9.2

规划 *IBM MQ*

IBM

注

在使用本资料及其支持的产品之前，请阅读第 177 页的『声明』中的信息。

本版本适用于 IBM® MQ V 9 发行版 2 以及所有后续发行版和修订版，直到在新版本中另有声明为止。

内容

规划 IBM MQ 体系结构

规划 IBM MQ 环境时，请考虑 IBM MQ 为单个和多个队列管理器体系结构以及点到点和发布/预订消息传递样式提供的支持。 还要规划资源需求以及日志记录和备份工具的使用。

关于此任务

在规划 IBM MQ 体系结构之前，请熟悉基本 IBM MQ 概念。 请参阅 IBM MQ 技术概述。

IBM MQ 体系结构从使用单个队列管理器的简单体系结构到相互连接的队列管理器的更复杂网络。 使用分布式排队技术将多个队列管理器连接在一起。 有关规划单个队列管理器和多个队列管理器体系结构的更多信息，请参阅以下主题:

- 第 16 页的『基于单个队列管理器的体系结构』
- 第 16 页的『基于多个队列管理器的体系结构』
  - 第 17 页的『规划分布式队列和集群』
  - 第 60 页的『规划分布式发布/预订网络』

z/OS 在 IBM MQ for z/OS 上，您可以使用共享队列和队列共享组来实现工作负载均衡，并使 IBM MQ 应用程序具有可伸缩性和高可用性。 有关共享队列和队列共享组的信息，请参阅 共享队列和队列共享组。

IBM MQ 提供了两种不同的发行版模型:

- 长期支持 (LTS) 发行版最适合需要长期部署和最大稳定性的系统
- Continuous Delivery (CD) 发行版适用于需要快速利用 IBM MQ 的最新功能增强功能的系统。

这两种发行版类型的安装方式都相同，但您需要了解与支持和迁移相关的注意事项。 请参阅 IBM MQ 发行版类型和版本控制 以获取更多信息。

有关规划多个安装，存储和性能需求以及使用客户机的信息，请参阅其他子主题。

相关概念

第 124 页的『在 z/OS 上规划 IBM MQ 环境』
规划 IBM MQ 环境时，必须考虑数据集，页集， Db2，耦合设施的资源需求以及日志记录和备份设施的需求。 使用本主题来规划运行 IBM MQ 的环境。

确保消息不丢失 (日志记录)

可用性、恢复和重新启动

相关任务

检查需求

IBM MQ 发行版类型: 规划注意事项

从 IBM MQ 9.0 开始，有两种主要发行版类型: Long Term Support (LTS) 发行版和 Continuous Delivery (CD) 发行版。 对于每个受支持的平台，您选择的发行版类型会影响订购，安装，维护和迁移。

有关发行版类型的详细信息，请参阅 IBM MQ 发行版类型和版本控制。

IBM MQ for Multiplatforms 的注意事项

Multi

顺序

在 Passport Advantage 中， IBM MQ 9.2 有两个单独的 eAssemblies 。 一个包含 IBM MQ 9.2.0 Long Term Support 发行版的安装映像，另一个包含 IBM MQ 9.2.x Continuous Delivery 发行版的安装映像。 根据您选择的发行版，从 eAssembly 下载安装映像。

所有 IBM MQ 版本以及 IBM MQ 9.2 的 LTS 发行版和 CD 发行版都属于同一产品标识。

使用 IBM MQ 的权利在整个产品 (PID) 中扩展，受许可组件和定价指标的约束。 这意味着您可以在 IBM MQ 9.2 的 LTS 发行版和 CD 发行版安装映像之间自由选择。

安装

从 Passport Advantage 下载安装映像后，应仅为安装选择已购买权利的组件。 请参阅 IBM MQ 许可证信息，以获取有关针对每个可收费组件包含哪些可安装组件的更多信息。

可以在同一操作系统映像上安装 IBM MQ 9.2.0 LTS 发行版和 IBM MQ 9.2.x CD 发行版。 如果执行此操作，那么这些组件将显示为 IBM MQ 多版本支持所支持的单独安装。 每个版本都有与该版本关联的不同队列管理器集。

每个新的 CD 发行版都作为安装映像提供。 新的 CD 发行版可以与现有发行版一起安装，也可以由安装程序将较早的 CD 发行版更新为新发行版。

CD 发行版包含功能增强以及最新的一组缺陷修订和安全性更新。 每个 CD 发行版都是累积发行版，并且完全替换该版本的 IBM MQ 的所有先前发行版。 因此，如果特定 CD 发行版不包含任何与企业相关的功能，那么可以跳过该发行版。

维护

LTS 发行版由修订包 (提供缺陷修订)和累积安全性更新 (CSU) (提供安全补丁)的应用程序提供服务。 修订包和 CSU 定期可用，并且是累积的。

对于 CD，仅针对最新的 CD 发行版 (可能在后续版本上) 生成 CSU。

有时， IBM 支持团队会指示您应用临时修订。 临时修订也称为紧急修订或测试修订，用于应用无法等待下一次维护交付的紧急更新。

LTS 发行版与 CD 发行版之间的迁移

存在一些约束和限制，但通常可以将单个队列管理器从使用 LTS 发行版代码迁移到 CD 发行版代码，或者从使用 CD 发行版代码迁移到 LTS 发行版代码，前提是目标发行版高于迁移之前使用的发行版。

可以采用两种方法:

- 安装新的代码发行版，以便更新 IBM MQ 的现有安装。 与安装关联的任何队列管理器都将在启动时使用新的代码发行版。
- 将新的代码发行版作为新安装安装，然后使用 setmqm 命令将各个队列管理器实例移至新安装。

当队列管理器开始运行代码的 CD 发行版时，将更新队列管理器命令级别以指示新的发行版级别。 这意味着将启用发行版中提供的任何新功能，并且您无法再使用具有较低 VRM 数字的代码发行版来重新启动队列管理器。

IBM MQ for z/OS 的注意事项

z/OS

顺序

订购 IBM MQ for z/OS 9.2 时， ShopZ 上提供了两个单独的功能部件。 这些功能部件对应于 LTS 发行版和 CD 发行版。 这两个功能部件都适用于相同的产品标识 (PID)。 这是已获得许可的产品标识，因此，如果一个功能部件已获得许可，那么将有权使用备用功能部件 (如果需要)。 订购时，选择与 LTS 发行版或 CD 发行版对应的功能部件。

如果要选择要包含在 ServerPac 中的产品，那么不能按同一 ServerPac 顺序同时选择 LTS 发行版和 CD 发行版，因为 SMP/E 无法将这些产品安装在同一目标区域中。

安装

LTS 和 CD 发行版在单独的 FMID 集合中提供。 请注意，这些 FMID 不能安装在同一个 SMP/E 目标区域中。如果需要 LTS 和 CD 发行版:

- 在单独的目标区域中安装 LTS 发行版和 CD 发行版
- 为两个发行版维护单独的目标库和分发库

如果队列管理器位于队列共享组中，那么在升级到最新 CD 版本时，必须升级该组中的所有队列管理器。

队列管理器的命令级别是三位数的 VRM 级别。 IBM MQ 程序可以通过传递 MQIA_COMMAND_LEVEL 选择器来调用 MQINQ，以获取它所连接到的队列管理器的命令级别。

由于发行版使用不同的 FMID，因此您无法通过维护 LTS 发行版或其他方法来更新 CD 发行版。 同样，无法将产品代码版本从 LTS 发行版切换到 CD 发行版或其他方式。 但是，您可以在发布模型之间切换队列管理器。 请参阅 LTS 发行版与 CD 发行版之间的迁移。

注:

IBM MQ 9.0.x 和 IBM MQ 9.1.x CD 发行版具有独立的版本和发行版从属 FMID。 因此，从 9.0.x CD 移动到 9.1.x CD 需要至少一个完整的 SMP/E 安装。

从 IBM MQ for z/OS 9.2.0 开始， CD 发行版使用一组 FMID，对于版本号为 9 的所有 IBM MQ for z/OS 发行版，这些 FMID 保持不变。 由于每个新版本的 IBM MQ 都作为 CD 和 LTS 发行版提供，因此您可以通过将 PTF 应用于单个 SMP/E 安装来升级 CD 发行版，即使在跨越主要版本边界时也是如此。 例如，您可以从 IBM MQ for z/OS 9.2.0 CD，到 IBM MQ for z/OS 9.2.2 CD，到 IBM MQ for z/OS 9.2.4 CD，到 IBM MQ for z/OS 9.3.0 CD，只需应用 PTF 即可。

您可以通过查看队列管理器作业日志中的 CSQY000I 消息来区分具有相同 VRM 级别的 LTS 和 CD 发行版。

维护

IBM MQ for z/OS 使用 PTF 进行维护。

LTS PTF 特定于对应于特定发行版级别的一组特定库。 对于 UNIX 系统服务功能部件 (即， JMS 和 WEB UI，连接器包和 Managed File Transfer)， z/OS PTF 直接与多平台修订包 和累积安全性更新 (CSU)保持一致。 这些修订是累积的，并且与等效的 Multiplatforms 修订包 或 CSU 同时可用。

CD CD CSU 通常在 CD 发行版之间不可用，但包含在下一个 IBM MQ for z/OS CD 发行版中。 您还可以联系支持人员以请求 + + USERMOD。

IBM MQ for z/OS 上的其他修订是特定部件上的不同修订。 这些修订可解决非累积的特定问题，并在产生这些修订时提供这些修订。

LTS 发行版与 CD 发行版之间的迁移

存在一些约束和限制，但通常可以将单个队列管理器从使用 LTS 发行版代码迁移到 CD 发行版代码，或者从使用 CD 发行版代码迁移到 LTS 发行版代码，前提是目标发行版高于迁移之前使用的发行版。

V 9.2.0 V 9.2.0 从 IBM MQ for z/OS 9.2.0 开始，您可以根据需要在具有相同 VRM 的 CD 和 LTS 发行版之间来回迁移多次，而不会影响向后迁移的能力。 例如，可以在 IBM MQ for z/OS 9.2.0 LTS 上启动队列管理器，在 IBM MQ for z/OS 9.2.0 CD 上关闭并启动，在 IBM MQ for z/OS 9.2.0 LTS 上关闭并启动队列管理器。

IBM MQ for z/OS 传统上提供了回退功能 (向后迁移)，以便在迁移后运行一段时间后，可以回退到先前发行版。

V 9.2.0 V 9.2.0 对于 LTS 发行版以及修饰符为 0 (例如 9.2.0 CD) 的 CD 发行版，将保留此功能，但当迁移的源或目标是具有非零修饰符号 (例如 9.1.5 或 9.2.1) 的 CD 发行版时，无法保留此功能。

以下是有效的迁移方案，并说明此原则如何工作:

源发行版	目标发行版	注意
8.0.0 LTS	9.2.0 LTS 或 9.2.0 CD	不支持向后迁移，因为 8.0.0 LTS 超出标准支持范围。
9.0.0 LTS	9.2.0 LTS 或 9.2.0 CD	支持向后迁移。
9.1.0 LTS	9.2.0 LTS 或 9.2.0 CD	支持向后迁移。
9.1.5 CD	9.2.0 LTS 或 9.2.0 CD	不支持向后迁移，因为源发行版是具有非零修饰符的 CD。

源发行版	目标发行版	注意
V 9.2.0 9.2.0 LTS 或 V 9.2.0 9.2.0 CD	9.2.1 CD	不支持向后迁移，因为目标发行版是具有非零修饰符的 CD。 发出 `Write to operator with reply` CSQY041D 以确认迁移。

相关任务
在 z/OS 上应用和除去维护
相关信息
下载 IBM MQ 9.2

针对 GDPR 就绪状态的 IBM MQ 和 IBM MQ Appliance 本地注意事项

适用于下列 PID：

- 5724-H72 IBM MQ
- 5655-AV9 IBM MQ Advanced for z/OS
- 5655-AV1 IBM MQ Advanced for z/OS Value Unit Edition
- 5655-AM9 IBM MQ Advanced Message Security for z/OS
- 5725-Z09 IBM MQ Appliance M2001
- 5737-H47 IBM MQ Appliance M2002
- 5655-MQ9 IBM MQ for z/OS
- 5655-VU9 IBM MQ for z/OS Value Unit Edition
- 5655-MF9 IBM MQ Managed File Transfer for z/OS
- 5655-ADV IBM WebSphere MQ Advanced for z/OS
- 5655-AMS IBM WebSphere MQ Advanced Message Security for z/OS
- 5724-A39 IBM WebSphere MQ for HP NonStop Server
- 5655-W97 IBM WebSphere MQ for z/OS
- 5655-VU8 IBM WebSphere MQ for z/OS Value Unit Edition
- 5655-VUE IBM WebSphere MQ for z/OS Value Unit Edition
- 5655-MFT IBM WebSphere MQ Managed File Transfer for z/OS

声明：

此文档旨在帮助您做好 GDPR 准备工作。 它提供的信息涉及您可以配置的 IBM MQ 功能、该产品的各方面使用，以及为了帮助组织针对 GDPR 做好准备而应该考虑的事情。 由于客户选择和配置功能部件的方式有很多种，并且产品单独使用以及与第三方应用程序和系统配合使用的方式也多种多样，因此这些信息并不是一份详尽的列表。

客户负责确保自己遵守各种法律法规，包括欧盟一般数据保护条例。 客户须自行负责从合格的法律顾问那里，就可能会影响客户业务和客户为了遵守此类法律和法规需要采取的任何行动，获得关于任何相关法律和法规的认定和解释的意见。

本文描述的产品、服务和其他功能不适用于所有客户情况，可能具有受限可用性。 IBM 不提供法律，会计或审计建议，也不表示或保证其服务或产品将确保客户遵守任何法律或法规。

目录

2. GDPR 的产品配置
3. 数据生命周期
4. 数据采集
5. 数据存储
6. 数据访问
7. 数据处理
8. 数据删除
9. 数据监视
10. 限制使用个人数据的功能
11. 文件处理

欧洲通用数据保护条例

欧盟（“EU”）已采用了《一般数据保护条例》(GDPR) 并于 2018 年 5 月 25 日起实施。

为什么 GDPR 很重要?

GDPR 建立了用于处理个人数据的更强大的数据保护监管框架。 GDPR 实现:

- 新增及增强的个人权利
- 更广泛的个人数据定义
- 新增的处理商义务
- 可能会因不合规而遭受重大经济处罚
- 强制性数据违规通知

请阅读关于 GDPR 的更多信息:

- 欧盟 GDPR 信息门户网站
- ibm.com/GDPR Web 站点

产品配置 - 为 GDPR 做准备时的注意事项

以下部分提供配置 IBM MQ 以帮助贵组织准备 GDPR 就绪的注意事项。

数据生命周期

IBM MQ 是面向事务性消息的中间件产品，使应用程序能够异步交换应用程序提供的数据。 IBM MQ 支持一系列用于连接应用程序的消息传递 API，协议和网桥。 因此，可以使用 IBM MQ 来交换多种形式的数据，其中一些数据可能受 GDPR 的约束。 IBM MQ 可能会与多种第三方产品交换数据。 其中一些是 IBM 拥有的，但其他许多是由其他技术供应商提供的。 软件产品兼容性报告 Web 站点 提供关联软件的列表。 有关第三方产品的 GDPR 就绪性的注意事项，请参阅该产品的文档。 IBM MQ 管理员通过队列，主题和预订的定义来控制 IBM MQ 与通过它传递的数据进行交互的方式。

哪些类型的数据流经 IBM MQ?

由于 IBM MQ 为应用程序数据提供异步消息传递服务，因此此问题没有一个确定的答案，因为用例因应用程序部署而异。 应用程序消息数据持久存储在队列文件 (页集或 z/OS 上的耦合设施)，日志和归档中，并且消息本身可能包含由 GDPR 管理的数据。 应用程序提供的消息数据也可能包含在为问题确定目的而收集的文件中，例如错误日志，跟踪文件和 FFST。 在 z/OS 应用程序上，提供的消息数据也可能包含在地址空间或耦合设施转储中。

以下是可使用 IBM MQ 交换的个人数据的一些典型示例:

- 客户的员工 (例如， IBM MQ 可能用于连接客户的工资单或 HR 系统)
- 客户自己的客户个人数据 (例如， IBM MQ 可能由客户用于在与其客户相关的应用程序之间交换数据，例如，获取销售线索并将数据存储在其 CRM 系统中)。

- 客户自己的客户的敏感个人数据 (例如， IBM MQ 可能在需要交换个人数据的行业环境中使用，例如，集成临床应用程序时基于 HL7-based 医疗保健记录)。

除了应用程序提供的消息数据外， IBM MQ 还会处理以下类型的数据:

- 认证凭证 (例如用户名和密码， API 密钥等)
- 技术上可识别的个人信息（例如，设备标识、基于使用情况的标识、IP 地址等 - 当链接到个人时）

用于与 IBM 在线联系的个人数据

IBM MQ 客户可以通过多种方式提交在线评论/反馈/请求，以联系 IBM IBM MQ 主题，主要包括:

- IBM Developer 上的 IBM MQ 区域 中的页面上的公共评论区域
- IBM Documentation 中的 IBM MQ 产品信息 页面上的公共评论区域
- IBM 支持论坛 中的公共评论
- IBM Integration Ideas 中的公共评论

通常，只有客户姓名和电子邮件地址用来启用联系主题的个人回复，而且个人数据的使用符合 IBM 在线隐私声明。

数据 收集

IBM MQ 可用于收集个人数据。 在评估您对 IBM MQ 的使用以及满足 GDPR 需求的需求时，应考虑在您的环境中传递 IBM MQ 的个人数据类型。 您可能想要考虑如下几个方面:

- 数据如何到达队列管理器? (跨哪些协议? 数据是否已加密？ 数据是否已签名？）
- 如何从队列管理器发送数据? (跨哪些协议? 数据是否已加密？ 数据是否已签名？）
- 数据在通过队列管理器时如何存储? (任何消息传递应用程序都有可能将消息数据写入有状态介质，即使消息是非持久的。 您是否知道消息传递功能可能如何公开通过产品传递的应用程序消息数据的各个方面?)
- 在 IBM MQ 需要访问第三方应用程序时，如何收集和存储凭证?

