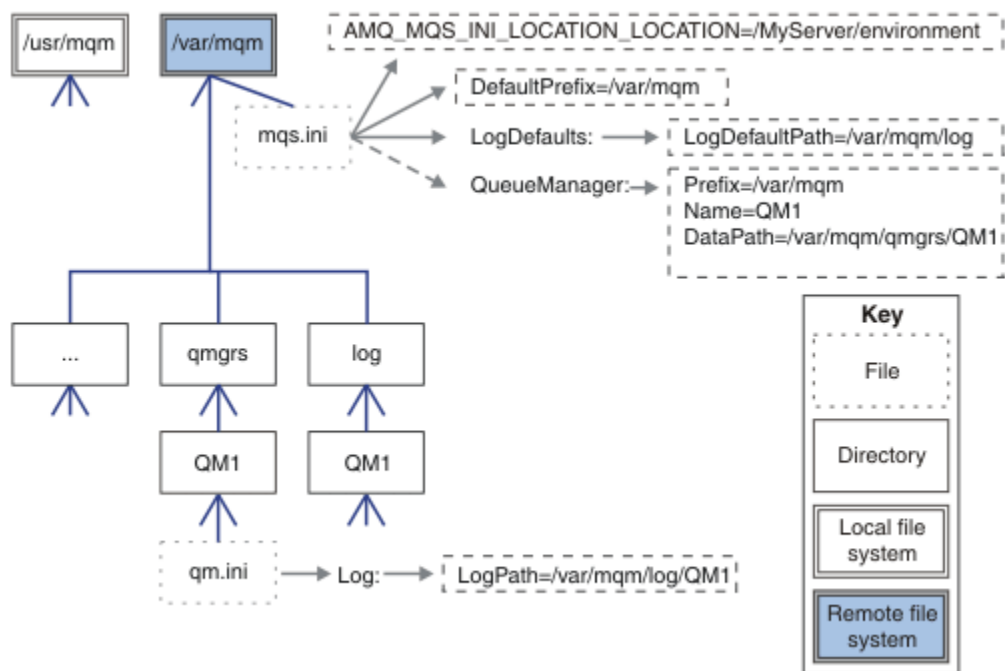


Figura 40. Compartilhar tudo

Observe que não é possível usar isto para gerenciadores de filas de várias instâncias. A razão é que é necessário que cada host em um gerenciador de filas de várias instâncias tenha sua própria cópia local de `/var/mqm` para acompanhar os dados locais, como semáforos e memória compartilhada. Essas entidades não podem ser compartilhadas entre hosts.

Windows Estrutura de Diretório em Sistemas Windows

Como localizar informações de configuração do gerenciador de filas e diretórios em Windows.

Os diretórios padrão para a instalação do IBM MQ for Windows são:

Diretório de Programa

C:\Program Files\IBM\MQ

Diretório de dados

C:\ProgramData \IBM \MQ

Importante: **Windows** Para instalações do Windows, os diretórios são conforme indicado, a menos que exista uma instalação anterior do produto que ainda contenha entradas de registro ou gerenciadores de filas ou ambos. Nesta situação, a nova instalação usa o antigo local do diretório de dados. Para obter mais informações, consulte [Locais do programa e do diretório de dados](#).

Se desejar saber qual diretório de instalação e qual diretório de dados estão sendo usados, execute o comando `dspmqr`.

O diretório de instalação é listado no campo **InstPath** e o diretório de dados é listado no campo **DataPath**.

Executar o comando **dspmqr** exibe, por exemplo, as seguintes informações:

```
>dspmqr
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

Gerenciadores de Filas de Várias Instâncias

Para configurar um gerenciador de filas de várias instâncias, os diretórios de dados e log devem ser colocados no armazenamento em rede, de preferência em um servidor diferente para quaisquer dos servidores que estiverem executando as instâncias do gerenciador de filas.

Dois parâmetros são fornecidos no comando **crtmqm, -md e -ld**, para facilitar, especifique o local dos diretórios de log e dados do gerenciador de filas. O efeito de especificar o parâmetro **-md** é quadruplicado:

1. A sub-rotina `mq.s.ini QueueManager\QmgrName` contém uma nova variável, `DataPath`, que aponta para o diretório de dados do gerenciador de filas. Diferente da variável `Prefix`, o caminho inclui o nome do diretório do gerenciador de filas.
2. As informações de configuração do gerenciador de filas armazenadas na fila `mq.s.ini` são reduzidas ao `Nome`, `Prefixo`, `Diretório` e `Caminho de Dados`.

Windows **Conteúdo de diretório**

Lista a localização e o conteúdo dos diretórios do IBM MQ.

Uma configuração do IBM MQ possui três conjuntos principais de arquivos e diretórios:

1. Executável e outros arquivos de leitura que são atualizados apenas quando a manutenção é aplicada. Por exemplo:
 - O arquivo leia-me
 - O plug-in e arquivos de ajuda do IBM MQ Explorer
 - Arquivos de LicençaEstes arquivos estão descritos em [Tabela 16 na página 139](#).
2. Os arquivos e diretórios potencialmente modificáveis que não são específicos a um determinado gerenciador de filas. Esses arquivos e diretórios são descritos em [Tabela 17 na página 140](#).
3. Arquivos e diretórios que são específicos para cada gerenciador de filas em um servidor. Esses arquivos e diretórios são descritos em [Tabela 18 na página 141](#).

Diretórios de recursos e arquivos

Os diretórios de recursos e arquivos contêm todos os códigos executáveis e recursos para executar um gerenciador de filas. A variável `FilePath`, na chave de registro da configuração do IBM MQ específica de instalação, contém o caminho para os diretórios de recursos.

Caminho de arquivo	Índice
<code>FilePath\bin</code>	Comandos e DLLs
<code>FilePath\bin64</code>	Comandos e DLLs (64 bits)
<code>FilePath\conv</code>	Tabelas de conversão de dados
<code>FilePath\doc</code>	Arquivos de ajuda do assistente
<code>FilePath\MQExplorer</code>	Explorer e plug-ins Eclipse da ajuda do Explorer
<code>FilePath\gskit8</code>	Kit de Segurança Global
<code>FilePath\java</code>	Recursos do Java, incluindo JRE

<i>Tabela 16. Diretórios e arquivos no diretório FilePath (continuação)</i>	
Caminho de arquivo	Índice
<i>FilePath\licenses</i>	Informações sobre licença
<i>FilePath\Non_IBM_License</i>	Informações sobre licença
<i>FilePath\properties</i>	Utilizadas internamente
<i>FilePath\Tivoli</i>	
<i>FilePath\tools</i>	Recursos e amostras de desenvolvimento
<i>FilePath\web</i>	Descrito na estrutura de arquivo do componente de instalação IBM MQ Console e REST API para arquivos não editáveis.
<i>FilePath\Uninst</i>	Utilizadas internamente
<i>FilePath\README.TXT</i>	Readme file

Diretórios não específicos para um gerenciador de filas

Alguns diretórios contêm arquivos, como arquivos de rastreamento e logs de erro, que não são específicos para um gerenciador de filas específico. A variável *DefaultPrefix* contém o caminho para esses diretórios. *DefaultPrefix* é parte da sub-rotina *AllQueueManagers*.

<i>Tabela 17. Diretórios e arquivos no diretório do DefaultPrefix</i>	
Caminho de arquivo	Índice
<i>DefaultPrefix\config</i>	Utilizadas internamente
<i>DefaultPrefix\conv</i>	<i>ccsid_part2.tbl</i> e <i>ccsid.tbl</i> data arquivo de controle de conversão, descrito em Conversão de dados
<i>DefaultPrefix\errors</i>	Logs de erros que não são do gerenciador de filas, AMQERR <i>nn</i> .LOG
<i>DefaultPrefix\exits</i>	Programas de Saída de Canal
<i>DefaultPrefix\exits64</i>	Programas de saída do canal (64 bits)
<i>DefaultPrefix\ipc</i>	Não utilizado
<i>DefaultPrefix\qmgrs</i>	Descrito em Tabela 18 na página 141
<i>DefaultPrefix\trace</i>	Arquivos de Rastreamento
<i>DefaultPrefix\web</i>	Descrito na estrutura do arquivo de componente de instalação IBM MQ Console e REST API para arquivos editáveis do usuário
<i>DefaultPrefix\amqmjpse.txt</i>	Utilizadas internamente

Diretórios do Gerenciador de Filas

Quando você cria um gerenciador de filas, um novo conjunto de diretórios, específico para o gerenciador de filas, é criado.

Se você criar um gerenciador de filas com o parâmetro **-md filepath**, o caminho será armazenado na variável *DataPath* na sub-rotina do gerenciador de filas do arquivo *mqs.ini*. Se você criar um gerenciador de filas sem configurar o parâmetro **-md filepath**, os diretórios do gerenciador de filas serão criados no caminho armazenado em *DefaultPrefix* o caminho será copiado na variável *Prefixo* na sub-rotina do gerenciador de filas do arquivo *mqs.ini*.

<i>Tabela 18. Diretórios e arquivos nos diretórios DataPath e Prefix\qmgrs\QmgrName</i>	
Caminho de arquivo	Índice
<i>DataPath\@ipcc</i>	Localização padrão para o AMQCLCHL . TAB, a tabela de conexão do cliente.
<i>DataPath\authinfo</i>	Usado internamente.
<i>DataPath\channel</i>	
<i>DataPath\clntconn</i>	
<i>DataPath\errors</i>	Logs de erro, AMQERR nn . LOG
<i>DataPath\listener</i>	Usado internamente.
<i>DataPath\namelist</i>	
<i>DataPath\plugcomp</i>	
<i>DataPath\procdef</i>	
<i>DataPath\qmanager</i>	
<i>DataPath\queues</i>	
<i>DataPath\services</i>	
<i>DataPath\ssl</i>	
<i>DataPath\startprm</i>	
<i>DataPath\topic</i>	
<i>DataPath\active</i>	
<i>DataPath\active.dat</i>	
<i>DataPath\amqalchk.fil</i>	
<i>DataPath\master</i>	
<i>DataPath\master.dat</i>	
<i>DataPath\qm.ini</i>	Configuração do Gerenciador de Filas
<i>DataPath\qmstatus.ini</i>	Status do gerenciador de filas
<i>DataPath\userdata</i>	Pode ser usado para armazenar o estado persistente dos aplicativos.
<i>Prefix\qmgrs\QmgrName</i>	Utilizadas internamente
<i>Prefix\qmgrs\@SYSTEM</i>	Não utilizado
<i>Prefix\qmgrs\@SYSTEM\errors</i>	
<i>DataPath\autocfg</i>	Usado para configuração automática

Estrutura de diretório no IBM i

Uma descrição do IFS é fornecida, e a estrutura de diretório IFS do IBM MQ é descrita para servidor, cliente e Java.

O sistema de arquivos integrado (IFS) faz parte do IBM i que suporta a entrada/saída do fluxo e o gerenciamento de armazenamento semelhante ao computador pessoal, sistemas operacionais AIX and Linux, enquanto fornece uma estrutura de integração sobre todas as informações armazenadas no servidor.

No IBM i, os nomes de diretórios começam com o caractere & (ampersand) em vez do caractere @ (at). Por exemplo, @system em IBM i é &system.

Sistema de Arquivo Raiz IFS para o Servidor IBM MQ

Ao instalar o IBM MQ Server para IBM i, os seguintes diretórios são criados no sistema de arquivos raiz IFS.

ProdData:

Visão Geral

QIBM