IBM MQ 可能需要与需要认证的其他系统和服务 (例如 LDAP) 进行通信。 需要时，认证数据 (用户标识和密码) 由 IBM MQ 配置和存储，以便在此类通信中使用。 尽可能避免使用个人凭证进行 IBM MQ 认证。 请考虑保护用于认证数据的存储器。 (请参阅下面的 "数据存储"。)

数据存储

当消息数据通过队列管理器传输时， IBM MQ 会将该数据直接持久存储 (可能有多个副本) 到有状态介质。 IBM MQ 用户可能希望考虑在消息数据处于静态状态时对其进行保护。

以下项目突出显示了 IBM MQ 持久存储应用程序提供的数据的区域，用户在确保符合 GDPR 时可能需要考虑这些区域。

- 应用程序消息队列:

  IBM MQ 提供消息队列以允许应用程序之间进行异步数据交换。 存储在队列上的非持久和持久消息将写入有状态介质。
- 文件传输代理队列:

  IBM MQ Managed File Transfer 利用消息队列来协调文件数据的可靠传输，包含个人数据和传输记录的文件存储在这些队列上。
- 传输队列:

  为了在队列管理器之间可靠地传输消息，将消息临时存储在传输队列上。
- 死信队列:

  在某些情况下，如果在队列管理器上配置了死信队列，那么无法将消息放入目标队列并将其存储在死信队列上。
- 回退队列:

JMS 和 XMS 消息传递接口提供了一种功能，允许在发生多次回退后将有害消息移至回退队列，以允许处理其他有效消息。

- AMS 错误队列:

  IBM MQ Advanced Message Security 会将不符合安全策略的消息移动到 SYSTEM.PROTECTION.ERROR.QUEUE 错误队列与死信队列相似。

- 保留出版物:

  IBM MQ 提供了保留发布功能，以允许预订应用程序重新调用先前的发布。

- 延迟交付:

  IBM MQ 支持 JMS 2.0 传递延迟功能，该功能允许将来将消息传递到其目标。 尚未传递的消息存储在 SYSTEM.DDELAY.LOCAL.QUEUE 队列。

阅读更多:

- 日志记录: 确保消息不会丢失
- MFT 代理队列设置
- 使用死信队列
- 在 IBM MQ classes for JMS 中处理有害消息
- AMS 错误处理
- 保留的发布内容
- JMS 2.0 发送延迟

以下项目突出显示了 IBM MQ 可能间接持久存储应用程序提供的数据的区域，用户在确保遵守 GDPR 时也可能希望考虑这些数据。

- 跟踪路由消息传递:

  IBM MQ 提供跟踪路由功能，用于记录消息在应用程序之间采用的路由。 生成的事件消息可能包括技术上可识别的个人信息，例如 IP 地址。

- 应用程序活动跟踪:

  IBM MQ 提供了应用程序活动跟踪，用于记录应用程序和通道的消息传递 API 活动，应用程序活动跟踪可以将应用程序提供的消息数据的内容记录到事件消息中。

- 服务跟踪:

  IBM MQ 提供了服务跟踪功能，用于记录消息数据流所通过的内部代码路径。 作为这些功能的一部分，IBM MQ 可以记录应用程序提供的消息数据的内容，以跟踪存储在磁盘上的文件。

- 队列管理器事件:

  IBM MQ 可以生成可包含个人数据的事件消息，例如权限，命令和配置事件。

阅读更多:

- 跟踪路由消息传递
- 使用跟踪
- 事件监视
- 队列管理器事件

要保护对应用程序提供的消息数据副本的访问权，请考虑以下操作:

- 限制特权用户对文件系统中 IBM MQ 数据的访问权，例如限制 UNIX and Linux® 平台上 "mqm" 组的用户成员资格。
- 通过专用队列和访问控制限制应用程序对 IBM MQ 数据的访问。 在适当情况下，避免不必要的资源 (例如应用程序之间的队列) 共享，并提供对队列和主题资源的细粒度访问控制。
- 限制对高可用性 (HA) 或灾难恢复 (DR) 配置中 IBM MQ 数据的复制副本的访问，并保护用于复制的连接。
- 使用 IBM MQ Advanced Message Security 提供消息数据的端到端签名和/或加密。
- 使用文件或卷级别加密来保护用于存储跟踪日志的目录的内容。

- 将服务跟踪上载到 IBM 后，如果您关注可能包含个人数据的内容，那么可以删除服务跟踪文件和 FFST 数据。

阅读更多:

- 特权用户
- 在 Multiplatforms 版上规划文件系统支持

IBM MQ 管理员可以使用凭证 (用户名和密码， API 密钥等) 配置队列管理器 用于 3rd 服务，例如 LDAP , Salesforce 等。 此数据通常存储在通过文件系统许可权保护的队列管理器数据目录中。

创建 IBM MQ 队列管理器时，将使用基于组的访问控制来设置数据目录，以便 IBM MQ 可以读取配置文件并使用凭证连接到这些系统。 IBM MQ 管理员被视为特权用户，并且是此组的成员，因此具有对文件的读访问权。 某些文件已加密，但未加密。 因此，要完全保护对凭证的访问权，您应该考虑以下操作:

- 限制特权用户对 IBM MQ 数据的访问权，例如限制 UNIX and Linux 平台上 "mqm" 组的成员资格。
- 使用文件或卷级别加密来保护队列管理器数据目录的内容。
- 对生产配置目录的备份进行加密，并使用相应的访问控制来存储这些备份。
- 考虑通过安全性，命令和配置事件为认证失败，访问控制和配置更改提供审计跟踪。

阅读更多:

- 保护 IBM MQ

数据访问

可以通过以下产品接口访问 IBM MQ 队列管理器数据，其中一些设计用于通过远程连接进行访问，其他设计用于通过本地连接进行访问。

- IBM MQ 控制台 [仅远程]
- IBM MQ 管理 REST API [仅远程]
- IBM MQ 消息传递 REST API [仅远程]
- MQI [本地和远程]
- JMS [本地和远程]
- XMS [本地和远程]
- IBM MQ 遥测 (MQTT) [仅远程]
- IBM MQ Light (AMQP) [仅远程]
- IBM MQ IMS 网桥 [仅本地]
- IBM MQ CICS 网桥 [仅本地]
- IBM MQ MFT 协议网桥 [仅远程]
- IBM MQ Connect:Direct 网桥 [仅远程]
- IBM MQ 网桥到 Salesforce [仅远程]
- IBM MQ 网桥到 Blockchain [仅远程]
- IBM MQ MQAI [本地和远程]
- IBM MQ PCF 命令 [本地和远程]
- IBM MQ MQSC 命令 [本地和远程]
- IBM MQ Explorer [本地和远程]
- IBM MQ 用户出口 [仅本地]
- IBM MQ Internet Pass-Thru [仅远程]
- Red Hat® OpenShift® 监视 (Prometheus) 度量 (这些度量是有关队列管理器统计信息的数字数据)
- IBM Cloud Pak for Integration Operations Dashboard 集成，用于将高级跟踪数据提交到中央源 (仅限 CP4I)
- IBM MQ Appliance 串行控制台 [仅本地]

- IBM MQ Appliance SSH [仅远程]
- IBM MQ Appliance REST API [仅远程]
- IBM MQ Appliance Web UI [仅远程]

这些接口旨在允许用户对 IBM MQ 队列管理器以及存储在其中的消息进行更改。 管理和消息传递操作是安全的，因此在发出请求时涉及三个阶段;

- 认证
- 角色映射
- 授权

认证:

如果从本地连接请求了消息或管理操作，那么此连接的源是同一系统上正在运行的进程。 运行该进程的用户必须已通过操作系统提供的任何认证步骤。 将从中建立连接的进程的所有者的用户名声明为身份。 例如，这可能是运行已从中启动应用程序的 shell 的用户的名称。 本地连接的可能认证形式为:

1. 已声明的用户名 (本地操作系统)
2. 可选用户名和密码 (操作系统， LDAP 或定制 3rd 存储库)

如果从远程连接请求了管理操作，那么将通过网络接口与 IBM MQ 进行通信。 以下形式的身份可以通过网络连接进行认证;

1. 已声明的用户名 (来自远程操作系统)
2. 用户名和密码 (操作系统， LDAP 或定制 3rd 存储库)
3. 源网络地址 (例如 IP 地址)
4. X.509 数字证书 (相互 SSL/TLS 认证)
5. 安全性令牌 (例如 LTPA2 令牌)
6. 其他定制安全性 (3rd 参与方出口提供的功能)
7. SSH 密钥

IBM MQ 与 IBM Cloud Pak for Integration 的集成为 Web 控制台添加了新的认证类型: 使用 Cloud Pak 进行单点登录。 (仅限 CP4I)

角色映射:

在角色映射阶段，可以将认证阶段中提供的凭证映射到备用用户标识。 如果允许映射的用户标识继续 (例如，管理用户可能被通道认证规则阻止) ，那么在针对 IBM MQ 资源授权活动时，映射的用户标识将转入最终阶段。

授权:

IBM MQ 使不同用户能够对不同的消息传递资源 (例如队列， 主题和其他队列管理器对象) 具有不同的权限。

日志记录活动:

IBM MQ 的某些用户可能需要创建访问 MQ 资源的审计记录。 理想审计日志的示例可能包括配置更改，其中包含有关更改及其请求者的信息。

以下信息来源可用于实现此要求:

1. 可以将 IBM MQ 队列管理器配置为在成功运行管理命令时生成命令事件。
2. 可以将 IBM MQ 队列管理器配置为在创建，变更或删除队列管理器资源时生成配置事件。
3. 可以将 IBM MQ 队列管理器配置为在资源的授权检查失败时生成权限事件。
4. 指示授权检查失败的错误消息将写入队列管理器错误日志。
5. 当认证，授权检查失败或者创建，启动，停止或删除队列管理器时， IBM MQ 控制台会将审计消息写入其日志。
6. IBM MQ Appliance 会将审计消息写入其日志以记录用户登录和系统更改。

在考虑这些类型的解决方案时， IBM MQ 用户可能希望考虑以下几点:

- 事件消息是非持久的，因此当队列管理器重新启动时，信息将丢失。 应将任何事件监视器配置为持续使用任何可用消息并将内容传输到持久介质。
- IBM MQ 特权用户具有足够的特权来禁用事件，清除日志或删除队列管理器。

有关保护对 IBM MQ 数据的访问并提供审计跟踪的更多信息，请参阅以下主题:

- IBM MQ 安全性机制
- 配置事件
- 命令事件
- 错误日志

数据处理

使用公共密钥基础结构进行加密:

您可以通过指定连接使用 TLS 来保护与 IBM MQ 的网络连接，这还可以提供连接起始端的相互认证。

使用传输机制所提供的 PKI 安全工具是保护 IBM MQ 数据处理的第一步。 但是，在不启用进一步的安全功能的情况下，消费应用程序的行为是处理传递给它的所有消息，而不验证消息的来源或在传输过程中是否发生了更改。

获得许可使用 Advanced Message Security (AMS) 功能的 IBM MQ 用户可以通过定义和配置安全策略来控制应用程序处理消息中保存的个人数据的方式。 安全策略允许将数字签名和/或加密应用于应用程序之间的消息数据。

在使用消息时，可以使用安全策略来要求和验证数字签名，以确保消息是真实的。 AMS 加密提供了一种方法，通过该方法将消息数据从可读格式转换为编码版本，只有当它是预期的接收方或消息并且有权访问正确的解密密钥时，才能由另一个应用程序进行解码。

有关使用 SSL 和证书来保护网络连接的更多信息，请参阅 IBM MQ 产品文档中的以下主题:

- 为 IBM MQ
- AMS 概述

数据删除

IBM MQ 提供了用于删除已提供给产品的数据的命令和用户界面操作。 这意味着如果需要， IBM MQ 的用户可以删除与特定个人相关的数据。

- 要考虑遵守 GDPR 客户机数据删除的 IBM MQ 行为区域
  - 通过以下方法删除存储在应用程序队列上的消息数据:
    - 使用消息传递 API 或工具或使用消息到期来除去个别消息。
    - 指定消息是非持久消息，保留在非持久消息类正常的队列上，并重新启动队列管理器。
    - 以管理方式清除队列。
    - 正在删除队列。
  - 通过以下方法删除存储在主题上的保留发布数据:
    - 指定消息是非持久消息，并重新启动队列管理器。
    - 将保留的数据替换为新数据或使用消息到期。
    - 以管理方式清除主题字符串。
  - 通过删除整个队列管理器以及用于高可用性或灾难恢复的任何复制副本，删除存储在队列管理器上的数据。
  - 通过删除跟踪目录中的文件来删除服务跟踪命令所存储的数据。
  - 删除通过删除 errors 目录中的文件存储的 FFST 数据。
  - 删除地址空间和耦合设施转储 (在 z/OS 上)。
  - 删除此类数据的归档，备份或其他副本。

- 要考虑遵守 GDPR 帐户数据删除的 IBM MQ 行为区域
  - 您可以通过删除 (包括归档，备份或其他复制副本) 来删除 IBM MQ 存储的用于连接到队列管理器和 3rd 服务的帐户数据和首选项:
    - 用于存储凭证的队列管理器认证信息对象。
    - 队列管理器权限记录引用用户标识。
    - 用于映射或阻止特定 IP 地址，证书 DN 或用户标识的队列管理器通道认证规则。
    - 由 IBM MQ Managed File Transfer 代理程序，记录器和 MQ Explorer MFT 插件用于向队列管理器和文件服务器进行认证的凭证文件。
    - X.509 数字证书，用于表示或包含有关可由 SSL/TLS 连接或 IBM MQ Advanced Message Security (AMS) 使用的密钥库中个人的信息。
    - 来自 IBM MQ Appliance 的个人用户帐户，包括系统日志文件中对这些帐户的引用。
    - IBM MQ Explorer 工作空间元数据和 Eclipse 设置。
    - IBM MQ Explorer 密码存储，如 密码首选项中所指定。
    - IBM MQ 控制台和 mqweb 服务器配置文件。
    - Salesforce 连接数据配置文件。
    - Blockchain 连接数据配置文件。
    - IBM MQ Internet Pass-Thru 配置文件和密钥库。

阅读更多:

- 配置 IBM MQ Bridge to Salesforce
- 配置 IBM MQ 以用于区块链
- MFT 和 IBM MQ 连接认证
- 使用 ProtocolBridgeCredentials.xml 文件映射文件服务器的凭证
- 配置 IBM MQ Console 用户和角色

数据监视

IBM MQ 提供了一系列监视功能，用户可以利用这些功能来更好地了解应用程序和队列管理器的执行方式。

IBM MQ 还提供了一些帮助管理队列管理器错误日志的功能。

阅读更多:

- 监视 IBM MQ 网络
- 诊断消息服务
- QMErrorLog 服务
- IBM MQ Appliance 监视和报告

限制使用个人数据的功能

通过使用本文档中概述的工具，IBM MQ 允许最终用户限制对其个人数据的使用。

IBM MQ 消息队列不应以与数据库相同的方式用作永久数据存储器，在处理受 GDPR 约束的应用程序数据时尤其如此。

与可以通过搜索查询找到数据的数据库不同，除非您知道消息的队列，消息和相关标识，否则很难找到消息数据。

提供的包含个人数据的消息可以随时识别和定位，可以使用标准 IBM MQ 消息传递功能来访问或修改消息数据。

文件处理

1. IBM MQ Managed File Transfer 不会对传输的文件执行恶意软件扫描。 按原样传输文件，并执行完整性检查以确保在传输期间不修改文件数据。 源和目标校验和作为传输状态发布的一部分发布。 建议最终用户在 MFT 将文件传输到远程端点之前以及在 MFT 将文件传递到远程端点之后，针对其环境实施相应的恶意软件扫描。
2. IBM MQ Managed File Transfer 不会根据 MIME 类型或文件扩展名执行操作。 MFT 读取文件并传输与从输入文件读取的字节完全相同的字节。