```
'-- ProdData
    '-- mqm
    '-- doc
    '-- inc
    '-- lib
    '-- samp
    '-- licenses
    '-- LicenseDoc
    '-- 5724H72_V8R0M0
```

/QIBM/ProdData/mqm

Os subdiretórios abaixo disso contêm todos os dados do produto, por exemplo, classes C++, arquivos de formato de rastreamento e arquivos de licença. Os dados neste diretório são excluídos e substituídos sempre que o produto estiver instalado.

/QIBM/ProdData/mqm/doc

Uma Referência de Comando para os comandos CL é fornecida no formato HTML e instalada aqui.

/QIBM/ProdData/mqm/inc

Os arquivos de cabeçalho para compilar seus programas C ou C++.

/QIBM/ProdData/mqm/lib

Os arquivos auxiliares usados pelo MQ.

/QIBM/ProdData/mqm/samp

Amostras adicionais.

/QIBM/ProdData/mqm/licenses

Arquivos de licença. Os dois arquivos para cada idioma são nomeados como LA_ *xx* e LI_ *xx* , em que *xx* é o identificador de idioma de 2 caracteres para cada idioma fornecido.

Além disso, o seguinte diretório armazena os arquivos dos contratos de licença:

/QIBM/ProdData/LicenseDoc/5724H72_V8R0M0

Arquivos de licença. Os arquivos são nomeados como 5724H72_V8R0M0_ *xx* , em que *xx* é o identificador de idioma de 2 ou 5 caracteres para cada idioma fornecido.

UserData:

Visão Geral

QIBM

```
'-- UserData
    '-- mqm
    '-- errors
    '-- trace
    '-- qmgrs
    '-- &system
    '-- qmgrname1
    '-- qmgrname2
```

'-- and so on

/QIBM/UserData/mqm

Os subdiretórios abaixo deste contêm todos os dados de usuário relacionados aos gerenciadores de filas.

Ao instalar o produto, um arquivo mqs.ini é criado no diretório /QIBM/UserData/mqm/ (a menos que há esteja nesse local a partir de uma instalação anterior).

Ao criar um gerenciador de filas, um arquivo qm.ini é criado no diretório /QIBM/UserData/mqm/qmgrs/ *QMGRNAME* / (em que *QMGRNAME* é o nome do gerenciador de filas).

Dados nos diretórios são retidos quando o produto é excluído.

Sistema de arquivos raiz IFS para o IBM MQ MQI client

Ao instalar o IBM MQ MQI client for IBM i, os seguintes diretórios criados no sistema de arquivos raiz do IFS:

ProdData:

Visão Geral

QIBM

```
'-- ProdData
    '-- mqm
    '-- lib
```

/QIBM/ProdData/mqm

Os subdiretórios abaixo deste diretório contêm todos os dados do produto. Os dados neste diretório são excluídos e substituídos sempre que o produto é substituído.

UserData:

Visão Geral

QIBM

```
'-- UserData
    '-- mqm
    '-- errors
    '-- trace
```

/QIBM/UserData/mqm

Os subdiretórios abaixo deste diretório contêm todos os dados do usuário.

Sistema de arquivos raiz IFS para o IBM MQ Java

Ao instalar o IBM MQ Java no IBM i, os seguintes diretórios são criados no sistema de arquivos raiz do IFS:

ProdData:

Visão Geral

QIBM

```
'-- ProdData
    '-- mqm
    '-- java
    '-- samples
    '-- bin
    '-- lib
```

/QIBM/ProdData/mqm/java

Os subdiretórios abaixo disso contêm todos os dados do produto, incluindo as classes do Java. Os dados neste diretório são excluídos e substituídos sempre que o produto é substituído.

/QIBM/ProdData/mqm/java/samples

Os subdiretórios abaixo deste contêm todas as classes de Java de amostra e dados.

Bibliotecas Criadas pelas Instalações de Cliente e Servidor

A instalação do servidor ou do cliente IBM MQ cria as seguintes bibliotecas:

- QMQM

A biblioteca do produto.

- QMQMSAMP

A biblioteca de amostras (se você optar por instalar as amostras).

- QMxxxx

Apenas servidor.

Cada vez que você cria um gerenciador de filas, o IBM MQ cria automaticamente uma biblioteca associada, com um nome como QMxxxx, em que xxxx é derivado do nome do gerenciador de filas. Esta biblioteca contém objetos específicos ao gerenciador de filas, incluindo diários e receptores associados. Por padrão, o nome desta biblioteca é derivado do nome do gerenciador de filas com prefixo dos caracteres QM. Por exemplo, para um gerenciador de filas denominado TEST, a biblioteca seria chamada QMTEST.

Nota: Quando você cria um gerenciador de filas, é possível especificar o nome de sua biblioteca, se desejar. Por exemplo:

```
CRTMQM MQMNAME(TEST) MQMLIB(TESTLIB)
```

É possível usar o comando WRKLIB para listar todas as bibliotecas que o IBM MQ para o IBM i criou. Nas bibliotecas do gerenciador de filas, você verá o texto QMGR: QMGRNAME. O formato do comando é:

```
WRKLIB LIB(QM*)
```

Essas bibliotecas associadas ao gerenciador de filas são retidas quando o produto é excluído.

Multi **Planejando o suporte do sistema de arquivos para MFT em Multiplataformas**

Os agentes do IBM MQ Managed File Transfer MFT podem ser usados para transferir dados para e de arquivos em um sistema de arquivos. Além disso, monitores de recursos em execução em um agente podem ser configurados para monitorar arquivos em um sistema de arquivos.

MFT tem um requisito de que esses arquivos sejam armazenados em um sistema de arquivos que suporte bloqueio. Há duas razões para isso:

- Um agente bloqueia um arquivo para assegurar que ele não seja alterado após ter iniciado a leitura de dados a partir dele ou a gravação de dados nele.
- O recurso monitora os arquivos de bloqueio para verificar se nenhum outro processo está usando atualmente.

Os agentes e os monitores de recursos usam o Java método **FileChannel.tryLock()** para executar o bloqueio e o sistema de arquivos deve ser capaz de bloquear arquivos quando solicitado a fazer isso usando essa chamada

Importante: Os sistemas de arquivos a seguir não são suportados, pois não atendem aos requisitos técnicos do MFT:

- GlusterFS
- NFS versão 3

Multi **Escolhendo a criação de log circular ou linear em multiplataformas**

No IBM MQ, é possível escolher a criação de log circular ou linear. As informações a seguir fornecem uma visão geral de ambos os tipos.

Vantagens da criação de log circular

As principais vantagens da criação de log circular são que ela é:

- Mais fácil de administrar.

Depois que a criação de log circular é configurada corretamente para sua carga de trabalho, não há necessidade de administração adicional. No entanto, para a criação de log linear, as imagens de mídia precisam ser registradas e as extensões de log que não são mais necessárias precisam ser arquivadas ou excluídas.

- Melhor desempenho

A execução da criação de log circular é melhor do que a criação de log linear, porque a criação de log circular é capaz de reutilizar as extensões de log que já foram formatadas. A criação de log linear, por outro lado, precisa alocar novas extensões de log e formatá-las.

Veja [Gerenciando logs](#) para obter informações adicionais.

Vantagens da criação de log linear

A principal vantagem da criação de log linear é que ela fornece proteção contra mais falhas.

Nem a criação de log circular nem a criação de log linear protegem contra um log corrompido ou excluído ou mensagens ou filas que foram excluídas por aplicativos ou pelo administrador.

A criação de log linear (mas não circular) permite que objetos danificados sejam recuperados. Portanto, a criação de log linear fornece proteção para que arquivos de fila não sejam corrompidos ou excluídos, pois essas filas danificadas podem ser recuperadas de um log linear.

Tanto a circular quanto a linear protegem contra perda de energia e falha na comunicação, conforme descrito em [Recuperando-se de perda de energia ou de falhas na comunicação](#).

Outras contraprestações

A escolha de linear ou circular depende da quantidade de redundância necessária.

Há um custo na escolha de mais redundância, que é a criação de log linear, causada pelo custo de desempenho e pelo custo de administração.

Veja [Tipos de criação de log](#) para obter mais informações.

AIX **memória compartilhada no AIX**

Se determinados tipos de aplicativo falham ao conectar-se devido a uma limitação de memória do AIX, na maioria dos casos, isso pode ser resolvido configurando a variável de ambiente EXTSHM=ON.

Alguns processos de 32 bits no AIX podem encontrar uma limitação do sistema operacional que afeta sua capacidade de se conectar a gerenciadores de filas do IBM MQ. Cada conexão padrão para o IBM MQ usa memória compartilhada, mas diferentemente de outras plataformas UNIX, o AIX permite que processos de 32 bits para anexar apenas 11 conjuntos de memória compartilhada.

A maioria dos processos de 32 bits não encontrará esse limite, mas aplicativos com altos requisitos de memória podem falhar ao se conectar ao IBM MQ com o código de razão 2102: MQRC_RESOURCE_PROBLEM. Os seguintes tipos de aplicativo podem ver este erro:

- Programas que executam em máquinas virtuais do Java de 32 bits
- Programas que usam os modelos de memória grandes ou muito grandes
- Programas que se conectam a vários gerenciadores de filas ou bancos de dados
- Programas que se conectam a conjuntos de memória compartilhada por conta própria

O AIX oferece um recurso de memória compartilhada estendida para processos de 32 bits que permitem que eles anexem mais memória compartilhada. Para executar um aplicativo com esse recurso, exporte a variável de ambiente EXTSHM=ON antes de iniciar seus gerenciadores de filas e seu programa. O recurso EXTSHM=ON evita este erro na maioria dos casos, mas ele é incompatível com programas que usam a opção SHM_SIZE da função shmctl.

Os aplicativos do IBM MQ MQI client e todos os processos de 64 bits não são afetados por essa limitação. Eles podem se conectar aos gerenciadores de filas do IBM MQ, independentemente de EXTSHM ter sido configurado.

Linux

AIX

Recursos IPC do IBM MQ e UNIX System V

Um gerenciador de filas usa alguns recursos IPC. Use o `ipcs -a` para descobrir quais recursos estão sendo usados

Essas informações se aplicam ao IBM MQ em execução em sistemas AIX and Linux apenas.

O IBM MQ usa os recursos de comunicação interprocessual (IPC) do System V (*semáforos e segmentos de memória compartilhada*) para armazenar e transmitir dados entre componentes do sistema. Esses recursos são usados pelos processos e aplicativos do gerenciador de filas que se conectam ao gerenciador de filas. O IBM MQ MQI clients não usa recursos IPC, exceto para controle de rastreamento do IBM MQ. Use o comando UNIX `ipcs -a` para obter informações completas sobre o número e o tamanho dos recursos do IPC atualmente em uso na máquina

Linux

AIX

Prioridade de processo do IBM MQ e do UNIX

Boas práticas ao configurar os valores *nice* de prioridade do processo.

Essas informações se aplicam ao IBM MQ em execução em sistemas AIX and Linux apenas.

Se você executar um processo em segundo plano, esse processo pode receber um valor *nice* mais alto (e portanto prioridade inferior) pelo shell de chamada. Isso pode ter implicações no desempenho geral do IBM MQ. Em situações com alto stress, se houver vários encadeamentos prontos para execução em uma prioridade mais alta e alguns em uma prioridade inferior, as características de planejamento do sistema operacional podem remover os encadeamentos de prioridade inferior do tempo do processador.

É uma boa prática que processos independentemente iniciados associados a gerenciadores de filas, como `runmqtsr`, tenham os mesmos valores *nice* que o gerenciador de filas com o qual estão associados. Certifique-se de que o shell não designe um valor *nice* mais alto para esses processos de segundo plano. Por exemplo, em ksh, use a configuração "set +o bgnice" para parar o ksh de levantar o valor *nice* dos processos de segundo plano.. É possível verificar os valores *nice* dos processos em execução, examinando a coluna *NI* de uma listagem "ps -efl".