基于单个队列管理器的体系结构

最简单的 IBM MQ 体系结构涉及单个队列管理器的配置和使用。

在规划 IBM MQ 体系结构之前，请熟悉基本 IBM MQ 概念。 请参阅 IBM MQ 技术概述。

以下部分描述了使用单个队列管理器的许多可能的体系结构:

- 第 16 页的『具有访问服务的本地应用程序的单个队列管理器』
- 第 16 页的『具有作为客户机访问服务的远程应用程序的单个队列管理器』
- 第 16 页的『具有发布/预订配置的单个队列管理器』

具有访问服务的本地应用程序的单个队列管理器

基于单个队列管理器的第一个体系结构是访问服务的应用程序与提供服务的应用程序在同一系统上运行。 IBM MQ 队列管理器提供请求服务的应用程序与提供服务的应用程序之间的异步相互通信。 这意味着即使其中一个应用程序处于脱机状态很长一段时间，应用程序之间的通信也可以继续。

具有作为客户机访问服务的远程应用程序的单个队列管理器

基于单个队列管理器的第二个体系结构具有从提供服务的应用程序远程运行的应用程序。 远程应用程序在服务的不同系统上运行。 应用程序作为客户机连接到单个队列管理器。 这意味着可以通过单个队列管理器向多个系统提供对服务的访问权。

此体系结构的一个限制是网络连接必须可供应用程序运行。 应用程序与队列管理器之间通过网络连接进行的交互是同步的。

具有发布/预订配置的单个队列管理器

使用单个队列管理器的备用体系结构是使用发布/预订配置。 在发布/预订消息传递中，可以将信息提供者与该信息的使用者分离。 这与先前描述的体系结构中的点到点消息传递样式不同，其中应用程序必须知道有关目标应用程序的信息，例如要将消息放入的队列名称。 通过使用 IBM MQ 发布/预订，发送应用程序将根据信息主题发布具有指定主题的消息。 IBM MQ 处理将消息分发到已通过预订在该主题中注册兴趣的应用程序。 接收应用程序也不需要知道要接收这些应用程序的消息源的任何信息。 有关更多信息，请参阅 发布/预订消息传递 和 单个队列管理器发布/预订配置的示例。

相关概念

IBM MQ 简介

相关任务

第 5 页的『规划 IBM MQ 体系结构』
规划 IBM MQ 环境时，请考虑 IBM MQ 为单个和多个队列管理器体系结构以及点到点和发布/预订消息传递样式提供的支持。 还要规划资源需求以及日志记录和备份工具的使用。

在 Multiplatforms 版上创建和管理队列管理器

基于多个队列管理器的体系结构

您可以使用分布式消息排队技术来创建涉及配置和使用多个队列管理器的 IBM MQ 体系结构。

在规划 IBM MQ 体系结构之前，请熟悉基本 IBM MQ 概念。 请参阅 IBM MQ 技术概述。

可以通过添加其他队列管理器来更改 IBM MQ 体系结构，而无需更改提供服务的应用程序。

应用程序可以与队列管理器托管在同一台机器上，然后与另一个系统上的另一个队列管理器上托管的服务进行异步通信。 或者，访问服务的应用程序可以作为客户机连接到队列管理器，然后在另一个队列管理器上提供对服务的异步访问。

用于连接不同队列管理器及其队列的路由是使用分布式排队技术定义的。 体系结构中的队列管理器使用通道进行连接。 通道用于根据队列管理器的配置在一个方向上自动将消息从一个队列管理器移动到另一个队列管理器。

有关规划 IBM MQ 网络的高级概述，请参阅 第 18 页的『设计分布式队列管理器网络』。

有关如何为 IBM MQ 体系结构规划通道的信息，请参阅 IBM MQ 分布式排队技术。

分布式队列管理使您能够创建和监视队列管理器之间的通信。 有关分布式队列管理的更多信息，请参阅 分布式队列管理简介。

相关任务

第 5 页的『规划 IBM MQ 体系结构』
规划 IBM MQ 环境时，请考虑 IBM MQ 为单个和多个队列管理器体系结构以及点到点和发布/预订消息传递样式提供的支持。 还要规划资源需求以及日志记录和备份工具的使用。

在 Multiplatforms 版上创建和管理队列管理器

规划分布式队列和集群

您可以手动连接分布式队列管理器上托管的队列，也可以创建队列管理器集群并让产品为您连接队列管理器。 要为分布式消息传递网络选择合适的拓扑，需要考虑您对手动控制，网络大小，更改频率，可用性和可伸缩性的需求。

开始之前

此任务假定您了解分布式消息传递网络是什么以及它们的工作方式。 有关技术概述，请参阅 分布式排队和集群。

关于此任务

要创建分布式消息传递网络，您可以手动配置通道以连接在不同队列管理器上托管的队列，也可以创建队列管理器集群。 集群使队列管理器能够相互通信，而无需设置额外的通道定义或远程队列定义，从而简化其配置和管理。

要为分布式发布/预订网络选择合适的拓扑，您需要考虑以下广泛问题:

- 您需要对网络中的连接进行多少手动控制?
- 您的网络将有多大?
- 它将具有多大的动态性?
- 您的可用性和可伸缩性需求是什么?

过程

- 请考虑您需要对网络中的连接进行多少手动控制。

  如果只需要几个连接，或者需要非常精确地定义各个连接，那么您可能应该手动创建网络。

  如果您需要多个逻辑相关的队列管理器，并且需要共享数据和应用程序，那么应考虑将它们分组到队列管理器集群中。

- 估算网络需要的大小。

  a) 估算需要多少队列管理器。 请记住，可以在多个队列管理器上托管队列。

  b) 如果您正在考虑使用集群，请添加两个额外的队列管理器以充当完整存储库。

  对于较大的网络，手动配置和维护连接可能非常耗时，您应该考虑使用集群。

- 请考虑网络活动的动态程度。

规划要在性能队列管理器上托管的繁忙队列。

如果您期望频繁创建和删除队列，请考虑使用集群。

- 考虑您的可用性和可伸缩性需求。

  a) 决定是否需要保证队列管理器的高可用性。 如果是这样，请估算此需求适用于多少队列管理器。

  b) 请考虑某些队列管理器的能力是否低于其他队列管理器。

  c) 请考虑到某些队列管理器的通信链路是否比其他队列管理器更脆弱。

  d) 请考虑在多个队列管理器上托管队列。

  手动配置的网络和集群都可以配置为高可用性和可伸缩性。 如果使用集群，那么需要将两个额外的队列管理器定义为完整存储库。 拥有两个完整存储库可确保集群在其中一个完整存储库变为不可用时继续运行。 确保完整的存储库队列管理器功能强大，性能良好，并且具有良好的网络连接。 请勿计划将完整存储库队列管理器用于任何其他工作。

- 根据这些计算，使用提供的链接来帮助您决定是手动配置队列管理器之间的连接，还是使用集群。

下一步做什么

现在，您已准备好配置分布式消息传递网络。

相关任务

配置分布式队列

配置队列管理器集群

设计分布式队列管理器网络

IBM MQ 使用队列管理器和通道在应用程序之间以及通过网络发送和接收数据。 网络规划涉及定义需求，以创建用于通过网络连接这些系统的框架。

可以在您的系统与需要进行通信的任何其他系统之间创建通道。 可以创建多中继段通道以连接到没有直接连接的系统。 方案中描述的消息通道连接在 第 19 页的图 1 中显示为网络图。

如果需要在不同物理网络上的系统之间创建通道，或者需要在通过防火墙进行通信的通道之间创建通道，那么使用 IBM MQ Internet Pass-Thru 可以简化配置。 有关更多信息，请参阅 IBM MQ Internet Pass-Thru。

通道和传输队列名称

可以为传输队列提供任何名称。 但是为了避免混淆，您可以为它们提供与目标队列管理器名称或队列管理器别名相同的名称 (视情况而定)。 这使传输队列与它们使用的路由相关联，提供了通过中间 (多跳跃) 队列管理器创建的并行路由的清晰概述。

对于通道名称而言，它并不是那么清晰。 例如， 第 19 页的图 1 中针对 QM2 的通道名称对于入局和出局通道必须不同。 所有通道名称仍可包含其传输队列名称，但必须对其进行限定以使其唯一。

例如，在 QM2 上，有一个 QM3 通道来自 QM1，还有一个 QM3 通道转至 QM3。 要使名称唯一，第一个名称可以命名为 QM3_from_QM1，第二个名称可以命名为 QM3_from_QM2。 这样，通道名称在名称的第一部分中显示传输队列名称。 方向和相邻队列管理器名称显示在名称的第二部分中。

第 19 页的表 1 中给出了 第 19 页的图 1 的建议通道名称表。

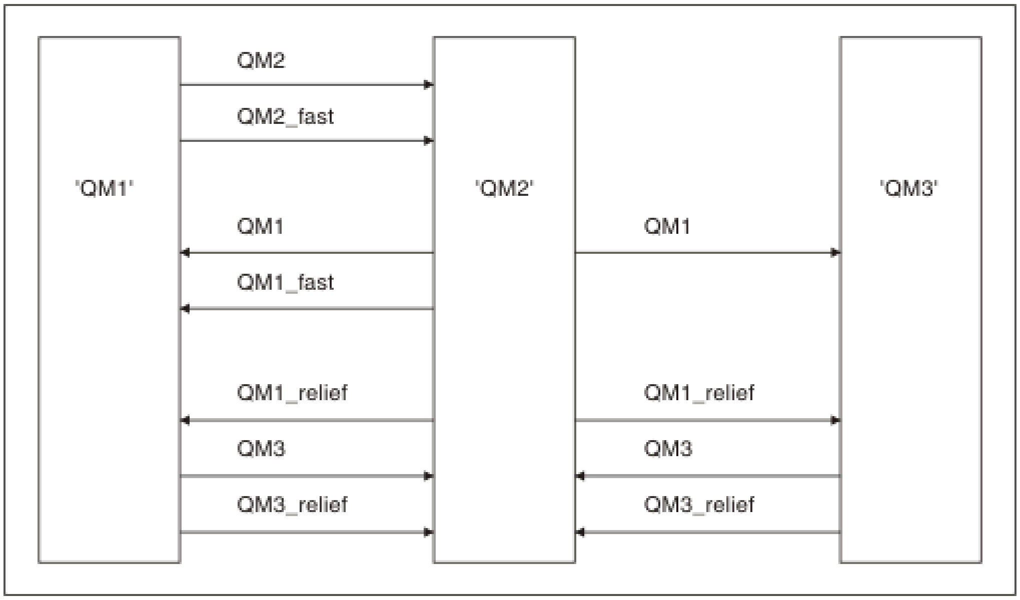


图 *1:* 显示所有通道的网络图

表 *1:* 通道名称示例

路由名称	队列管理器托管通道	传输队列的名称	建议的通道名称
QM1	QM1 & QM2	QM1 (位于 QM2)	QM1.from.QM2
QM1	QM2 & QM3	QM1 (位于 QM3)	QM1.from.QM3
QM1_fast	QM1 & QM2	QM1_fast (位于 QM2)	QM1_fast.from.QM2
QM1_relief	QM1 & QM2	QM1_relief (位于 QM2)	QM1_relief.from.QM2
QM1_relief	QM2 & QM3	QM1_relief (位于 QM3)	QM1_relief.from.QM3
QM2	QM1 & QM2	QM2 (位于 QM1)	QM2.from.QM1
QM2_fast	QM1 & QM2	QM2_fast (位于 QM1)	QM2_fast.from.QM1
QM3	QM1 & QM2	QM3 (位于 QM1)	QM3.from.QM1
QM3	QM2 & QM3	QM3 (位于 QM2)	QM3.from.QM2
QM3_relief	QM1 & QM2	QM3_relief (位于 QM1)	QM3_relief.from.QM1
QM3_relief	QM2 & QM3	QM3_relief (位于 QM2)	QM3_relief.from.QM2

注:

1. z/OS 在 IBM MQ for z/OS 上，队列管理器名称限制为 4 个字符。
2. 对网络中的所有通道进行唯一命名。 如 第 19 页的表 1 中所示，在通道名称中包含源队列管理器名称和目标队列管理器名称是一种很好的方法。

网络策划员

创建网络假定有另一个更高级别的 网络规划员 功能，其计划由团队的其他成员实施。

对于广泛使用的应用程序，使用本地访问站点之间的宽带链路进行消息流量集中的本地访问站点方面的思考更为经济，如 第 20 页的图 2 所示。

在这个例子中，有两个主要系统和一些卫星系统。 实际配置将取决于业务注意事项。 有两个集中器队列管理器位于方便的中心。 每个 QM 集中器都具有到本地队列管理器的消息通道:

- QM 集中器 1 具有到三个本地队列管理器 (QM1， QM2 和 QM3) 中每一个的消息通道。 使用这些队列管理器的应用程序可以通过 QM 集中器相互通信。
- QM 集中器 2 具有到三个本地队列管理器 (QM4， QM5 和 QM6) 的消息通道。 使用这些队列管理器的应用程序可以通过 QM 集中器相互通信。
- QM 集中器之间有消息通道，因此允许队列管理器中的任何应用程序与另一个队列管理器中的任何其他应用程序交换消息。

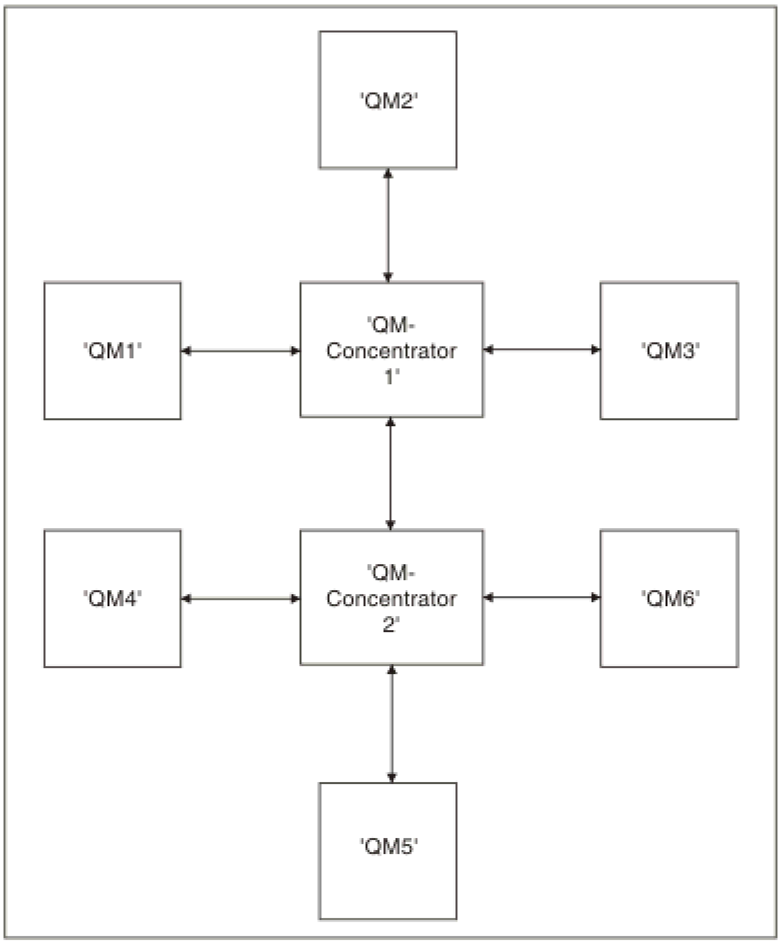


图 *2:* 显示 *QM* 集中器的网络图

设计集群

集群提供了一种用于以简化初始配置和持续管理的方式互连队列管理器的机制。 必须对集群进行精心设计，以确保它们正常运行，并确保它们达到所需的可用性和响应能力级别。

开始之前

有关集群概念的简介，请参阅以下主题:

- 分布式队列和集群
- 第 26 页的『集群与分布式排队的比较』
- 集群的组件

在设计队列管理器集群时，必须做出一些决策。 您必须首先决定集群中的哪些队列管理器将保存集群信息的完整存储库。 您创建的任何队列管理器都可以在集群中工作。 您可以为此目的选择任意数量的队列管理

器，但理想数量为两个。 有关选择队列管理器以保存完整存储库的信息，请参阅 第 28 页的『如何选择集群队列管理器以保存完整存储库』。

请参阅以下主题以获取有关设计集群的更多信息:

- 第 33 页的『示例集群』
- 第 29 页的『组织集群』
- 第 29 页的『集群命名约定』
- z/OS 第 30 页的『队列共享组和集群』
- 第 31 页的『重叠集群』

下一步做什么

有关配置和使用集群的更多信息，请参阅以下主题:

- 在集群中建立通信
- 配置队列管理器集群
- 将消息路由到集群以及从集群路由消息
- 使用集群进行工作负载管理

有关帮助您配置集群的更多信息，请参阅第 31 页的『集群提示』。

规划如何使用多个集群传输队列

您可以显式定义传输队列，或者让系统为您生成传输队列。 如果您自己定义传输队列，那么可以对队列定义进行更多控制。 z/OS 在 z/OS 上，您还可以更多地控制保存消息的页集。

定义传输队列

定义传输队列有两种方法:

- 自动使用队列管理器属性 DEFCLXQ，如下所示:

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

  DEFCLXQ (SCTQ) 指示所有集群发送方通道的缺省传输队列为 SYSTEM.CLUSTER.TRANSMIT.QUEUE。 这是缺省值。

  DEFCLXQ (CHANNEL) 指示缺省情况下，每个集群发送方通道都使用名为 SYSTEM.CLUSTER.TRANSMIT.*channel name*。 每个传输队列由队列管理器自动定义。 有关更多信息，请参阅 第 22 页的『自动定义的集群传输队列』。
- 通过使用为 CLCHNAME 属性指定的值来手动定义传输队列。 CLCHNAME 属性指示哪些集群发送方通道应使用传输队列。 请参阅 第 24 页的『规划手动定义的集群传输队列』 以获取更多信息。

我需要什么安全?