Além disso, inicie os processos de aplicativo do IBM MQ com o mesmo valor *nice* que o gerenciador de filas. Se eles estiverem sendo executados com valores *nice* diferentes, um encadeamento de aplicativos pode bloquear um encadeamento do gerenciador de filas ou vice-versa, fazendo com que o desempenho seja prejudicado.

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

[“Planejando uma arquitetura do IBM MQ” on page 5](#)

Ao planejar seu ambiente do IBM MQ, considere o suporte que o IBM MQ fornece para arquiteturas de gerenciador de filas único e múltiplo e para estilos de sistema de mensagem ponto a ponto e publicar/ assinar. Além disso, planeje os requisitos de recursos e sua utilização de recursos de criação de log e backup.

[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

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 149](#)

- [“System LX usage” on page 149](#)
- [“Storage configuration” on page 150](#)
- [“Disk storage” on page 155](#)

See, [“Where to find more information about storage and performance requirements” on page 155](#) 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 150](#) for more information.

Library storage

You must allocate disk storage for the product libraries. The exact figures depend on your configuration, and should include both the target and distribution libraries, as well as the SMP/E libraries.

The target libraries used by IBM MQ for z/OS use PDSE formats. Ensure that any PDSE target libraries are not shared outside a sysplex. For more information about the required libraries and their sizes and the required format, see the Program Directory. Para links de download para os diretórios do programa, consulte [IBM MQ for z/OS Arquivos PDF do diretório do programa](#) .

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.

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.

Storage configuration

 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.




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 151
- “Channel initiator storage configuration from IBM MQ 9.4.0” on page 153

Table 19. Suggested definitions for REGION and MEMLIMIT	
Address space	Storage configuration
Queue manager	REGION=0M, MEMLIMIT=3G
 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.

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**.

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.

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.

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 174](#) 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 191](#) describes how to determine how much storage Db2 requires for the IBM MQ data.

- **coupling facility storage**

[“Defining coupling facility resources” on page 181](#) 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 156](#) 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.

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:

Topic	Where to look
System parameters	Using CSQ6SYSP and Customizing your queue managers
Storage required to install IBM MQ	Program Directory. Para links de download para os diretórios do programa, consulte IBM MQ for z/OS Arquivos PDF do diretório do programa .
IEALIMIT and IEFUSI exits	See IEALIMIT and IEFUSI in the <i>z/OS:MVS Installation Exits</i> documentation.

Table 20. Where to find more information about storage requirements (continued)

Topic	Where to look
Latest information	IBM MQ SupportPac website SupportPacs para IBM MQ e outras áreas do projeto .
Workload management and defining goals through the service definition	z/OS MVS Planning: Workload Management

Planning your page sets and buffer pools

Information to help you with planning the initial number, and sizes of your page data sets, and buffer pools.

This topic contains the following sections:

- [“Plan your page sets” on page 156](#)
 - [Page set usage](#)
 - [Number of page sets](#)
 - [Size of page sets](#)
 - [Planning for z/OS data set encryption](#)
- [“Calculate the size of your page sets” on page 157](#)
 - [Page set zero](#)
 - [Page set 01 - 99](#)
 - [Calculating the storage requirement for messages](#)
- [“Enabling dynamic page set expansion” on page 159](#)
- [“Defining your buffer pools” on page 161](#)

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 161](#) 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 159](#). 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 193](#).

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.

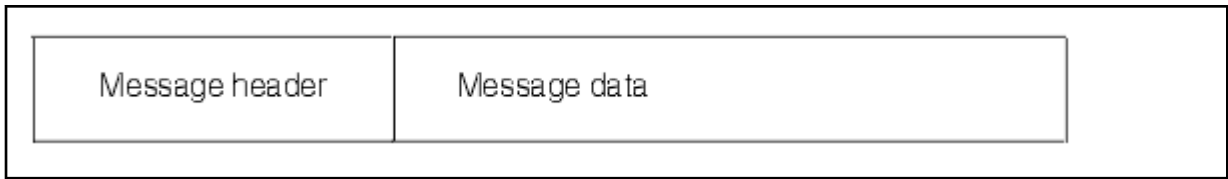
Note: The sizes of the structures and control information given in this section are liable to change between major releases. For details specific to your release of IBM MQ, refer to SupportPac [MP16 - IBM MQ para z/OS Planejamento e ajuste de capacidade](#) and [IBM MQ Família-Relatórios de Desempenho](#)

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 159:

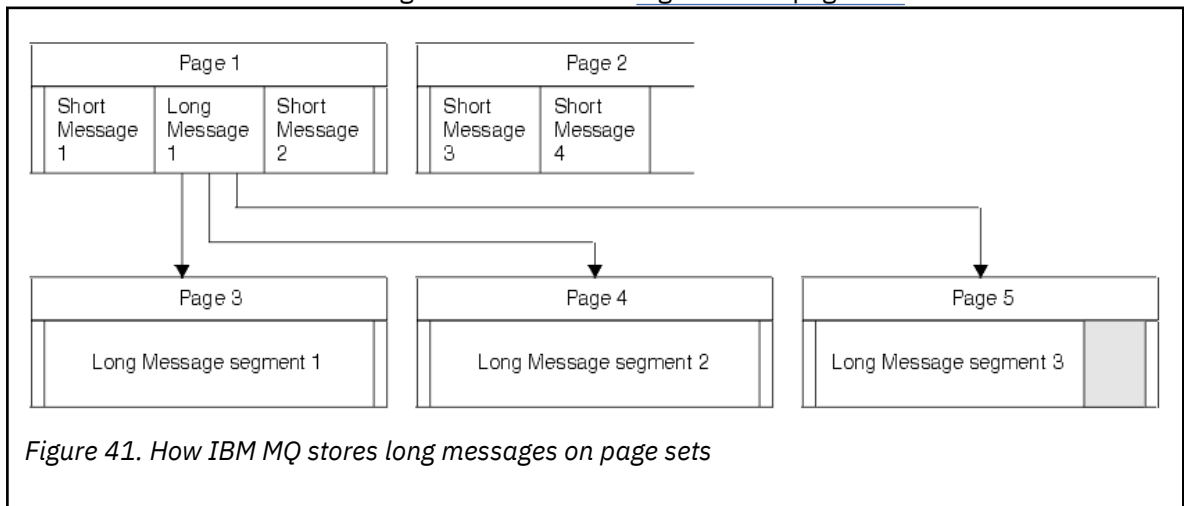


Figure 41. How IBM MQ stores long messages on page sets

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 161](#)
2. [“Decide on the settings for each buffer pool” on page 162](#)
3. [“Monitor the performance of buffer pools under expected load” on page 162](#)
4. [“Adjust buffer pool characteristics” on page 162](#)

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 163](#) 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 161, 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

<i>Table 21. Suggested definitions for buffer pool settings</i>		
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 162, if required.

Use the performance statistics from [“Monitor the performance of buffer pools under expected load”](#) on page 162 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 para z/OS Planejamento e ajuste de capacidade](#) provides further information about tuning buffer pools.

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:

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 165](#)
- [Active log placement](#)

- [“Active log encryption with z/OS data set encryption”](#) on page 166

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 175 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.

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

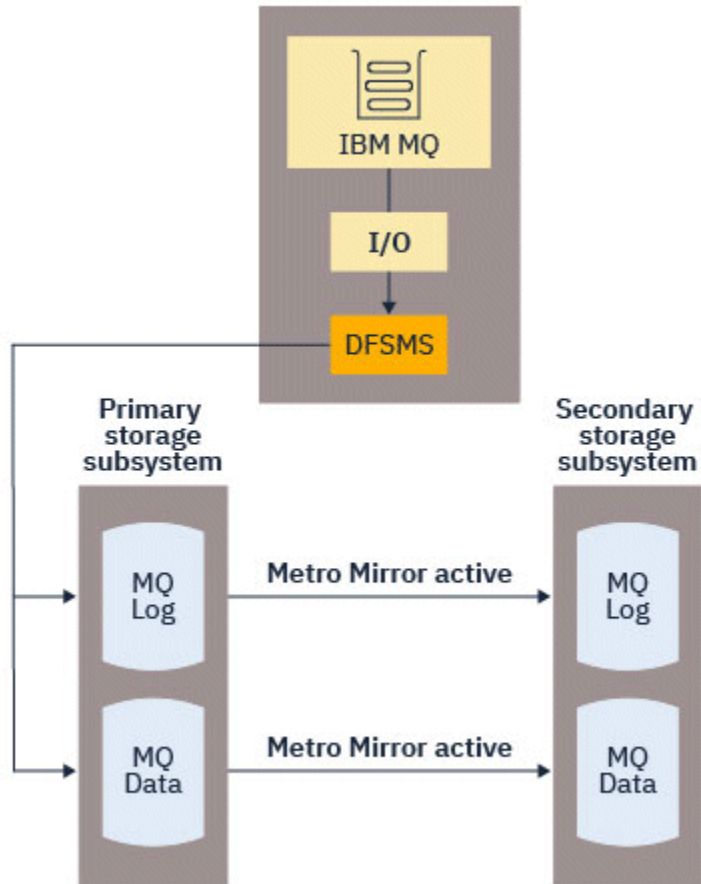
All of the following IBM MQ data set types can be replicated using Metro Mirror. However, exactly which ones are replicated depends on the availability requirements of your enterprise:

- 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.

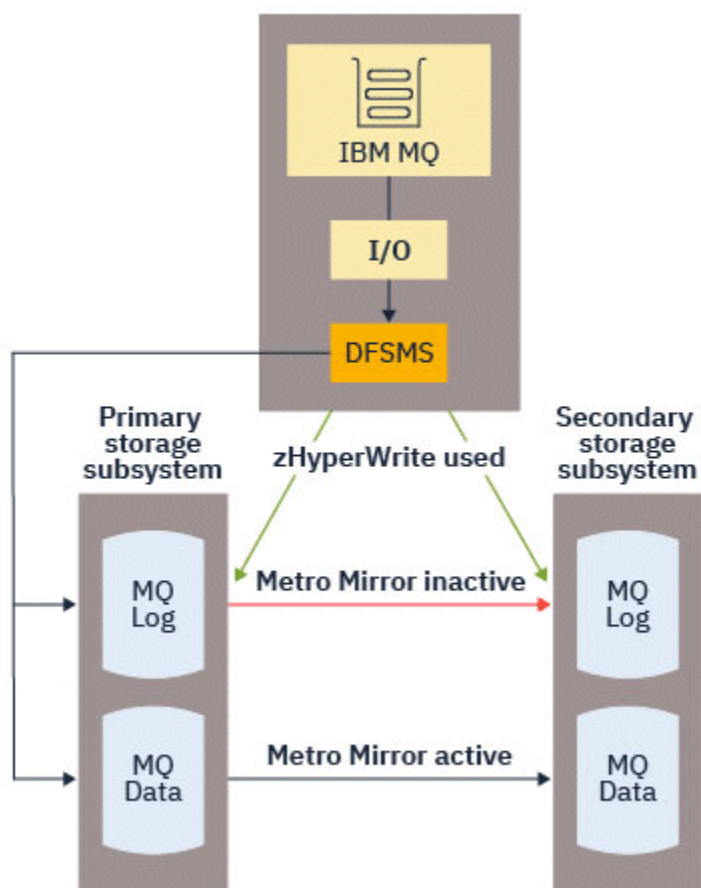


Because both writes to the primary and secondary storage subsystems need to complete before the write returns to IBM MQ, use of Metro Mirror can have a performance impact. You need to balance this performance impact against the availability benefits of using Metro Mirror.

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
```

Rendimento de log mais rápido com o zHyperLink

A tecnologia zHyperLink foi projetada para reduzir a latência de entrada / saída (E/S) fornecendo um caminho de comunicação rápido, confiável e direto entre a CPU e o dispositivo de E/S.

Visão Geral do Link zHyper

O zHyperLink pode melhorar o rendimento do log ativo e reduzir o tempo de transação do IBM MQ em até 3.5 vezes. Esse objetivo é realizado instalando adaptadores de link zHyper no host z/OS, selecionando o hardware de armazenamento IBM e conectando-os usando cabos de link zHyper. Isso cria uma conexão ponto a ponto entre a CPU e o dispositivo de E/S, que reduz o tempo de resposta de E/S em até 10 vezes, em comparação com o IBM z High-Performance FICON® (zHPF). Esse tempo de resposta baixo é obtido usando solicitações de E/S síncronas.

As vantagens de E/S Síncrona sobre E/S Assíncronas

A tarefa do criador de logs IBM MQ consiste em um loop aguardando a próxima parte de dados que precisa ser gravada no log. Quando esses dados estiverem disponíveis, o criador de logs planeja a gravação, aguarda sua conclusão e, em seguida, move para a próxima parte de dados.

A E/S tradicional é mais lenta que a CPU, portanto é mais eficiente executar a E/S de forma assíncrona para liberar a CPU para outras tarefas. Portanto, a E/S assíncrona tradicional requer que a tarefa do criador de logs seja suspensa até a conclusão da gravação. Quando a gravação for concluída, a tarefa do criador de logs deverá aguardar que uma CPU se torne disponível, incluindo um curto atraso de novo despacho, bem como atrasos causados pelo reenchimento do cache da CPU.

O Link zHyper fornece tempos de E/S muito mais rápidos, que estão mais próximos da velocidade da CPU, portanto, com o Link zHyper, a E/S pode ser executada de forma síncrona, o que significa que a tarefa do criador de logs não é suspensa durante a operação de gravação, removendo atrasos relacionados ao novo despacho e ao cache.

Enquanto a gravação está acontecendo, a tarefa do criador de logs ainda está usando ativamente a CPU, que aumenta o uso da CPU em comparação com a E/S tradicional.

Se o gerenciador de filas tentar usar o link zHyper e a gravação do link zHyper falhar, por exemplo, por causa de problemas de configuração, o gerenciador de filas efetuará fallback de forma transparente para a E/S tradicional.

Requisitos mínimos de hardware

- IBM z14 ou mais recente

- DS8880 ou posterior

Requisitos de Software

- zHyperLink Express é suportado no z/OS 2.3 ou posterior.
- A imagem z/OS deve ser executada em uma LPAR, não como um convidado em IBM z/VM®..
- O zHyperLink requer que IBM z High-Performance FICON (zHPF) seja ativado.

Usando o zHyperLink com logs ativos do IBM MQ

Para usar o zHyperLink com os registros ativos de um gerenciador de filas, é necessário:

- Configure o IBM MQ para usar o zHyperLink e
- Certifique-se de que os logs ativos estejam nos volumes com capacidade de zHyperLink.

Consulte [Introdução ao IBM zHyperLink para z/OS](#) para obter mais informações..

É possível configurar o IBM MQ para usar o zHyperLink usando um dos métodos a seguir:

- Especifique `ZHYLINK(YES)` nos parâmetros de log
- Emita o comando `SET LOG ZHYLINK(YES)`.

Notas:

- zHyperLink requer que zHyperWrite esteja ativado. Isso significa que, para usar `ZHYLINK`, `ZHYWRITE` também deve ser ativado nos parâmetros de log. Ao especificar apenas `ZHYLINK (YES)` quando `ZHYWRITE (NO)` é configurado no gerenciador de filas, o parâmetro `ZHYWRITE` substitui automaticamente para `YES`.
- Tentar explicitamente configurar `ZHYLINK (YES)` com `ZHYWRITE (NO)` resulta em uma conclusão anormal do comando `SET LOG`.
- Configurar `ZHYLINK=YES` nos ZPRMs substitui `ZHYWRITE` em `YES`.

Se você estiver tendo problemas, consulte [Resolução de problemas zHyperLink](#) para obter mais informações

O IBM MQ não requer que todos os conjuntos de dados do log ativo estejam em volumes compatíveis com zHyperLink, mas é aconselhável fazer isso. Se o IBM MQ detectar que alguns conjuntos de dados do log ativo estão em volumes com capacidade de link zHyper outros não, ele emitirá a mensagem CSQJ601E e executará o processamento.

IBM MQ verifica se os conjuntos de dados do log ativo são zHypercompatíveis quando os conjuntos de dados são abertos pela primeira vez. Os conjuntos de dados do log são abertos na inicialização do gerenciador de fila ou ao incluir dinamicamente usando o comando `DEFINE LOG`. Se os conjuntos de dados de log forem feitos compatíveis com o link zHyper enquanto um gerenciador de filas os tiver aberto, o gerenciador de filas não detectará isso até que ele tenha sido reiniciado

Se `ZHYLINK (YES)` for especificado, o IBM MQ sempre tentará usar o zHyperLink ao gravar nos logs ativos, independentemente de os logs estarem nos volumes com capacidade para zHyperLink. Não há efeitos negativos de tentar usar o link zHyper se os logs não estiverem nos volumes com capacidade de link do zHyper

É possível usar a saída do comando `DISPLAY LOG` para indicar o status do zHyperLink para os atuais conjuntos de dados do log ativo:

```
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
```

O status do link zHyperé um dos seguintes:

SIM

O link zHyper está ativado no gerenciador de fila e será tentado em todas as gravações

NÃO

zHyperLink não está ativado no gerenciador de fila e o conjunto de dados **não é** em zHyper volumes compatíveis com Link.

capaz

zHyper O link não está ativado no gerenciador de filas e no conjunto de dados **é** em um volume compatível com zHyperLink.

Há várias estatísticas de SMF adicionais para monitorar e entender o desempenho do link do zHyper; consulte [zHyper Estatísticas de link](#) para obter detalhes

Sessões de gravação

Ao usar o link zHyper, uma ou mais sessões de gravação de link do zHyper são estabelecidas com o DASD O DASD atual suporta um máximo de 64 sessões de gravação simultânea, portanto, você deve considerar cuidadosamente quais gerenciadores de filas ativar o link zHyper se outros subsistemas, como Db2, também estão usando o link zHyper para gravar no mesmo DASD. Se você ficar sem sessões de gravação disponíveis, o gerenciador de filas alternará automaticamente de volta para usar E/S assíncronas tradicionais.

É possível calcular o número de sessões de gravação do zHyperLink conforme a seguir:

```
Number of log copies (either 1 or 2) * number of stripes per log copy * 2  
if Metro Mirror (PPRC) is used.
```

Portanto, um gerenciador de filas no modo de criação de log único com uma faixa, e nenhum Metro Mirror usa uma única sessão de gravação Um gerenciador de fila no modo de criação de log dual, com duas faixas e PPRC usa 8 sessões de gravação.

Nota: Enquanto Metro Mirror resulta em duas vezes mais sessões de gravação sendo usadas, essas sessões de gravação são divididas uniformemente entre os dois DASDs espelhados.

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 173 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 DFSMS dynamically allocates during aggregate backup processing](#).

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 172 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 172, 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.

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.

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](#)

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 (VIP). 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 178](#) 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 [“Planejando filas distribuídas e clusters” on page 20](#) for more information.

Actions before you configure the channel initiator

1. Decide if you are using TCP/IP or APPC.
2. If you are using TCP, allocate at least one port for IBM MQ.
3. If you need a a DNS server, configure the server to be highly available if required.
4. If you are using APPC, allocate an LU name, and configure APPC.