要自动或手动启动交换机，您需要启动通道的权限。

要定义用作传输队列的队列，您需要标准 IBM MQ 权限来定义队列。

什么时候是实施变革的合适时机?

在更改集群发送方通道所使用的传输队列时，您需要分配进行更新的时间，请考虑以下几点:

- 通道切换传输队列所需的时间取决于旧传输队列上的消息总数，需要移动的消息数以及消息大小。
- 应用程序可以在发生更改时继续将消息放入传输队列。 这可能导致过渡时间增加。
- 您可以随时更改任何传输队列或 DEFCLXQ 的 CLCHNAME 参数，最好是在工作负载较低时。

  请注意，不会立即发生任何情况。

- 仅当通道启动或重新启动时才会发生更改。 当通道启动时，它将检查当前配置并在需要时切换到新的传输队列。
- 有几个更改可能会改变集群发送方通道与传输队列的关联:
  - 更改传输队列的 CLCHNAME 属性的值，使 CLCHNAME 不那么具体或为空。
  - 更改传输队列的 CLCHNAME 属性的值，使 CLCHNAME 更具体。
  - 正在删除指定了 CLCHNAME 的队列。
  - 正在变更队列管理器属性 DEFCLXQ。

交换机需要多长时间?

在过渡期间，通道的任何消息都将从一个传输队列移至另一个传输队列。 通道切换传输队列所需的时间取决于旧传输队列上的消息总数以及需要移动的消息数。

对于包含几千条消息的队列，移动消息应该需要不到一秒时间。 实际时间取决于消息的数量和大小。 您的队列管理器应该能够以每秒兆字节数移动消息。

应用程序可以在发生更改时继续将消息放入传输队列。 这可能导致过渡时间增加。

必须重新启动每个受影响的集群发送方通道才能使更改生效。 因此，最好在队列管理器不忙时更改传输队列配置，很少消息存储在集群传输队列上。

runswchl 命令或 z/OS 或 z/OS 上的 CSQUTIL 中的 SWITCH CHANNEL (*) STATUS 命令， 可用于查询集群发送方通道的状态以及对其传输队列配置未完成的暂挂更改。

如何实施更改

请参阅 使用多个集群传输队列实现系统， 以获取有关如何自动或手动更改多个集群传输队列的详细信息。

正在撤销更改

有关在迂到问题时如何回退更改的详细信息，请参阅 撤销更改 。

自动定义的集群传输队列
您可以让系统为您生成传输队列。

关于此任务

如果通道没有与之关联的手动定义的集群传输队列，并且您指定 DEFCLXQ (CHANNEL) ，那么当通道启动时，队列管理器会自动为集群发送方通道定义永久动态队列。 模型队列 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 用于自动定义名为 SYSTEM.CLUSTER.TRANSMIT.`ChannelName`。

z/OS 要手动设置集群传输队列，请参阅 第 24 页的『规划手动定义的集群传输队列』。

要点: z/OS

如果将队列管理器迁移到 IBM MQ 8.0，那么队列管理器没有 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE。

首先定义此队列，以便 ALTER QGMR DEFCLXQ (CHANNEL) 命令生效。

以下 JCL 是可用于定义模型队列的代码示例:

```
//CLUSMODL JOB MSGCLASS=H,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=(MVCC)
//MQCMD EXEC PGM=CSQUTIL,REGION=4096K,PARM='CDLK'
//STEPLIB DD DISP=SHR,DSN=SCEN.MQ.V000.COM.BASE.SCSQAUTH
// DD DISP=SHR,DSN=SCEN.MQ.V000.COM.BASE.SCSQANLE
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(CMDINP)
/*
//CMDINP DD *
DEFINE QMODEL( 'SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE' ) +
QSGDISP( QMGR ) +
```

```
* COMMON QUEUE ATTRIBUTES
DESCR( 'SYSTEM CLUSTERING TRANSMISSION MODEL QUEUE' ) +
PUT( ENABLED ) +
DEFPRTY( 5 ) +
DEFPSIST( YES ) +

* MODEL QUEUE ATTRIBUTES
DEFTYPE( PERMDYN ) +

* LOCAL QUEUE ATTRIBUTES
GET( ENABLED ) +
SHARE +
DEFSOPT( EXCL ) +
MSGDLVSQ( PRIORITY ) +
RETINTVL( 999999999 ) +
MAXDEPTH( 999999999 ) +
MAXMSGL( 4194304 ) +
NOHARDENBO +
BOTHRESH( 0 ) +
BOQNAME( ' ' ) +
STGCLASS( 'REMOTE' ) +
USAGE( XMITQ ) +
INDXTYPE( CORRELID ) +
CFSTRUCT( ' ' ) +
MONQ( OFF ) ACCTQ( OFF ) +

* EVENT CONTROL ATTRIBUTES
QDPMAXEV( ENABLED ) +
QDPHIEV( DISABLED ) +
QDEPTHHI( 80 ) +
QDPLOEV( DISABLED ) +
QDEPTHLO( 40 ) +
QSVCIEV( NONE ) +
QSVCINT( 999999999 ) +

* TRIGGER ATTRIBUTES
TRIGGER +
TRIGTYPE( FIRST ) +
TRIGMPRI( 0 ) +
TRIGDPTH( 1 ) +
TRIGDATA( ' ' ) +
PROCESS( ' ' ) +
INITQ( ' ' )
/*
```

过程

1. 使用 *DEFCLXQ* 队列管理器属性。

   有关此属性的更多信息，请参阅 ALTER QMGR。

   有两个选项:

   Sctq
   此选项是缺省值，表示您使用单个 SYSTEM.CLUSTER.TRANSMIT.QUEUE。

   通道
   表示您使用多个集群传输队列。

2. 要切换到新关联:

   - 停止并重新启动通道。
   - 通道使用新的传输队列定义。
   - 通过过渡切换进程将消息从旧队列传输到新的传输队列。

   请注意，任何应用程序消息都将放入旧定义中。

   当旧队列上的消息数达到零时，新消息将直接放在新传输队列上。

3. 要监视切换过程何时完成:

   a) 由通道启动的传输队列切换在后台运行，管理员可以监视队列管理器作业日志以确定它何时完成。

   b) 监视作业记录上的消息以显示交换机的进度。

   c) 要确保仅希望使用此传输队列的通道，请发出命令 DIS CLUSQMGR (*) ，其中，例如，定义传输队列的传输队列属性为 `APPQMGR.CLUSTER1.XMITQ`。

   d) z/OS
   在 CSQUTIL 下使用 SWITCH CHANNEL (*) STATUS 命令。
   此选项告诉您哪些暂挂更改未完成，以及需要在传输队列之间移动多少条消息。

结果

您已设置集群传输队列。

相关任务

第 24 页的『规划手动定义的集群传输队列』
如果您自己定义传输队列，那么可以对定义以及保存消息的页集进行更多控制。

相关参考

ALTER QMGR
DISPLAY CLUSQMGR

规划手动定义的集群传输队列
如果您自己定义传输队列，那么可以对定义以及保存消息的页集进行更多控制。

关于此任务

管理员手动定义传输队列，并使用新的队列属性 CLCHNAME 来定义哪些集群发送方通道将使用此队列作为其传输队列。

请注意， CLCHNAME 可以在开头或结尾包含通配符，以允许将单个队列用于多个通道。

要自动设置集群传输队列，请参阅 第 22 页的『自动定义的集群传输队列』。

过程

1. 例如，输入以下内容：

   ```
   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)
   ```

   提示: 您需要规划用于传输队列的页集 (和缓冲池)。 您可以针对不同的队列具有不同的页集，并在它们之间提供隔离，因此一个页集填满，不会影响其他页集中的传输队列。

   请参阅 使用集群传输队列和集群发送方通道， 以获取有关每个通道如何选择相应队列的信息。

   当通道启动时，它会将其关联切换到新的传输队列。 为了确保不丢失消息，队列管理器会自动按顺序将消息从旧集群传输队列传输到新传输队列。
2. 使用 CSQUTIL SWITCH 函数来更改为新的关联。

   请参阅 切换与集群发送方通道 (SWITCH) 关联的传输队列 以获取更多信息。

   a) 停止要更改其传输队列的通道， 以使其处于 STOPPED 状态。

      例如：

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

   b) 更改传输队列上的 CLCHNAME(XXXX) 属性。
   c) 使用 SWITCH 函数来切换消息或监视正在发生的情况。

      使用命令

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

      以在不启动通道的情况下移动消息。
   d) 启动一个或多个通道，并检查该通道是否正在使用正确的队列。

      例如：

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

提示:

- 以下进程使用 CSQUTIL SWITCH 功能; 有关更多信息，请参阅 切换与集群发送方通道 (SWITCH) 关联的传输队列。

  您不必使用此函数，但使用此函数会提供更多选项:

  – 使用 SWITCH CHANNEL (*) STATUS 可轻松识别集群发送方通道的切换状态。 它允许管理员查看当前正在切换的通道，以及那些具有暂挂切换的通道在这些通道下次启动时生效。

    如果没有此功能，那么管理员需要使用多个 DISPLAY 命令，然后处理生成的输出以确定此信息。 管理员还可以确认配置更改是否具有必需的结果。

  – 如果使用 CSQUTIL 来启动交换机，那么 CSQUTIL 将继续监视此操作的进度，并且仅在交换机完成时结束。

    这样可以更轻松地批量执行这些操作。 此外，如果运行 CSQUTIL 以切换多个通道，那么 CSQUTIL 会按顺序执行这些操作; 与并行运行的多个交换机相比，这对您的企业的影响较小。

结果

您已设置集群传输队列。

访问控制和多个集群传输队列

在应用程序将消息放入远程集群队列时选择三种检查方式。 这些方式是针对集群队列进行远程检查，针对 SYSTEM.CLUSTER.TRANSMIT.QUEUE 进行本地检查，或者针对集群队列或集群队列管理器的本地概要文件进行检查。

IBM MQ 允许您选择在本地或本地和远程检查用户是否有权将消息放入远程队列。 典型的 IBM MQ 应用程序仅使用本地检查，并且依赖于远程队列管理器信任对本地队列管理器进行的访问检查。 如果未使用远程检查，那么将使用远程消息通道进程的权限将消息放入目标队列。 要使用远程检查，必须将接收通道的放置权限设置为上下文安全性。

将针对应用程序打开的队列进行本地检查。 在分布式排队中，应用程序通常会打开远程队列定义，并且会针对远程队列定义进行访问检查。 如果将消息与完整路由头放在一起，那么将针对传输队列进行检查。 如果应用程序打开不在本地队列管理器上的集群队列，那么没有要检查的本地对象。 将针对集群传输队列 SYSTEM.CLUSTER.TRANSMIT.QUEUE 执行访问控制检查。 即使有多个集群传输队列，也会针对 SYSTEM.CLUSTER.TRANSMIT.QUEUE 对远程集群队列进行本地访问控制检查。

选择本地或远程检查是两个极端之间的选择。 远程检查是细颗粒度的。 每个用户都必须在集群中的每个队列管理器上具有访问控制概要文件才能放入任何集群队列。 本地检查是粗粒度的。 每个用户在其连接到的队列管理器上只需要一个集群传输队列的访问控制概要文件。 通过该概要文件，他们可以将消息放入任何集群中任何队列管理器上的任何集群队列。

管理员还有另一种方法来设置集群队列的访问控制。 您可以使用 **setmqaut** 命令在集群中的任何队列管理器上为集群队列创建安全概要文件。 如果在本地打开远程集群队列 (仅指定队列名称) ，那么概要文件将生效。 您还可以为远程队列管理器设置概要文件。 如果执行此操作，那么队列管理器可以通过提供标准名称来检查打开集群队列的用户的概要文件。

仅当将队列管理器节 **ClusterQueueAccessControl** 更改为 RQMName 时，新概要文件才有效。 缺省值为 Xmitq。 您必须为现有应用程序使用集群队列的所有集群队列创建概要文件。 如果将节更改为 RQMName 而不创建概要文件，那么应用程序可能会失败。

提示: 集群队列访问检查不适用于远程排队。 仍会针对本地定义进行访问检查。 这些更改意味着您可以遵循相同的方法来配置对集群队列和集群主题的访问检查。 z/OS 这些更改还使集群队列的访问检查方法与 z/OS 更紧密地对齐。 用于在 z/OS 上设置访问权检查的命令有所不同，但这两个命令都检查对概要文件的访问权，而不是对对象本身的访问权。

相关概念

第 41 页的『集群: 使用多个集群传输队列进行应用程序隔离』
您可以在集群中的队列管理器之间隔离消息流。 您可以将由不同集群发送方通道传输的消息放入不同的集群传输队列中。 您可以在单个集群中使用此方法，也可以将此方法与重叠的集群配合使用。 本主题提供了一些示例和一些最佳实践，以指导您选择使用方法。

相关任务

设置 ClusterQueueAccessControl

集群与分布式排队的比较

比较需要定义的组件，以使用分布式排队和集群来连接队列管理器。

如果不使用集群，那么队列管理器是独立的，并使用分布式排队进行通信。 如果一个队列管理器需要将消息发送到另一个队列管理器，那么必须定义:

- 传输队列
- 到远程队列管理器的通道

第 26 页的图 3 显示了分布式排队所需的组件。

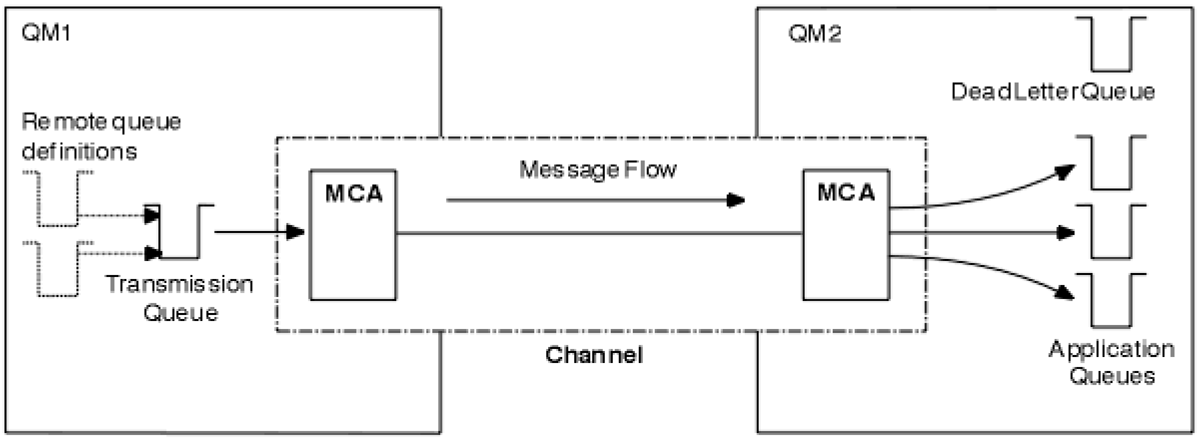


图 *3:* 分布式 排队

如果对集群中的队列管理器进行分组，那么任何队列管理器上的队列都可供集群中的任何其他队列管理器使用。 任何队列管理器都可以在没有显式定义的情况下向同一集群中的任何其他队列管理器发送消息。 您不会为每个目标提供通道定义，远程队列定义或传输队列。 集群中的每个队列管理器都有一个传输队列，可以从中将消息传输到集群中的任何其他队列管理器。 集群中的每个队列管理器仅需要定义:

- 要在其上接收消息的一个集群接收方通道
- 一个集群发送方通道，通过该通道引入自身并了解集群

用于设置集群与分布式排队的定义

查看 第 27 页的图 4，其中显示了四个队列管理器，每个队列管理器具有两个队列。 请考虑使用分布式排队连接这些队列管理器所需的定义数。 比较将同一网络设置为集群所需的定义数。

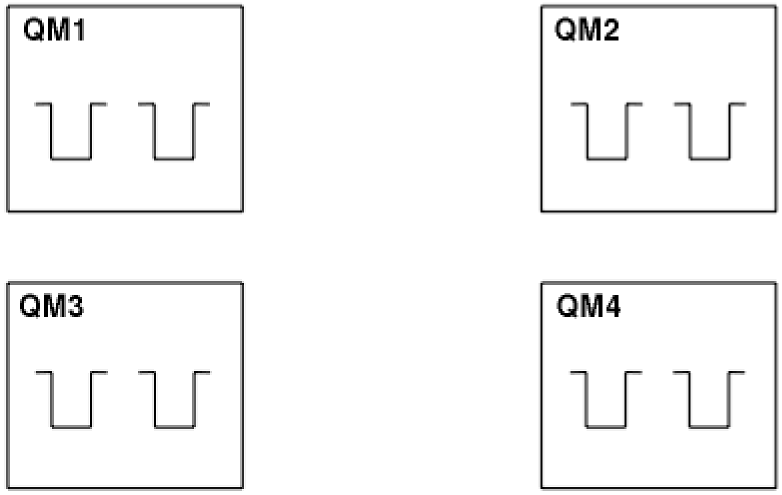


图 *4:* 由四个队列管理器组成的网络

用于使用分布式排队来设置网络的定义

要使用分布式排队来设置 第 26 页的图 3 中显示的网络，您可能具有以下定义:

表 *2:* 分布式排队的定义

描述	每个队列管理器的数目	总数
要将消息发送到其他每个队列管理器的通道的发送方通道定义	3	12
要在其上从每个其他队列管理器接收消息的通道的接收方通道定义	3	12
传输队列到每个其他队列管理器的传输队列定义	3	12
每个本地队列的本地队列定义	2	8
此队列管理器要将消息放入的每个远程队列的远程队列定义	6	24

您可以使用通用接收方通道定义来减少此数目的定义。 每个队列管理器上的最大定义数可能多达 17 个，此网络的定义总数为 68 个。

用于使用集群设置网络的定义

要使用集群设置 第 26 页的图 3 中显示的网络，您需要以下定义:

表 *3:* 集群的定义

描述	每个队列管理器的数目	总数
要将消息发送到存储库队列管理器的通道的集群发送方通道定义	1	4
要在其上从集群中的其他队列管理器接收消息的通道的集群接收方通道定义	1	4
每个本地队列的本地队列定义	2	8

要设置此队列管理器集群 (具有两个完整存储库) ，每个队列管理器上需要四个定义，总共需要十六个定义。 您还需要更改两个队列管理器的队列管理器定义，以使它们成为集群的完整存储库队列管理器。

仅需要一个 `CLUSSDR` 和一个 `CLUSRCVR` 通道定义。 定义集群时，您可以添加或除去队列管理器 (存储库队列管理器除外) ，而不会对其他队列管理器造成任何干扰。

使用集群可减少设置包含许多队列管理器的网络所需的定义数。

定义较少，因此发生错误的风险较低:

- 对象名始终匹配，例如发送方/接收方对中的通道名称。
- 通道定义中指定的传输队列名称始终与正确的传输队列定义或远程队列定义中指定的传输队列名称相匹配。
- QREMOTE 定义始终指向远程队列管理器上的正确队列。

设置集群后，您可以将集群队列从一个队列管理器移至集群中的另一个队列管理器，而不必对任何其他队列管理器执行任何系统管理工作。 没有可能忘记删除或修改通道，远程队列或传输队列定义。 您可以将新的队列管理器添加到集群中，而不会对现有网络造成任何干扰。

如何选择集群队列管理器以保存完整存储库

在每个集群中，必须至少选择一个队列管理器，最好选择两个队列管理器以保存完整存储库。 除了最例外的情况外，两个完整的存储库足以应对所有情况。 如果可能，请选择在稳健且永久连接的平台上托管的队列管理器，这些队列管理器不会发生重合的中断，并且在地理位置上处于中心位置。 另外，请考虑将系统专用为完整存储库主机，而不要将这些系统用于任何其他任务。

完整存储库 是保持集群状态完整的队列管理器。 要共享此信息，每个完整存储库都由 CLUSSDR 通道 (及其相应的 CLUSRCVR 定义) 连接到集群中的每个其他完整存储库。 您必须手动定义这些通道。

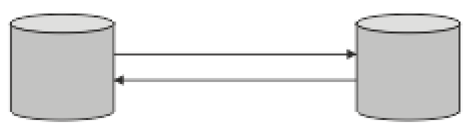


图 *5:* 两个已连接的完整存储库。

集群中的每个其他队列管理器都会在 部分存储库中维护其当前了解的集群状态的图片。 这些队列管理器使用任何两个可用的完整存储库发布有关自身的信息，并请求有关其他队列管理器的信息。 如果所选完整存储库不可用，那么将使用另一个完整存储库。 当所选的完整存储库再次变为可用时，它将从其他存储库收集最新的新信息和已更改的信息，以便它们保持同步。 如果所有完整存储库都不起作用，那么其他队列管理器将使用其部分存储库中的信息。 但是，它们仅限于使用它们所拥有的信息; 无法处理新信息和更新请求。 当完整存储库重新连接到网络时，将交换消息以更新所有存储库 (包括完整存储库和部分存储库)。

规划完整存储库的分配时，请包括以下注意事项:

- 选择用于保存完整存储库的队列管理器需要可靠且受管。 选择在健壮且永久连接的平台上托管的队列管理器。
- 请考虑托管完整存储库的系统的计划中断，并确保这些系统不会发生重合的中断。
- 考虑网络性能: 选择在地理位置上处于中心位置的队列管理器，或者选择与集群中的其他队列管理器共享同一系统的队列管理器。
- 请考虑队列管理器是否是多个集群的成员。 使用同一队列管理器来托管多个集群的完整存储库在管理上可能很方便，前提是此优势与您期望队列管理器的繁忙程度相平衡。
- 请考虑将某些系统专用于仅包含完整存储库，而不将这些系统用于任何其他任务。 这将确保这些系统只需要维护队列管理器配置，并且不会从维护其他业务应用程序的服务中除去。 它还确保维护存储库的任务不会与系统资源的应用程序竞争。 这在大型集群 (例如，包含超过 1000 个队列管理器的集群) 中特别有用，在这些集群中，完整存储库在维护集群状态方面的工作负载要高得多。

可能有两个以上完整存储库，但很少建议使用。 虽然对象定义 (即，队列，主题和通道) 流向所有可用的完整存储库，但请求仅从部分存储库流向最多两个完整存储库。 这意味着当定义了两个以上的完整存储库，并且任何两个完整存储库变为不可用时，某些部分存储库可能不会收到它们期望的更新。 请参阅 MQ 集群: 为什么只有两个完整存储库?

您可能发现定义两个以上完整存储库很有用的一种情况是在将现有完整存储库迁移到新硬件或新队列管理器时。 在这种情况下，您应该先引入替换完整存储库，并确认它们已完全填充，然后再移除先前的完整存储库。 每当添加完整存储库时，请记住必须使用 CLUSSDR 通道将其直接连接到其他所有完整存储库。

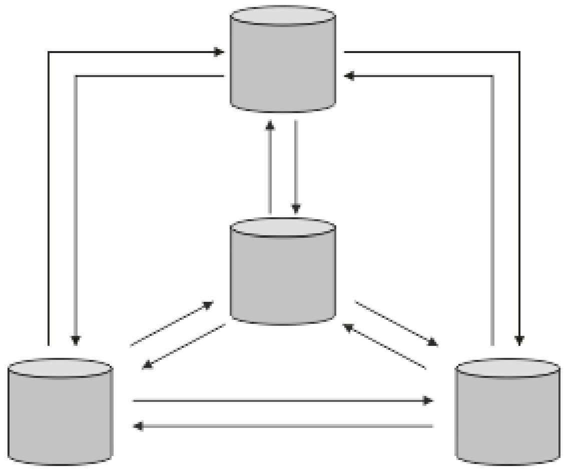


图 *6:* 两个以上已连接的完整存储库

相关信息

MQ 集群: 为什么只有两个完整存储库?

MQ 集群有多大?

组织集群

选择要链接到哪个完整存储库的队列管理器。 请考虑性能影响，队列管理器版本以及是否需要多个 CLUSSDR 通道。

在选择队列管理器以保存完整存储库之后，您需要决定要链接到哪个完整存储库的队列管理器。 CLUSSDR 通道定义将队列管理器链接到完整存储库，从中可以发现集群中的其他完整存储库。 从此，队列管理器将消息发送到任意两个完整存储库。 它始终首先尝试使用其具有 CLUSSDR 通道定义的通道。 您可以选择将队列管理器链接到任一完整存储库。 在选择时，请考虑配置的拓扑以及队列管理器的物理或地理位置。

由于所有集群信息都将发送到两个完整存储库，因此可能存在您想要生成第二个 CLUSSDR 通道定义的情况。 您可以在集群中定义第二个 CLUSSDR 通道，该集群具有分布在广泛区域中的许多完整存储库。 然后，您可以控制将信息发送到哪个两个完整存储库。

集群命名约定

请考虑使用用于标识队列管理器所属集群的命名约定来命名同一集群中的队列管理器。 对通道名称使用类似的命名约定，并对其进行扩展以描述通道特征。

命名 MQ 集群时的最佳实践

虽然集群名称最多可以有 48 个字符，但在将命名约定应用于其他对象时，相对较短的集群名称很有用。 请参阅 第 30 页的『选择集群通道名称时的最佳实践』。

在选择集群名称时，通常有助于表示集群的 "用途" (可能是长期存在) 而不是 "内容"。 例如， "B2BPROD" 或 "ACTTEST" 而不是 "QM1_QM2_QM3_CLUS"。

选择集群队列管理器名称时的最佳实践

如果要从头开始创建新集群及其成员，请考虑队列管理器的命名约定，以反映其集群使用情况。 每个队列管理器都必须具有不同的名称。 但是，您可以为集群中的队列管理器提供一组类似的名称，以帮助识别和记住逻辑分组 (例如， "ACTTQM1， ACTTQM2")。

相对较短的队列管理器名称 (例如，少于 8 个字符) 可帮助您选择对通道名称使用下一节中描述的约定或类似的约定。

选择集群通道名称时的最佳实践

由于队列管理器和集群的名称最多可包含 48 个字符，并且通道名称限制为 20 个字符，因此在第一个命名对象时请注意避免在项目中途更改命名约定 (请参阅先前部分)。

定义通道时，请记住，在集群中的任何队列管理器上自动创建的集群发送方通道会从集群中的接收队列管理器上配置的相应集群接收方通道中获取其名称，因此这些通道必须是唯一的，并且 在集群中的远程队列管理器上 有意义。

一种常见方法是使用以集群名称开头的队列管理器名称。 例如，如果集群名称为 CLUSTER1，队列管理器为 QM1， QM2，那么集群接收方通道为 CLUSTER1.QM1， CLUSTER1.QM2。

如果通道具有不同的优先级或使用不同的协议，那么可以扩展此约定。 例如:

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

在此示例中， S1 可能是第一个 SNA 通道， N3 可能是网络优先级为 3 的 NetBIOS 通道， T4 可能是使用 IPV4 网络的 TCP IP。

命名共享通道定义

可以跨多个集群共享单个通道定义，在这种情况下，此处建议的命名约定需要修改。 但是，如 管理通道定义 中所述，在任何情况下，通常最好为每个集群定义离散通道。

较旧的通道命名约定

在集群环境之外，使用 "FROMQM.TO.TARGETQM" 命名约定在历史上是常见的，因此您可能会发现现有集群使用了类似的内容 (例如 CLUSTER.TO.TARGET)。 建议不要将此作为新集群命名方案的一部分，因为这将进一步减少可用字符以在通道名称中传递 "有用" 信息。

z/OS IBM MQ for z/OS 上的通道名称

您可以定义 VTAM 通用资源或 *Dynamic Domain Name Server* (DDNS) 通用名称。 您可以使用通用名称来定义连接名称。 但是，创建集群接收方定义时，请勿使用通用连接名称。

对集群接收方定义使用通用连接名称的问题如下: 如果使用通用 CONNAME 定义 CLUSRCVR，那么不保证 CLUSSDR 通道指向您期望的队列管理器。 初始 CLUSSDR 可能最终指向队列共享组中的任何队列管理器，而不一定是托管完整存储库的队列管理器。 如果通道再次开始尝试连接，那么它可能会重新连接到具有相同通用名称的其他队列管理器，从而中断消息流。

z/OS 队列共享组和集群

共享队列可以是集群队列，队列共享组中的队列管理器也可以是集群队列管理器。

在 IBM MQ for z/OS 上，可以将队列管理器分组到队列共享组中。 队列共享组中的队列管理器可以定义最多由 32 个队列管理器共享的本地队列。

共享队列也可以是集群队列。 此外，队列共享组中的队列管理器也可以位于一个或多个集群中。

您可以定义 VTAM 通用资源或 *Dynamic Domain Name Server* (DDNS) 通用名称。 您可以使用通用名称来定义连接名称。 但是，创建集群接收方定义时，请勿使用通用连接名称。

对集群接收方定义使用通用连接名称的问题如下: 如果使用通用 CONNAME 定义 CLUSRCVR，那么不保证 CLUSSDR 通道指向您期望的队列管理器。 初始 CLUSSDR 可能最终指向队列共享组中的任何队列管理器，而不一定是托管完整存储库的队列管理器。 如果通道再次开始尝试连接，那么它可能会重新连接到具有相同通用名称的其他队列管理器，从而中断消息流。

无法启动使用组侦听器端口的 CLUSRCVR 通道，因为如果是这种情况，那么无法告知 CLUSRCVR 每次将连接到哪个队列管理器。 不共享在其上保留有关集群的信息的集群系统队列。 每个队列管理器都有自己的队列管理器。

集群通道不仅用于传输应用程序消息，还用于传输有关集群设置的内部系统消息。 集群中的每个队列管理器都必须接收这些内部系统消息以正确参与集群，因此需要它自己的唯一 CLUSRCVR 通道来接收这些消息。

共享 CLUSRCVR 可以在队列共享组 (QSG) 中的任何队列管理器上启动，因此会导致向 QSG 队列管理器提供不一致的内部系统消息，这意味着没有任何消息可以正确参与集群。 为了确保不能使用共享 CLUSRCVR 通道，任何尝试都将失败并显示 CSQX502E 消息。

重叠集群

重叠的集群提供了额外的管理功能。 使用名称列表来减少管理重叠集群所需的命令数。

您可以创建重叠的集群。 您可以定义重叠集群的原因有很多; 例如:

- 让不同的组织有自己的管理。
- 允许单独管理独立应用程序。
- 创建服务类。

在 第 31 页的图 7 中，队列管理器 STF2 是这两个集群的成员。 当队列管理器是多个集群的成员时，您可以利用名称列表来减少所需的定义数。 名称列表包含名称列表，例如，集群名称。 您可以创建命名集群的名称列表。 在 STF2 的 ALTER QMGR 命令上指定名称列表，以使其成为两个集群的完整存储库队列管理器。

如果网络中有多个集群，那么必须为它们提供不同的名称。 如果两个具有相同名称的集群曾经合并，那么无法再次将它们分开。 给集群和通道不同的名称也是一个好主意。 当您查看 DISPLAY 命令的输出时，可以更轻松地对它们进行区分。 队列管理器名称在集群中必须是唯一的，才能使其正常工作。

定义服务类

想象一下，有一个大学，每个工作人员和每个学生都有一个队列管理器。 工作人员之间的消息是在具有高优先级和高带宽的通道上传输。 学生之间的信息是在更便宜，更慢的渠道上旅行。 您可以使用传统的分布式排队技术来设置此网络。 IBM MQ 通过查看目标队列名称和队列管理器名称来选择要使用的通道。

要明确区分职员和学生，您可以将他们的队列管理器分组到两个集群中，如 第 31 页的图 7 中所示。 IBM MQ 仅通过人员集群中定义的通道将消息移至该集群中的会议队列。 学生集群中 gossip 队列的消息将通过该集群中定义的通道，并接收相应的服务类。

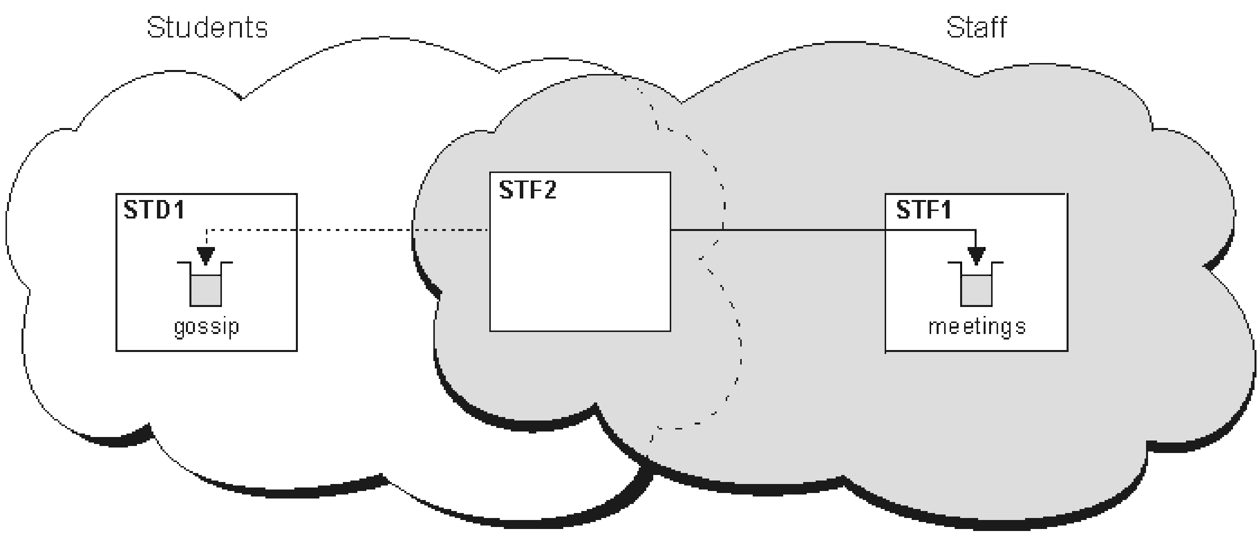


图 *7:* 服务等级

集群提示

在使用集群之前，您可能需要对系统或应用程序进行一些更改。 与分布式排队的行为既有相似之处，也有不同之处。

- 必须将手动配置定义添加到集群外部的队列管理器，以便它们访问集群队列。
- 如果合并具有相同名称的两个集群，那么不能再次将其分隔开。 因此，建议为所有集群提供唯一名称。
- 如果消息到达队列管理器，但那里没有要接收该消息的队列，那么会将该消息放在死信队列上。 如果没有死信队列，那么通道将失败并重试。 死信队列的使用与分布式排队相同。

- 维护持久消息的完整性。 消息不会因使用集群而重复或丢失。
- 使用集群可减少系统管理。 集群使您能够轻松地将更大的网络与更多的队列管理器连接起来，而不需要您考虑使用分布式排队。 如果尝试在集群中的每个队列管理器之间启用通信，那么可能会消耗过多的网络资源。
- 如果使用以树结构显示队列管理器的 IBM MQ Explorer，那么大型集群的视图可能很繁琐。
- Multi 分发列表的目的是使用单个 MQPUT 命令将同一消息发送到多个目标。 IBM MQ for Multiplatforms 上支持分发列表。 您可以将分发列表与队列管理器集群配合使用。 在集群中，所有消息在 MQPUT 时展开。 在网络流量方面的优势，并不像在非集群环境中那么大。 分发列表的优点是不需要手动定义众多通道和传输队列。
- 如果要使用集群来平衡工作负载，请检查应用程序。 查看它们是要求消息由特定队列管理器处理还是按特定顺序处理。 此类应用程序据说具有消息亲缘关系。 您可能需要先修改应用程序，然后才能在复杂集群中使用这些应用程序。
- 您可以选择在 MQOPEN 上使用 MQOO_BIND_ON_OPEN 选项，以强制将消息发送到特定目标。 如果目标队列管理器不可用，那么直到队列管理器再次可用之后才会传递消息。 由于存在重复的风险，因此不会将消息路由到另一个队列管理器。
- 如果队列管理器要托管集群存储库，那么您需要知道其主机名或 IP 地址。 在加入集群的其他队列管理器上创建 CLUSSDR 定义时，必须在 CONNAME 参数中指定此信息。 如果使用 DHCP，那么 IP 地址可能会更改，因为 DHCP 可以在每次重新启动系统时分配新的 IP 地址。 因此，不得在 CLUSSDR 定义中指定 IP 地址。 即使所有 CLUSSDR 定义都指定了主机名而不是 IP 地址，这些定义仍然不可靠。 DHCP 不一定使用新地址更新主机的 DNS 目录条目。 如果必须将队列管理器指定为使用 DHCP 的系统上的完整存储库，请安装保证 DNS 目录保持最新的软件。
- 请勿使用通用名称，例如 VTAM 通用资源或动态域名服务器 (DDNS) 通用名称作为通道的连接名称。 如果您这样做，那么您的通道可能会连接到与预期不同的队列管理器。
- 只能从本地集群队列获取消息，但可以将消息放入集群中的任何队列。 如果打开队列以使用 MQGET 命令，那么队列管理器将打开本地队列。
- 如果设置了简单 IBM MQ 集群，那么不需要更改任何应用程序。 应用程序可以在 MQOPEN 调用上命名目标队列，并且不需要了解队列管理器的位置。 如果为工作负载管理设置集群，那么必须复审应用程序并根据需要对其进行修改。
- 您可以使用 DISPLAY CHSTATUS 和 DISPLAY QSTATUS **runmqsc** 命令来查看通道或队列的当前监视和状态数据。 监控信息可用于帮助评估系统的性能和运行状况。 监视由队列管理器，队列和通道属性控制。 可以使用 MONACLS 队列管理器属性来监视自动定义的集群发送方通道。

相关概念

集群

第 26 页的『集群与分布式排队的比较』
比较需要定义的组件，以使用分布式排队和集群来连接队列管理器。

集群的组件

相关任务

配置队列管理器集群

设置新集群

队列管理器存储库保留信息的时间长度?

队列管理器存储库将信息保留 30 天。 自动进程高效地刷新正在使用的信息。

当队列管理器发出有关自身的一些信息时，完整和部分存储库队列管理器会将该信息存储 30 天。 例如，当队列管理器发布创建新队列的广告时，将发送信息。 为了防止此信息到期，队列管理器将在 27 天后自动重新发送有关自身的所有信息。 如果部分存储库在 30 天生存期内发送新的信息部分请求，那么到期时间仍为原来的 30 天。

当信息到期时，不会立即将其从存储库中除去。 而是举行 60 天的宽限期。 如果在宽限期内未收到任何更新，那么将除去该信息。 宽限期允许队列管理器可能在到期日期暂时停止服务。 如果队列管理器与集群断开连接超过 90 天，那么它将停止成为集群的一部分。 但是，如果它重新连接到网络，那么它将再次成为集群的一部分。 完整存储库不使用已到期的信息来满足来自其他队列管理器的新请求。

同样，当队列管理器从完整存储库发送对最新信息的请求时，该请求将持续 30 天。 27 天后， IBM MQ 将检查请求。 如果在 27 天内引用了该参数，那么将自动刷新该参数。 如果没有，那么它将保留为到期，并由队列管理器刷新 (如果再次需要)。 到期请求会阻止来自休眠队列管理器的信息请求的累积。

注: 您应该下载并安装 APAR PH43191 的 PTF ，这将修订计算预订到期时间时发生的系统错误。 这些错误可能导致预订提前到期 (导致发出消息 CSQX456I) ，或在对象到期后到期 (导致错误 MQRC 2085 (MQRC_UNKNOWN_OBJECT) 错误)。

对于大型集群，如果许多队列管理器同时自动重新发送有关其自身的所有信息，那么这可能具有破坏性。 请参阅在大型集群中刷新可能会影响集群的性能和可用性。

相关概念

第 58 页的『集群：使用 REFRESH CLUSTER 最佳实践』

使用 **REFRESH CLUSTER** 命令可废弃有关集群的所有本地保存的信息，并从集群中的完整存储库重建该信息。 您不应该使用此命令，除非在特殊情况下。 如果确实需要使用它，那么有关如何使用它有特殊注意事项。 此信息是基于测试和客户反馈的指南。

示例集群

第一个示例显示了两个队列管理器的最小可能集群。 第二个和第三个示例显示了三个队列管理器集群的两个版本。

最小的可能集群仅包含两个队列管理器。 在这种情况下，两个队列管理器都包含完整存储库。 您只需要几个定义即可设置集群，然而每个队列管理器都有高度的自主性。

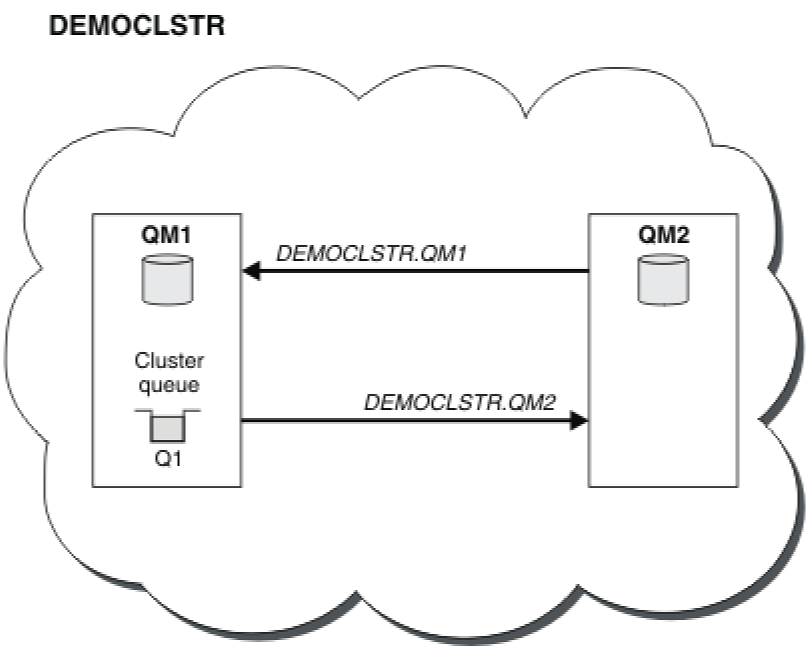


图 *8:* 两个队列管理器的小型集群

- 队列管理器可以具有长名称，例如 LONDON 和 NEWYORK。 z/OS 在 IBM MQ for z/OS 上，队列管理器名称限制为四个字符。
- 通常在单独的机器上配置每个队列管理器。 但是，您可以在同一机器上具有多个队列管理器。

有关设置类似示例集群的指示信息，请参阅 设置新集群。

第 34 页的图 9 显示了名为 CLSTR1 的集群的组件。

- 在此集群中，有三个队列管理器: QM1， QM2 和 QM3。
- 有关集群中所有队列管理器和集群相关对象的信息的 QM1 和 QM2 主机存储库。 它们称为 完整存储库队列管理器。 存储库在图中由阴影柱面表示。

- QM2 和 QM3 托管一些可供集群中任何其他队列管理器访问的队列。 可供集群中任何其他队列管理器访问的队列称为 集群队列。 在图中，集群队列由阴影队列表示。 可以从集群中的任何位置访问集群队列。 IBM MQ 集群代码确保在引用集群队列的任何队列管理器上创建集群队列的远程队列定义。

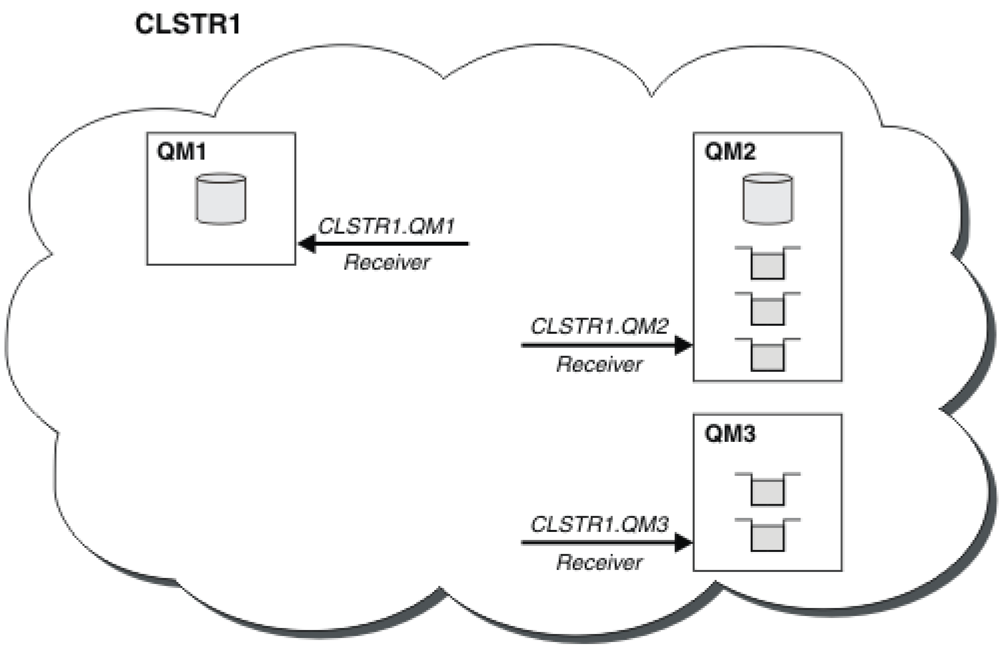
  与分布式排队一样，应用程序使用 MQPUT 调用将消息放在集群中任何队列管理器上的集群队列上。 应用程序使用 MQGET 调用仅在队列所在的队列管理器上从集群队列检索消息。
- 每个队列管理器都有一个手动创建的定义，用于接收名为 *cluster_name. queue_manager_name* 的通道的接收端，在该通道上可以接收消息。 在接收队列管理器上， *cluster_name. queue_manager_name* 是集群接收方通道。 集群接收方通道类似于分布式排队中使用的接收方通道; 它接收队列管理器的消息。 此外，它还接收有关集群的信息。
- 

  图 *9:* 队列管理器集群
- 在 第 35 页的图 10 中，每个队列管理器还具有通道发送端的定义。 它连接到其中一个完整存储库队列管理器的集群接收方通道。 在发送队列管理器上， *cluster_name. queue_manager_name* 是集群发送方通道。 QM1 和 QM3 将集群发送方通道连接到 CLSTR1.QM2，请参阅虚线 “2”。

  QM2 具有连接到 CLSTR1.QM1 的集群发送方通道，请参阅虚线 “3”。 集群发送方通道类似于分布式排队中使用的发送方通道; 它将消息发送到接收队列管理器。 此外，它还会发送有关集群的信息。

  一旦定义了通道的集群接收方端和集群发送方端，该通道将自动启动。

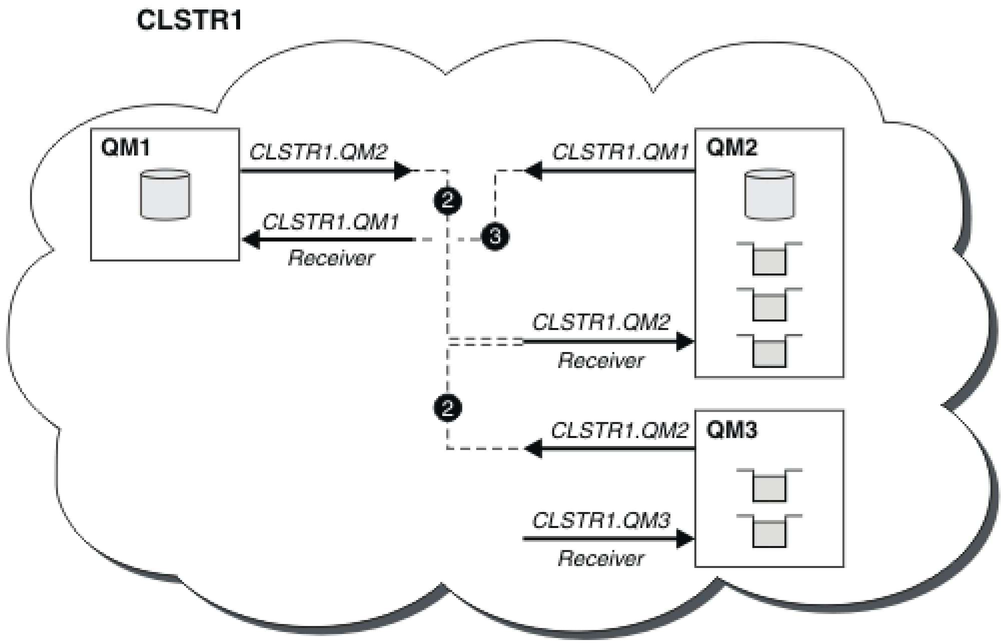


图 *10:* 具有发送方通道的队列管理器集群

在本地队列管理器上定义集群发送方通道会将该队列管理器引入到其中一个完整存储库队列管理器。 完整存储库队列管理器相应地更新其完整存储库中的信息。 然后，它会自动将集群发送方通道创建回原始队列管理器，并发送有关集群的队列管理器信息。 因此，队列管理器学习有关集群的信息，而集群学习有关队列管理器的信息。