Actions after you have configured the channel initiator, before you go into production

1. Plan what connections you will have:
 - a. Client connections from remote applications.
 - b. MCA channels to and from other queue managers. Typically you have a channel to and from each remote queue manager.
2. Set up clustering, or join an existing clustering environment.

3. Consider whether you need to use multiple TCP stacks, VIPA, or an external router for availability in front of the channel initiator.
4. If you are planning on using TLS:
 - a. Set up the key ring
 - b. Set up certificates
5. If you are planning on using channel authentication:
 - a. Decide the criteria for mapping inbound sessions to MCA user IDs
 - b. Enable reverse DNS lookup by setting the queue manager parameter **REVDNS**
 - c. Review security. For example, delete the default channels, and specify user IDs with only the necessary authority in the **MCAUSER** attribute for a channel.
6. Capture the accounting and statistics SMF records produced by the channel initiator and post process them.
7. Automate the monitoring of job log messages.
8. If necessary, tune your network environment to improve throughput. With TCP, large send and receive buffers improve throughput. You can force MQ to use specific TCP buffer sizes using the commands:

```
RECOVER QMGR(TUNE CHINTCPBDYNSZ 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 147](#)

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.

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](#)

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 TCPSENDBFRSIZE 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 qmqzName . 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 147](#)

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.

Planning your coupling facility and offload storage environment

Use this topic when planning the initial sizes, and formats of your coupling facility (CF) structures, and shared message data set (SMDS) environment or Db2 environment.

This section contains information about the following topics:

- [“Defining coupling facility resources” on page 181](#)
 - [Deciding your offload storage mechanism](#)
 - [Planning your structures](#)
 - [Planning the size of your structures](#)
 - [Mapping shared queues to structures](#)
- [“Planning your shared message data set \(SMDS\) environment” on page 187](#)
- [“Planning your Db2 environment” on page 190](#)

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-nameCSQ_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-nameCSQSYSAPPL* 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-nameCSQSYAPPL* 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-nameCSQSYSAPPL* structure. You can use up to 63 (62 if *qsg-nameCSQSYSAPPL* 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 be 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 MQ 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 183 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.

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

<i>Table 22. Minimum administrative structure sizes (continued)</i>	
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 183](#). 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

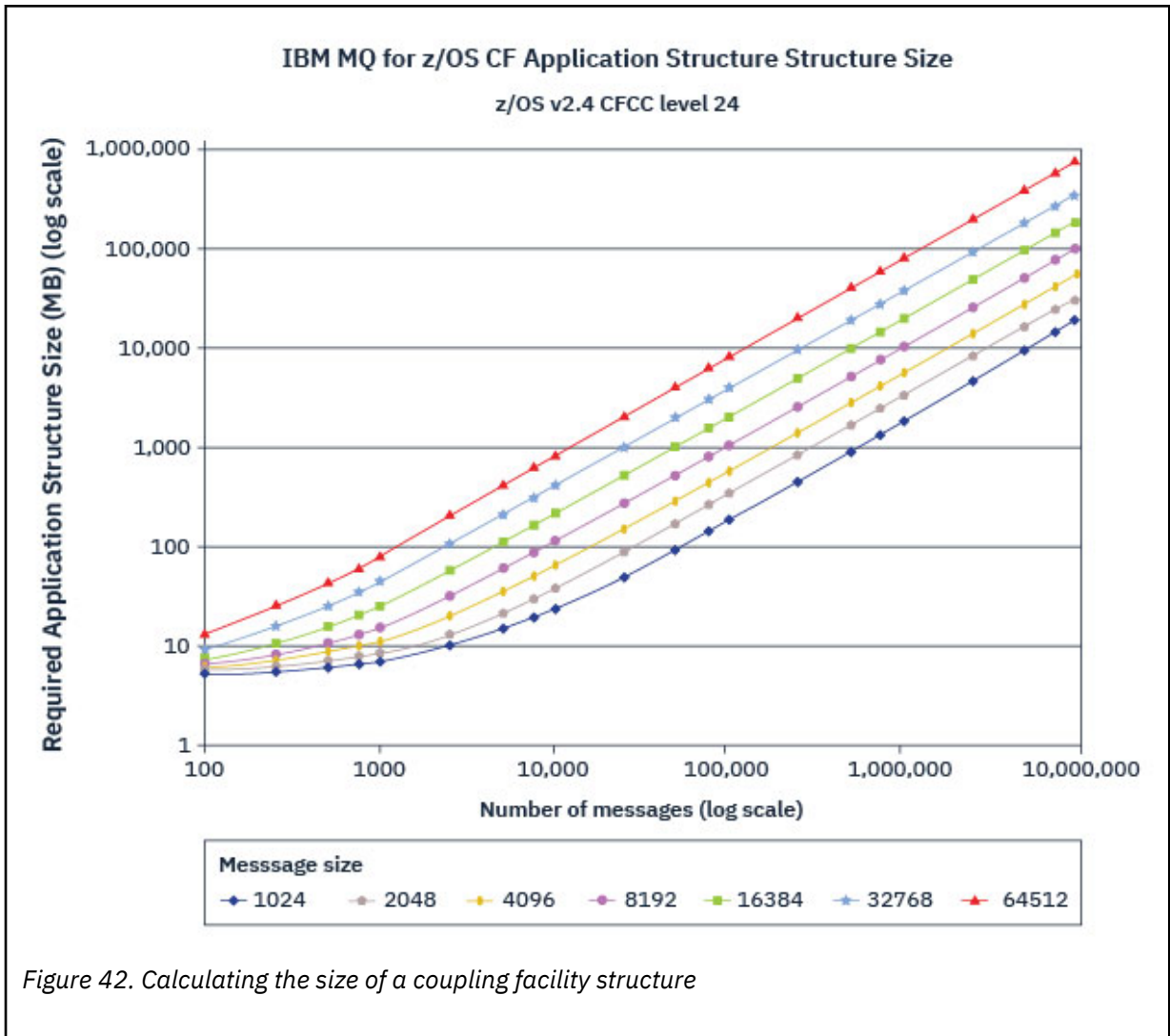
The size of the application structures required to hold IBM MQ messages depends on the likely number and size of the messages to be held on a structure concurrently.

The graph in [Figure 42 on page 185](#) 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 185](#) provides the same information in tabular form.



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.

<i>Table 24. Table showing CSQSYSAPPL usage against sizing.</i>	
<i>qsg-name</i> 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-nameAPPLICATION1* 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)
```

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.

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*.

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	<ul style="list-style-type: none"> • 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. 	<ul style="list-style-type: none"> • A queue with the QSGDISP(GROUP) attribute is deleted. • A queue with the QSGDISP(SHARED) attribute is deleted. • A dynamic queue with the DEFTYPE(SHAREDYN) attribute is closed with the DELETE option.
CSQ.OBJ_B_NAMELIST	Up to 15127 bytes	A namelist with the QSGDISP(GROUP) attribute is defined.	A namelist with the QSGDISP(GROUP) attribute is deleted.
CSQ.OBJ_B_CHANNEL	Up to 14127 bytes	A channel with the QSGDISP(GROUP) attribute is defined.	A channel with the QSGDISP(GROUP) attribute is deleted.
CSQ.OBJ_B_STGCLASS	Up to 2865 bytes	A storage class with the QSGDISP(GROUP) attribute is defined.	A storage class with the QSGDISP(GROUP) attribute class is deleted.
CSQ.OBJ_B_PROCESS	Up to 3347 bytes	A process with the QSGDISP(GROUP) attribute is defined.	A process with the QSGDISP(GROUP) attribute is deleted.
CSQ.OBJ_B_TOPIC	Up to 14520 bytes	A topic object with QSGDISP(GROUP) attribute is defined.	A topic object with QSGDISP(GROUP) attribute is deleted.
CSQ.EXTEND_B_QMGR	Less than 430 bytes	A queue manager is added to the table with the ADD QMGR function of the CSQ5PQSG utility.	A queue manager is removed from the table with the REMOVE QMGR function of the CSQ5PQSG utility.
CSQ.ADMIN_B_MESSAGES	87 bytes	For large message PUT (1 per BLOB).	For large message GET (1 per BLOB).

Table 25. Planning your Db2 storage requirements (continued)

Db2 table name	Length of row	A row is added when:	A row is deleted when:
CSQ.ADMIN_MSGS_BAUX1 CSQ.ADMIN_MSGS_BAUX2 CSQ.ADMIN_MSGS_BAUX3 CSQ.ADMIN_MSGS_BAUX4		These 4 tables contain message payload for large messages added into one of these 4 tables for each BLOB of the message. BLOBS are up to 511 KB in length, so if the message size is > 711 KB, there will be multiple BLOBs for this message.	

The use of large numbers of shared queue messages of size greater than 63 KB can have significant performance implications on your IBM MQ system. For more information, see SupportPac MP16, Capacity Planning and Tuning for IBM MQ for z/OS, at: [SupportPacs for IBM MQ and other project areas](#).

z/OS Planning for backup and recovery

Developing backup and recovery procedures at your site is vital to avoid costly and time-consuming losses of data. IBM MQ provides means for recovering both queues and messages to their current state after a system failure.

This topic contains the following sections:

- [“Recovery procedures” on page 193](#)
- [“Tips for backup and recovery” on page 194](#)
- [“Recovering page sets” on page 196](#)
- [“Recovering CF structures” on page 197](#)
- [“Achieving specific recovery targets” on page 197](#)
- [“Backup considerations for other products” on page 199](#)
- [“Recovery and CICS” on page 199](#)
- [“Recovery and IMS” on page 200](#)
- [“Preparing for recovery on an alternative site” on page 200](#)
- [“Example of queue manager backup activity” on page 200](#)

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 [Administrando IBM MQ for z/OS](#) for information about these.

Become familiar with the procedures used at your site for the following:

- Recovering from a hardware or power failure.
- Recovering from a z/OS component failure.

- Recovering from a site interruption, using off-site recovery.

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 \(CSQUTIL\)](#) 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.

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.

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.

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):

$$\text{Backup frequency (in hours)} = \frac{\text{Required restart time (in secs)} * \text{System recovery log read rate (in MB/sec)}}{\text{Application log write rate (in MB/hour)}}$$

Formula (A)

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:

$$100 * (1 + 1.3) \text{ KB} * 3600 = \text{approximately } 800 \text{ MB}$$

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:

$$3600 \text{ seconds} * 2.7 \text{ MB a second} / 800 \text{ MB an hour} = 12.15 \text{ hours}$$

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.

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.

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.

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 201](#) 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.

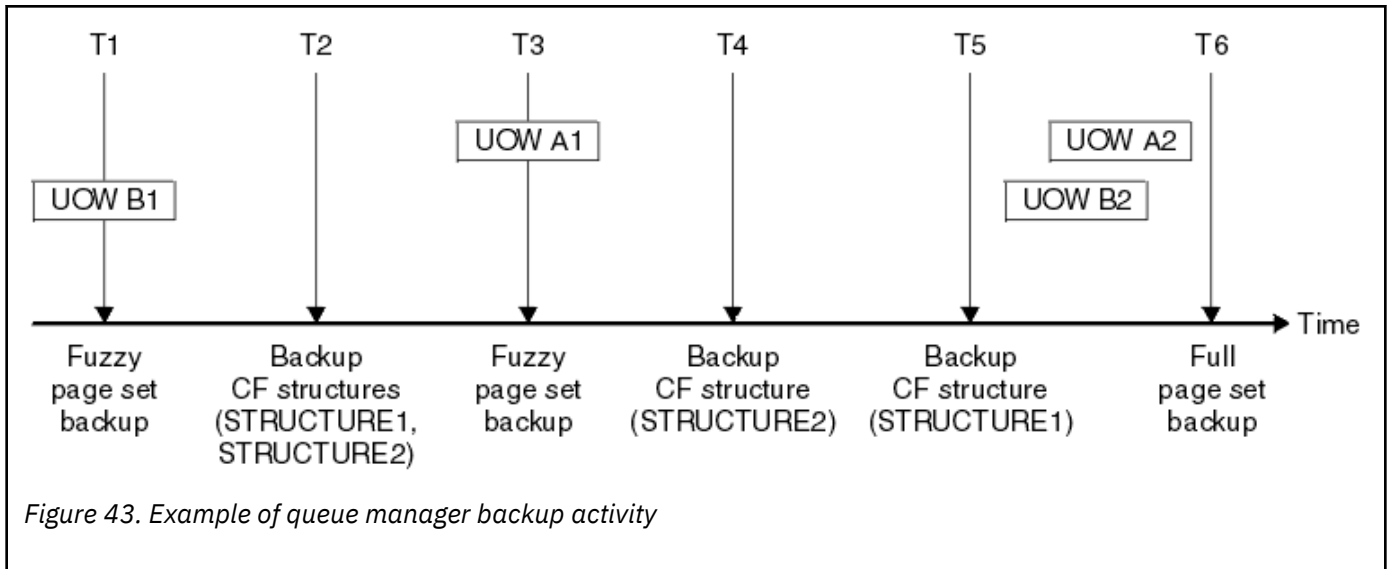


Figure 43. Example of queue manager backup activity

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. The 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 (CSQOUTIL) 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 147](#)

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](#).

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.

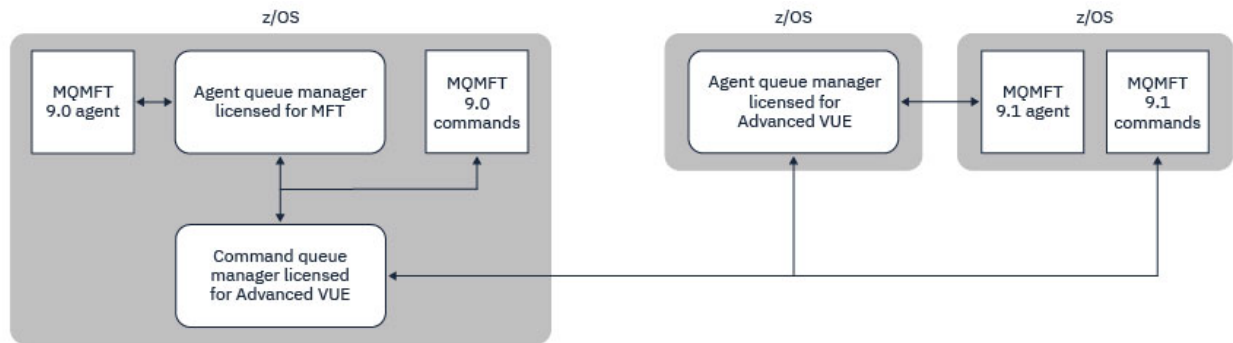


Figure 46. Commands connect to the command queue manager for an MFT topology. When running these commands on z/OS, the command queue manager must also be on z/OS

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.

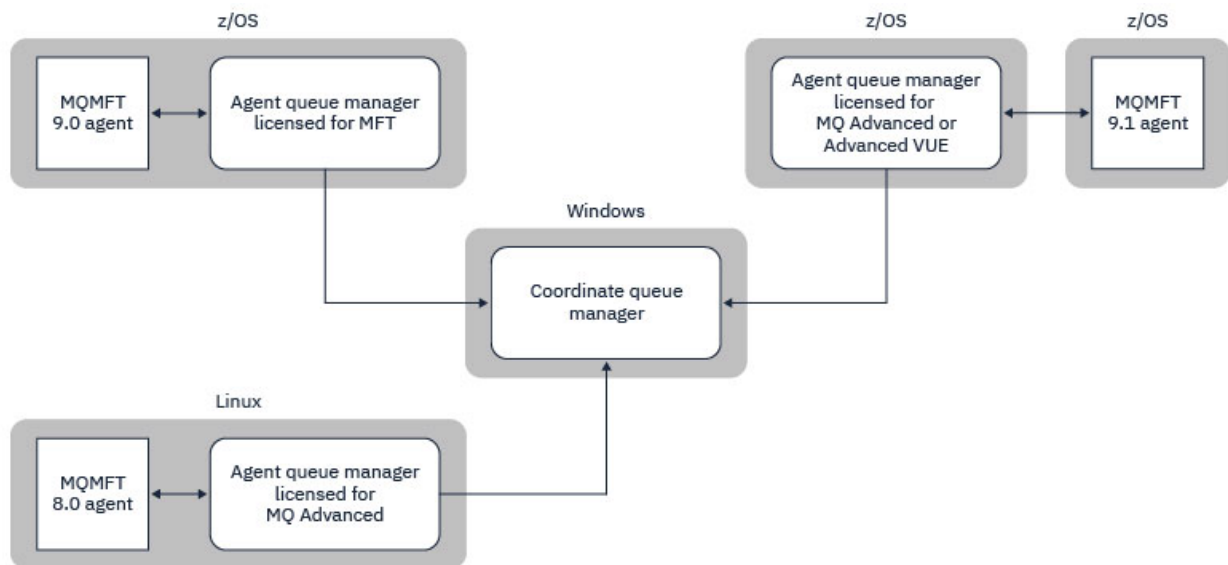


Figure 47. MFT agents running on z/OS can be part of an MFT topology where the coordination queue manager is running on an IBM MQ multiplatform.