再次查看 第 34 页的图 9。 假设连接到队列管理器 QM3 的应用程序想要将一些消息发送到位于 QM2 的队列。 QM3 首次必须访问这些队列时，它会通过查阅完整存储库来发现这些队列。 在此情况下，完整存储库为 QM2，可使用发送方通道 CLSTR1.QM2 进行访问。 通过存储库中的信息，它可以自动为这些队列创建远程定义。 如果队列位于 QM1 上，那么此机制仍有效，因为 QM2 是完整存储库。 完整存储库具有集群中所有对象的完整记录。 在后一种情况下， QM3 还将自动创建与 QM1 上的集群接收方通道对应的集群发送方通道，从而允许两者之间进行直接通信。

第 36 页的图 11 显示了同一集群，以及自动创建的两个集群发送方通道。 集群发送方通道由与集群接收方通道 CLSTR1.QM3 连接的两条虚线表示。 它还显示集群传输队列 SYSTEM.CLUSTER.TRANSMIT.QUEUE，QM1 使用该队列来发送其消息。 集群中的所有队列管理器都有一个集群传输队列，可以从该队列将消息发送到同一集群中的任何其他队列管理器。

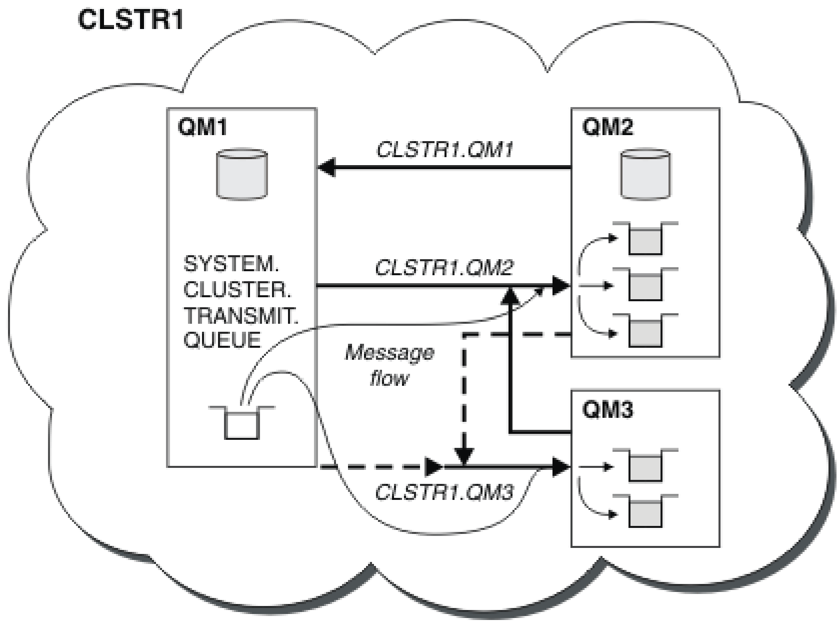


图 *11:* 队列管理器集群，显示自动定义的通道

注: 其他图仅显示您为其进行手动定义的通道的接收端。 由于发送端大多在需要时自动定义，因此省略了发送端。 大多数集群发送方通道的自动定义对于集群的功能和效率至关重要。

相关概念

第 26 页的『集群与分布式排队的比较』
比较需要定义的组件，以使用分布式排队和集群来连接队列管理器。

集群的组件

相关任务

配置队列管理器集群

设置新集群

集群*:* 最佳实践

集群提供了一种用于互连队列管理器的机制。 本节中描述的最佳实践基于客户的测试和反馈。

成功的集群设置依赖于良好的规划和对 IBM MQ 基础知识 (例如， 良好的应用程序管理和网络设计) 的全面了解。 请确保您熟悉相关主题中的信息，然后再继续。

相关概念

分布式排队和集群

集群

相关任务

第 20 页的『设计集群』
集群提供了一种用于以简化初始配置和持续管理的方式互连队列管理器的机制。 必须对集群进行精心设计，以确保它们正常运行，并确保它们达到所需的可用性和响应能力级别。

监视集群

集群*:* 重叠集群的特殊注意事项
本主题提供有关规划和管理 IBM MQ 集群的指导。 此信息是基于测试和客户反馈的指南。

集群所有权

请先熟悉重叠的集群，然后再阅读以下信息。 请参阅 第 31 页的『重叠集群』 和 配置集群之间的消息路径 以获取必要的信息。

在配置和管理由重叠集群组成的系统时，最好遵循以下内容:

- 尽管如前所述， IBM MQ 集群 "松散耦合" ，但将集群视为单个管理单元很有用。 使用此概念是因为各个队列管理器上的定义之间的交互对于集群的顺利运行至关重要。 例如: 使用工作负载均衡的集群队列时，单个管理员或团队必须了解消息的完整可能目标集，这取决于分布在整个集群中的定义。 更重要的是，集群发送方/接收方通道对必须自始至终兼容。
- 考虑到先前的这一概念; 当多个集群满足 (将由单独的团队/个人管理) 时，重要的是要有明确的策略来控制网关队列管理器的管理。
- 将重叠集群视为单个名称空间很有用: 通道名称和队列管理器名称在单个集群中必须唯一。 在整个拓扑中唯一时，管理更容易。 最好遵循合适的命名约定， 第 29 页的『集群命名约定』中描述了可能的约定。
- 有时，行政和系统管理合作是必不可少的/不可避免的: 例如，拥有需要重叠的不同集群的组织之间的合作。 明确了解谁拥有哪些内容和可执行的规则/约定有助于集群在重叠集群时平稳运行。

重叠集群: 网关

通常，单个集群比多个集群更易于管理。 因此，创建大量小型集群 (例如，针对每个应用程序创建一个集群) 通常是要避免的。

但是，要提供服务类，您可以实现重叠集群。 例如:

- 如果您有同心集群，其中较小的集群用于发布/预订。 请参阅 如何调整系统大小 以获取更多信息。
- 如果某些队列管理器将由不同的团队管理。 请参阅上一部分 第 37 页的『集群所有权』以获取更多信息。
- 如果从组织或地理角度讲是有意义的。
- 如果等效集群使用名称解析，例如在现有集群中实现 TLS 时。

重叠集群没有安全优势; 允许由两个不同团队管理的集群重叠，有效地加入团队以及拓扑。 任何:

- 在此类集群中公布的名称可供其他集群访问。
- 在一个集群中公布的名称可以在另一个集群中公布，以提取符合条件的消息。
- 可以从网关所属的任何集群解析与网关相邻的队列管理器上的非广告对象。

名称空间是两个集群的并集，必须视为单个名称空间。 因此，重叠集群的所有权在两个集群的所有管理员之间共享。

当系统包含多个集群时，可能需要将消息从一个集群中的队列管理器路由到另一个集群中的队列管理器上的队列。 在此情况下，必须以某种方式互连多个集群: 要遵循的良好模式是在集群之间使用网关队列管理器。 这种安排避免了建立起一个难以管理的点对点通道网格，并为管理安全策略等问题提供了一个好的场所。 实现这一安排有两种不同的方法:

1. 使用第二个集群接收方定义在两个集群中放置一个 (或多个) 队列管理器。 这种安排涉及较少的管理定义，但如前所述，意味着重叠集群的所有权由两个集群的所有管理员共享。
2. 使用传统的点到点通道将集群 1 中的队列管理器与集群 2 中的队列管理器配对。

在任何一种情况下，都可以使用各种工具来适当路由流量。 特别是，可以使用队列或队列管理器别名来路由到其他集群，并且具有空白 **RQMNAME** 属性的队列管理器别名会在需要时重新驱动工作负载均衡。

集群命名约定

此信息包含先前关于命名约定的指南以及当前指南。 随着 IBM MQ 技术的改进，以及客户以新的或不同的方式使用技术，必须针对这些场景提供新的建议和信息。

集群命名约定: 先前的指导

设置新集群时，请考虑队列管理器的命名约定。 每个队列管理器都必须具有不同的名称，但如果您为它们提供一组类似的名称，那么这可能有助于您记住哪些队列管理器分组在何处。

每个集群接收方通道还必须具有唯一名称。

如果有多个到同一队列管理器的通道，每个通道具有不同的优先级或使用不同的协议，那么可以扩展名称以包含不同的协议; 例如 QM1.S1， QM1.N3 和 QM1.T4。 在此示例中， S1 可能是第一个 SNA 通道， N3 可能是网络优先级为 3 的 NetBIOS 通道。

最终限定符可能描述通道提供的服务类。 有关更多信息，请参阅 定义服务类。

请记住，所有集群发送方通道都具有与其对应的集群接收方通道相同的名称。

请勿在集群接收方定义上使用通用连接名称。 在 IBM MQ for z/OS 中，可以定义 VTAM 通用资源或 *Dynamic Domain Name Server* (DDNS) 通用名称，但如果您正在使用集群，请勿执行此操作。 如果使用通用 **CONNAME** 定义 CLUSRCVR，那么不保证 CLUSSDR 通道指向您想要的队列管理器。 初始 CLUSSDR 可能最终指向队列共享组中的任何队列管理器，而不一定是托管完整存储库的队列管理器。 此外，如果通道进入重试状态，那么它可能会重新连接到具有相同通用名称的其他队列管理器，并且消息流会中断。

集群命名约定: 当前指导

此部分中的先前指南 第 37 页的『集群命名约定: 先前的指导』仍然有效。 但是，以下指南旨在作为设计新集群时的更新。 此更新建议可确保跨多个集群的通道的唯一性，从而允许多个集群成功重叠。 由于队列管理器和集群的名称最多可以包含 48 个字符，并且通道名称限制为 20 个字符，因此从头开始对对象进行命名时必须小心，以避免在项目中途更改命名约定。

设置新集群时，请考虑队列管理器的命名约定。 每个队列管理器都必须具有不同的名称。 如果为集群中的队列管理器提供一组类似的名称，那么这可能有助于您记住将哪些队列管理器分组到何处。

在定义通道时，请记住，集群中任何队列管理器上自动创建的所有集群发送方通道都具有与集群中接收队列管理器上配置的相应集群接收方通道相同的名称，因此必须唯一且在集群中对该集群的管理员有意义。 通道名称限制为最多 20 个字符。

一种可能是使用以集群名称开头的队列管理器名称。 例如，如果 cluster-name 为 CLUSTER1，并且队列管理器为 QM1， QM2，那么集群接收方通道为 CLUSTER1.QM1， CLUSTER1.QM2。

如果通道具有不同的优先级或使用不同的协议 (例如 CLUSTER1.QM1.S1， CLUSTER1.QM1.N3 和 CLUSTER1.QM1.T4)，那么可以扩展此约定。 在此示例中， S1 可能是第一个 SNA 通道， N3 可能是网络优先级为 3 的 NetBIOS 通道。

最终限定符可能描述通道提供的服务类。

IBM MQ for z/OS 注意事项

z/OS

在 IBM MQ for z/OS 中，可以定义 VTAM 通用资源或 *Dynamic Domain Name Server* (DDNS) 通用名称。 您可以使用通用名称来定义连接名称。 但是，创建集群接收方定义时，请勿使用通用连接名称。

对集群接收方定义使用通用连接名称的问题如下所示。 如果使用通用 CONNAME 定义 CLUSRCVR，那么不保证 CLUSSDR 通道指向您想要的队列管理器。 初始 CLUSSDR 可能最终指向队列共享组中的任何队列管理器，而不一定是托管完整存储库的队列管理器。 如果通道再次开始尝试连接，那么它可能会重新连接到具有相同通用名称的其他队列管理器，从而中断消息流。

集群: 拓扑设计注意事项

本主题提供有关规划和管理 IBM MQ 集群的指导。 此信息是基于测试和客户反馈的指南。

通过提前思考用户应用程序和内部管理流程的位置，可以避免许多问题，或者在稍后的日期将其最小化。 本主题包含有关设计决策的信息，这些决策可以提高性能，并随着集群的扩展而简化维护任务。

- 第 39 页的『集群基础结构的性能』
- 第 39 页的『完整存储库』
- 第 40 页的『应用程序是否应该在完整存储库上使用队列?』
- 第 40 页的『管理通道定义』
- 第 40 页的『多个通道上的工作负载均衡』

集群基础结构的性能

当应用程序尝试在集群中的队列管理器上打开队列时，队列管理器会向该队列的完整存储库注册其兴趣，以便它可以了解该队列在集群中的位置。 队列位置或配置的任何更新都将由完整存储库自动发送到相关队列管理器。 此相关注册在内部称为预订 (这些预订与 IBM MQ 中用于发布/预订消息传递的 IBM MQ 预订不同)

有关集群的所有信息都将通过每个完整存储库。 因此，完整存储库始终在集群中用于管理消息流量。 在管理这些预订时，系统资源的高使用率，以及这些预订的传输和生成的配置消息，都可能对集群基础结构造成相当大的负载。 在确保尽可能了解并最小化此负载时，需要考虑许多事项:

- 使用集群队列的单个队列管理器越多，系统中的预订就越多，因此发生更改时的管理开销越大，需要通知感兴趣的订户，尤其是在完整存储库队列管理器上。 将不必要的流量和完整存储库负载最小化的一种方法是将类似的应用程序 (即，那些使用相同队列的应用程序) 连接到较少数量的队列管理器。
- 除了系统中影响性能的预订数之外，集群对象配置的更改率也会影响性能，例如频繁更改集群队列配置。
- 当队列管理器是多个集群的成员 (即，它是重叠集群系统的一部分) 时，队列中的任何兴趣都会导致对它是其成员的每个集群进行预订，即使相同的队列管理器是多个集群的完整存储库也是如此。 此安排会增加系统上的负载，并且是考虑是否需要多个重叠集群 (而不是单个集群) 的一个原因。
- 应用程序消息流量 (即， IBM MQ 应用程序发送到集群队列的消息) 不会通过完整存储库到达目标队列管理器。 此消息流量直接在消息进入集群的队列管理器与集群队列所在的队列管理器之间发送。 因此，不必适应与完整存储库队列管理器相关的应用程序消息流量的高速率，除非完整存储库队列管理器恰好是提及的这两个队列管理器中的任一个。 因此，建议不要将完整存储库队列管理器用于集群基础结构负载较大的集群中的应用程序消息流量。

完整存储库

存储库是有关作为集群成员的队列管理器的信息集合。 托管集群中每个队列管理器的完整信息集的队列管理器具有完整的存储库。 有关完整存储库和部分存储库的更多信息，请参阅 集群存储库。

完整存储库必须保存在可靠且尽可能高可用性的服务器上，并且必须避免单点故障。 集群设计必须始终具有两个完整存储库。 如果完整存储库发生故障，那么集群仍可运行。

集群中的队列管理器对集群资源进行的任何更新的详细信息; 例如，集群队列从该队列管理器发送到该集群中最多两个完整存储库 (如果集群中只有一个完整存储库队列管理器，那么发送到一个完整存储库)。 这些完整存储库保存信息，并将其传播到集群中对其感兴趣的任何队列管理器 (即，他们预订该信息)。 要确保集群的每个成员都具有其中的集群资源的最新视图，每个队列管理器必须能够在任何时候与至少一个完整存储库队列管理器进行通信。

如果由于任何原因，队列管理器无法与任何完整存储库通信，那么它可以根据其已高速缓存的信息级别在一段时间内继续在集群中运行，但没有新的更新或对先前未使用的集群资源的访问权。

因此，您必须始终保持这两个完整存储库可用。 但是，这一安排并不意味着必须采取极端措施，因为集群在没有完整存储库的短时间内充分运作。

还有一个原因是集群必须具有两个完整存储库队列管理器 (集群信息的可用性除外): 此原因是为了确保保存在完整存储库高速缓存中的集群信息存在于两个位置以进行恢复。 如果只有一个完整存储库，并且它丢失了有关集群的信息，那么需要对集群中的所有队列管理器进行手动干预才能使集群再次工作。 但是，如果有两个完整存储库，那么由于始终将信息发布到两个完整存储库并从两个完整存储库预订信息，因此可以尽最大努力恢复失败的完整存储库。

- 可以在两个完整存储库集群设计中对完整存储库队列管理器执行维护，而不影响该集群的用户: 该集群仅使用一个存储库继续运行，因此在可能的情况下，请关闭存储库，应用维护，然后一次备份一个存储库。 即使在第二个完整存储库上发生中断，运行中的应用程序也至少不受影响三天。
- 除非有充分的理由使用第三个存储库，例如出于地理原因使用地理上的本地完整存储库，否则请使用两个存储库设计。 具有三个完整存储库意味着您永远不知道当前正在使用的两个存储库，并且可能存在由多个工作负载管理参数之间的交互所导致的管理问题。 建议不要有两个以上的完整存储库。
- 如果仍需要更好的可用性，请考虑将完整存储库队列管理器作为多实例队列管理器托管，或者使用特定于平台的高可用性支持来提高其可用性。
- 您必须将所有完整存储库队列管理器与手动定义的集群发送方通道完全互连。 当由于某种合理原因，集群具有两个以上完整存储库时，必须特别小心。 在这种情况下往往可能错过一个或多个渠道，为的是不至于

立竿见影。 当完全互联不发生时，往往会出现难以诊断的问题。 它们很难诊断，因为某些完整存储库未保存所有存储库数据，因此导致集群中的队列管理器具有不同的集群视图，具体取决于它们所连接的完整存储库。

应用程序是否应该在完整存储库上使用队列?

完整存储库在大多数方面与任何其他队列管理器完全相同，因此可以在完整存储库上托管应用程序队列，并将应用程序直接连接到这些队列管理器。 应用程序是否应该在完整存储库上使用队列?

普遍接受的答案是 "否?"。 虽然可以进行此配置，但许多客户希望保留这些队列管理器以专用于维护完整的存储库集群高速缓存。 此处描述了在决定任一选项时要考虑的点，但最终集群体系结构必须适合环境的特定需求。

- 升级: 通常，为了在 IBM MQ 的新发行版中使用新的集群功能，必须首先升级该集群的完整存储库队列管理器。 当集群中的应用程序想要使用新功能时，能够更新完整存储库 (以及部分存储库的部分子集) 而无需测试多个共存的应用程序可能很有用。
- 维护: 以类似方式，如果必须对完整存储库应用紧急维护，那么可以使用 **REFRESH** 命令重新启动或刷新这些存储库，而无需接触应用程序。
- 性能: 随着集群的增长以及对完整存储库集群高速缓存维护的需求越来越大，使应用程序保持独立会降低通过争用系统资源而影响应用程序性能的风险。
- 硬件需求: 通常，完整存储库不需要强大功能; 例如，具有良好可用性期望的简单 UNIX 服务器就足够了。 或者，对于非常大或不断变化的集群，必须考虑完整存储库计算机的性能。
- 软件需求: 需求通常是选择在完整存储库上托管应用程序队列的主要原因。 在小型集群中，并置可能意味着需要减少所有队列管理器/服务器的数量。

管理通道定义

即使在单个集群中，也可以存在多个通道定义，从而在两个队列管理器之间提供多个路由。

有时在单个集群中具有并行通道的优势，但必须彻底考虑此设计决策; 除了增加复杂性外，此设计可能导致通道使用不足，从而降低性能。 发生此情况是因为测试通常涉及以恒定速率发送大量消息，因此完全使用并行通道。 但在非恒定消息流的现实世界条件下，随着消息流从通道切换到通道，工作负载均衡算法会导致性能下降。

当队列管理器是多个集群的成员时，存在将单个通道定义与集群名称列表配合使用的选项，而不是为每个集群定义单独的 CLUSRCVR 通道。 但是，此设置可能会在以后导致管理困难; 例如，考虑将 TLS 应用于一个集群而不是另一个集群的情况。 因此，最好创建单独的定义，并且 第 29 页的『集群命名约定』 中建议的命名约定支持此操作。

多个通道上的工作负载均衡

此信息旨在作为对主题的高级理解。 有关此主题的基本说明 (在使用此处的信息之前必须了解) ，请参阅 使用集群进行工作负载管理， 集群中的工作负载均衡和 集群工作负载管理算法。

集群工作负载管理算法提供了一组庞大的工具，但如果不充分了解它们的工作和交互方式，它们就不能全部相互配合使用。 对于工作负载均衡过程，通道的重要性可能并不是很明显: 工作负载管理循环法算法的行为就像拥有集群队列的队列管理器的多个集群通道被视为该队列的多个实例一样。 以下示例更详细地说明了此过程:

1. 集群中有两个托管队列的队列管理器: QM1 和 QM2。
2. 有五个集群接收方通道到 QM1。
3. 只有一个集群接收方通道到 QM2。
4. 当 QM3 上的 **MQPUT** 或 **MQOPEN** 选择实例时，算法将消息发送到 QM1 的可能性比发送到 QM2 的可能性高五倍。
5. 发生步骤 4 中的情况是因为算法看到六个选项可供选择 (5 + 1) ，并在所有五个通道之间循环至 QM1 和单个通道至 QM2。

另一个微妙的行为是，即使将消息放入正好在本地队列管理器上配置了一个实例的集群队列， IBM MQ 也会使用本地集群接收方通道的状态来决定是将消息放入队列的本地实例还是队列的远程实例。 在此场景中:

1. 在放置消息时，工作负载管理算法不会查看个别集群队列，而是会查看可到达这些目标的集群通道。
2. 要到达本地目标，本地接收方通道将包括在此列表中 (尽管它们不用于发送消息)。
3. 当本地接收方通道停止时，工作负载管理算法在缺省情况下首选备用实例 (如果其 CLUSRCVR 未停止)。如果目标有多个本地 CLUSRCVR 实例，并且至少有一个实例未停止，那么本地实例仍符合条件。

集群: 使用多个集群传输队列进行应用程序隔离

您可以在集群中的队列管理器之间隔离消息流。 您可以将由不同集群发送方通道传输的消息放入不同的集群传输队列中。 您可以在单个集群中使用此方法，也可以将此方法与重叠的集群配合使用。 本主题提供了一些示例和一些最佳实践，以指导您选择使用方法。

部署应用程序时，您可以选择它与其他应用程序共享哪些 IBM MQ 资源以及它不共享哪些资源。 可以共享多种类型的资源，主要资源是服务器本身，队列管理器，通道和队列。 您可以选择配置共享资源较少的应用程序; 将单独的队列，通道，队列管理器甚至服务器分配给各个应用程序。 如果执行此操作，那么整体系统配置将变得更大且更复杂。 使用 IBM MQ 集群可降低管理更多服务器，队列管理器，队列和通道的复杂性，但它会引入另一个共享资源，即集群传输队列 SYSTEM.CLUSTER.TRANSMIT.QUEUE。

第 42 页的图 12 是大型 IBM MQ 部署中的一个切片，用于说明共享 SYSTEM.CLUSTER.TRANSMIT.QUEUE 的重要性。 在图中，应用程序 Client App 已连接到集群 CL1 中的队列管理器 QM2 。 来自 Client App 的消息由应用程序 Server App 处理。 此消息由 Server App 从 CLUSTER2 中队列管理器 QM3 上的集群队列 Q1 中检索。 由于客户机和服务器应用程序不在同一集群中，因此消息由网关队列管理器 QM1 传输。

配置集群网关的正常方法是使网关队列管理器成为所有集群的成员。 在网关队列管理器上，为所有集群中的集群队列定义了集群别名队列。 集群队列别名在所有集群中都可用。 放入集群队列别名的消息通过网关队列管理器路由到其正确目标。 网关队列管理器将发送到集群别名队列的消息放到 QM1 上的公共 SYSTEM.CLUSTER.TRANSMIT.QUEUE 上。

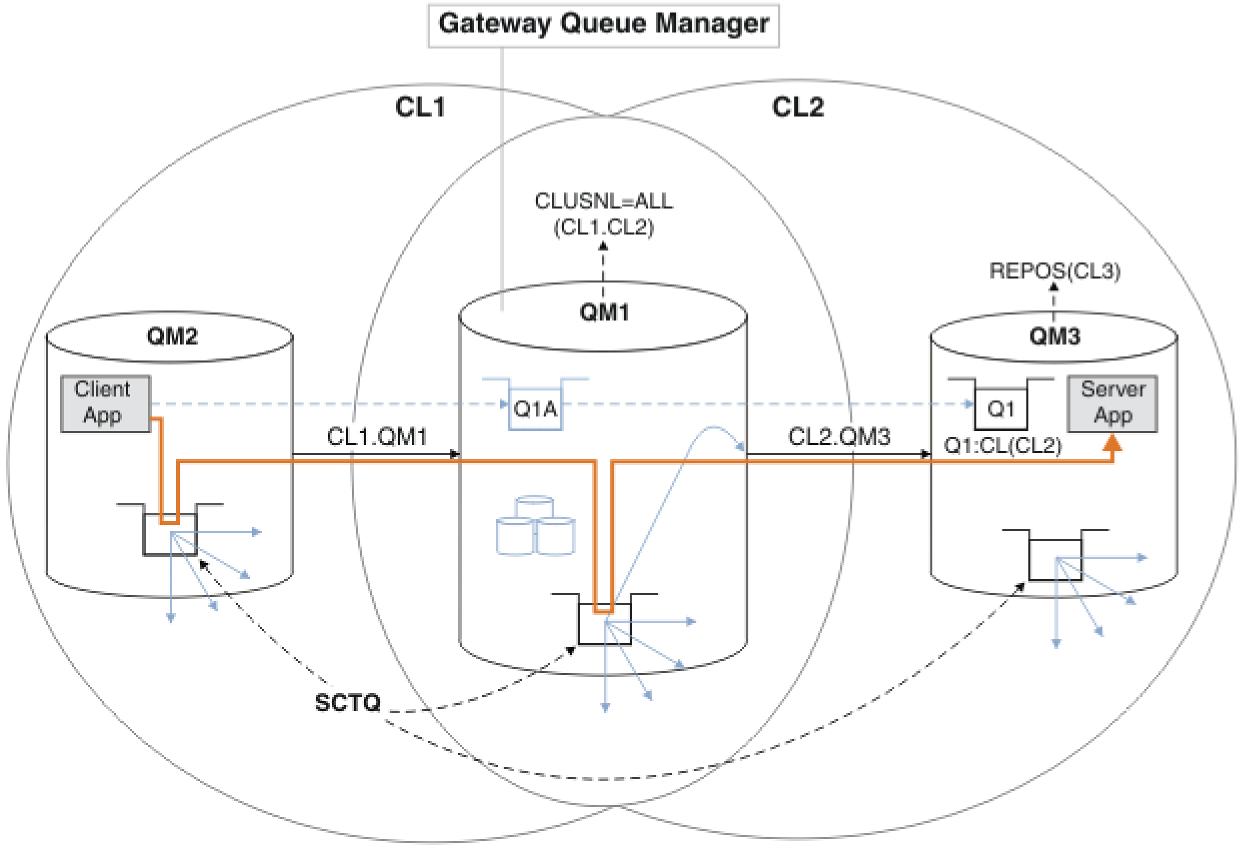
中心和辐射体系结构要求集群之间的所有消息都通过网关队列管理器。 结果是所有消息都流经 QM1 SYSTEM.CLUSTER.TRANSMIT.QUEUE 上的单集群传输队列。

从性能角度来看，单个队列不是问题。 公共传输队列通常不表示性能瓶颈。 网关上的消息吞吐量在很大程度上取决于与其连接的通道的性能。 吞吐量通常不受队列数或使用通道的队列上的消息数的影响。

从其他一些角度来看，将单个传输队列用于多个应用程序有缺点:

- 不能将消息流到一个目标与消息流到另一个目标隔离。 即使目标位于不同队列管理器上的不同集群中，也无法在转发消息之前将其存储分开。

  如果一个集群目标变为不可用，那么将在单个传输队列中构建该目标的消息，并最终填充该消息。 一旦传输队列已满，它就会停止将消息放置到任何集群目标的传输队列上。
- 监视消息到不同集群目标的传输并不容易。 所有消息都在单个传输队列上。 显示传输队列的深度几乎不会指示是否将消息传输到所有目标。



注: 第 42 页的图 12 和下图中的箭头具有不同类型。 实心箭头表示消息流。 实线箭头上的标签是消息通道名称。 灰色实线箭头是从 SYSTEM.CLUSTER.TRANSMIT.QUEUE 到集群发送方通道的潜在消息流。 黑色虚线将标签连接到其目标。 灰色虚线箭头是引用; 例如，从 MQOPEN 调用 Client App 到集群别名队列定义 Q1A。

图 *12:* 使用 *IBM MQ* 集群将客户机/服务器应用程序部署到中心和辐射体系结构

在 第 42 页的图 12 中， Server App 的客户机打开队列 Q1A。 消息将放置到 QM2 上的 SYSTEM.CLUSTER.TRANSMIT.QUEUE ，传输到 QM1 上的 SYSTEM.CLUSTER.TRANSMIT.QUEUE ，然后传输到 QM3 上的 Q1 ， Server App 应用程序将在此接收到这些消息。

来自 Client App 的消息通过 QM2 和 QM1 上的系统集群传输队列传递。 在 第 42 页的图 12 中，目标是将网关队列管理器上的消息流与客户机应用程序隔离，以便其消息不会存储在 SYSTEM.CLUSTER.TRANSMIT.QUEUE 上。 您可以隔离任何其他集群队列管理器上的流。 您还可以将其他方向的流隔离回客户机。 为了使解决方案的描述简短，这些描述仅考虑来自客户机应用程序的单个流。

用于隔离集群网关队列管理器上的集群消息流量的解决方案

解决问题的一种方法是使用队列管理器别名或远程队列定义在集群之间进行桥接。 创建集群远程队列定义，传输队列和通道，以分隔网关队列管理器上的每个消息流; 请参阅 添加远程队列定义以隔离从网关队列管理器发送的消息。

从 IBM WebSphere MQ 7.5 开始，集群队列管理器不限于单个集群传输队列。 您有两种选择:

1. 手动定义其他集群传输队列，并定义从每个传输队列传输消息的集群发送方通道; 请参阅 添加集群传输队列以隔离从网关队列管理器发送的集群消息流量。
2. 允许队列管理器自动创建和管理其他集群传输队列。 它为每个集群发送方通道定义不同的集群传输队列; 请参阅 更改缺省值以分隔集群传输队列以隔离消息流量。

您可以将某些集群发送方通道的手动定义的集群传输队列与队列管理器一起管理其余的集群传输队列。 传输队列的组合是在 添加集群传输队列以隔离从网关队列管理器发送的集群消息流量中采用的方法。 在该解决

方案中，集群之间的大多数消息都使用公共 SYSTEM.CLUSTER.TRANSMIT.QUEUE。 一个应用程序是关键的，它的所有消息流都通过使用一个手动定义的集群传输队列与其他流隔离。

添加集群传输队列以隔离从网关队列管理器发送的集群消息流量 中的配置受到限制。 它不会将消息流量与另一个集群队列分隔到同一集群中同一队列管理器上的集群队列。 您可以使用属于分布式排队的远程队列定义来分隔到各个队列的消息流量。 借助集群，通过使用多个集群传输队列，您可以将进入不同集群发送方通道的消息流量分开。 同一集群中的多个集群队列在同一队列管理器上共享一个集群发送方通道。 这些队列的消息存储在同一传输队列上，然后再从网关队列管理器转发。 在 添加集群和集群传输队列以隔离从网关队列管理器发送的集群消息流量中的配置中，通过添加另一个集群并使队列管理器和集群队列成为新集群的成员来避免此限制。 新的队列管理器可能是集群中唯一的队列管理器。 您可以向集群添加更多队列管理器，并使用同一集群来隔离这些队列管理器上的集群队列。

相关概念

第 25 页的『访问控制和多个集群传输队列』
在应用程序将消息放入远程集群队列时选择三种检查方式。 这些方式是针对集群队列进行远程检查，针对 SYSTEM.CLUSTER.TRANSMIT.QUEUE 进行本地检查，或者针对集群队列或集群队列管理器的本地概要文件进行检查。

使用集群传输队列和集群发送方通道

第 31 页的『重叠集群』
重叠的集群提供了额外的管理功能。 使用名称列表来减少管理重叠集群所需的命令数。

相关任务

授权将消息放入远程集群队列

添加远程队列定义以隔离从网关队列管理器发送的消息

添加集群传输队列以隔离从网关队列管理器发送的集群消息流量

添加集群和集群传输队列以隔离从网关队列管理器发送的集群消息流量

将缺省值更改为单独的集群传输队列以隔离消息流量

使用网关队列管理器创建两个重叠的集群

配置集群之间的消息路径

保护

相关参考

塞特 MQaut

集群: 规划如何配置集群传输队列

将指导您选择集群传输队列。 您可以配置一个公共缺省队列，单独的缺省队列或手动定义的队列。

开始之前

复审 第 45 页的『如何选择要使用的集群传输队列类型』。

关于此任务

当您计划如何配置队列管理器以选择集群传输队列时，可以选择一些选项。

1. 什么是集群消息传输的缺省集群传输队列?

   a. 公共集群传输队列 SYSTEM.CLUSTER.TRANSMIT.QUEUE。

   b. 单独的集群传输队列。 队列管理器管理单独的集群传输队列。 它会将它们创建为模型队列 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 