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.

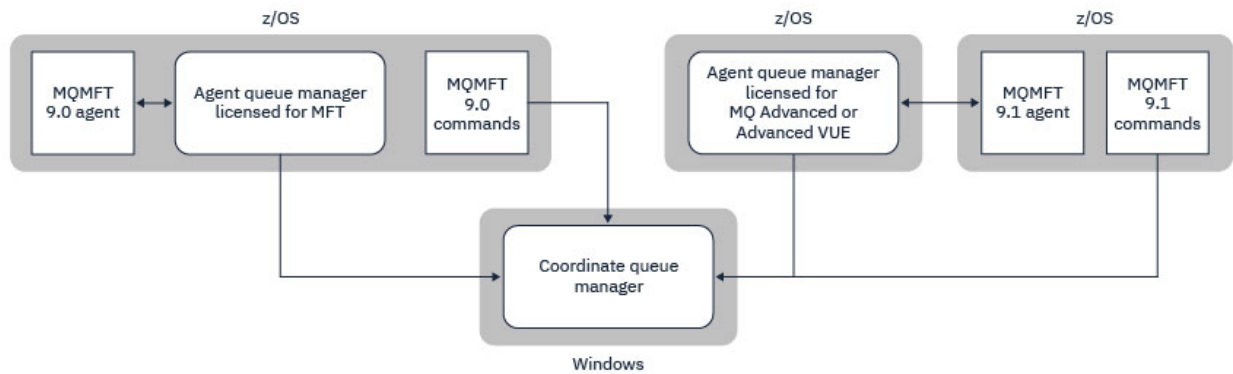


Figure 48. Certain commands, such as **fteListAgents**, connect directly to the coordination queue manager for an MFT topology.

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

<i>Table 26. MQADMIN resource class (continued)</i>	
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

<i>Table 27. MQQUEUE resource class</i>	
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:

<i>Table 28. Example user ID together with access to specific resources</i>			
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 MQWebUser1 role. For more information about the available roles for the IBM MQ Console and REST API, and the access that these roles grant, see [Roles on the IBM MQ Console and REST API](#).

For more information about configuring security for the IBM MQ Console and REST API, see [IBM MQ Console and REST API security](#).

Avisos

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos Estados Unidos.

É possível que a IBM não ofereça os produtos, serviços ou recursos discutidos nesta publicação em outros países. Consulte seu representante local do IBM para obter informações sobre produtos e serviços disponíveis atualmente em sua área. Qualquer referência a produtos, programas ou serviços IBM não significa que apenas produtos, programas ou serviços IBM possam ser utilizados. Qualquer outro produto, programa ou serviço, funcionalmente equivalente, poderá ser utilizado em substituição daqueles, desde que não infrinja nenhum direito de propriedade intelectual da IBM. Entretanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço não IBM são de responsabilidade do Cliente.

A IBM pode ter patentes ou aplicativos de patentes pendentes relativas aos assuntos tratados nesta publicação. O fornecimento desta publicação não garante ao Cliente nenhum sobre tais patentes. É possível enviar pedidos de licença, por escrito, para:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138-146
Botafogo
Rio de Janeiro, RJ
U.S.A.

Para pedidos de licença relacionados a informações de DBCS (Conjunto de Caracteres de Byte Duplo), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos de licença, por escrito, para:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

O parágrafo a seguir não se aplica a nenhum país em que tais disposições não estejam de acordo com a legislação local: A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "NO ESTADO EM QUE SE ENCONTRA", SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS A ELAS NÃO SE LIMITANDO, AS GARANTIAS IMPLÍCITAS DE NÃO INFRAÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO. Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, essa disposição pode não se aplicar ao Cliente.

Essas informações podem conter imprecisões técnicas ou erros tipográficos. São feitas alterações periódicas nas informações aqui contidas; tais alterações serão incorporadas em futuras edições desta publicação. IBM pode aperfeiçoar e/ou alterar no produto(s) e/ou programa(s) descritos nesta publicação a qualquer momento sem aviso prévio.

Todas as referências nessas informações a websites não IBM são fornecidas somente por conveniência e de forma alguma são um endosso a esses websites. Os materiais contidos nesses websites não fazem parte dos materiais desse produto IBM e a utilização desses websites é de inteira responsabilidade do Cliente.

A IBM pode utilizar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Os licenciados deste programa que desejarem obter informações sobre este assunto com o propósito de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) o uso mútuo das informações trocadas, deverão entrar em contato com:

Av. Pasteur, 138-146
Av. Pasteur, 138-146

Botafogo
Rio de Janeiro, RJ
U.S.A.

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriadas, incluindo em alguns casos o pagamento de uma taxa.

O programa licenciado descrito nesta publicação e todo o material licenciado disponível para ele são fornecidos pela IBM sob os termos do IBM Customer Agreement, IBM Contrato de Licença do Programa Internacional ou qualquer contrato equivalente entre as partes.

Todos os dados de desempenho aqui contidos foram determinados em um ambiente controlado. Portanto, os resultados obtidos em outros ambientes operacionais podem variar significativamente. Algumas medidas podem ter sido tomadas em sistemas em nível de desenvolvimento e não há garantia de que estas medidas serão iguais em sistemas geralmente disponíveis. Além disto, algumas medidas podem ter sido estimadas através de extrapolação. Os resultados reais podem variar. usuários deste documento devem verificar os dados aplicáveis para seu ambiente específico.

As informações relativas a produtos não IBM foram obtidas junto aos fornecedores dos respectivos produtos, de seus anúncios publicados ou de outras fontes disponíveis publicamente. A IBM não testou estes produtos e não pode confirmar a precisão de seu desempenho, compatibilidade nem qualquer outra reivindicação relacionada a produtos não IBM. Dúvidas sobre os recursos de produtos não IBM devem ser encaminhadas diretamente a seus fornecedores.

Todas as declarações relacionadas aos objetivos e intenções futuras da IBM estão sujeitas a alterações ou cancelamento sem aviso prévio e representam somente metas e objetivos.

Essas informações contêm exemplos de dados e relatórios utilizados em operações diárias de negócios. Para ilustrá-los da forma mais completa possível, os exemplos incluem nomes de indivíduos, empresas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com os nomes e endereços utilizados por uma empresa real é mera coincidência.

LICENÇA DE COPYRIGHT:

Estas informações contêm programas de aplicativos de amostra na linguagem fonte, ilustrando as técnicas de programação em diversas plataformas operacionais. O Cliente pode copiar, modificar e distribuir estes programas de amostra sem a necessidade de pagar à IBM, com objetivos de desenvolvimento, uso, marketing ou distribuição de programas aplicativos em conformidade com a interface de programação de aplicativo para a plataforma operacional para a qual os programas de amostra são criados. Esses exemplos não foram testados completamente em todas as condições. Portanto, a IBM não pode garantir ou implicar a confiabilidade, manutenção ou função destes programas.

Se estiver visualizando estas informações em formato eletrônico, as fotografias e ilustrações coloridas poderão não aparecer.

Informações sobre a Interface de Programação

As informações da interface de programação, se fornecidas, destinam-se a ajudá-lo a criar software aplicativo para uso com este programa.

Este manual contém informações sobre as interfaces de programação desejadas que permitem que o cliente grave programas para obter os serviços do IBM MQ

No entanto, estas informações também podem conter informações sobre diagnósticos, modificações e ajustes. As informações sobre diagnósticos, modificações e ajustes são fornecidas para ajudá-lo a depurar seu software aplicativo.

Importante: Não use essas informações de diagnóstico, modificação e ajuste como uma interface de programação, pois elas estão sujeitas a mudanças

Marcas comerciais

IBM, o logotipo IBM , ibm.com, são marcas registradas da IBM Corporation, registradas em várias jurisdições no mundo todo Uma lista atual de marcas registradas da IBM está disponível na Web em "Informações de copyright e marca registrada" www.ibm.com/legal/copytrade.shtml. Outros nomes de produtos e serviços podem ser marcas comerciais da IBM ou de outras empresas.

Microsoft e Windows são marcas registradas da Microsoft Corporation nos Estados Unidos e/ou em outros países.

UNIX é uma marca registrada do The Open Group nos Estados Unidos e em outros países.

Linux é uma marca registrada de Linus Torvalds nos Estados Unidos e/ou em outros países.

Este produto inclui software desenvolvido pelo Projeto Eclipse (<https://www.eclipse.org/>).

Java e todas as marcas registradas e logotipos baseados em Java são marcas ou marcas registradas da Oracle e/ou de suas afiliadas.



Part Number:

(1P) P/N: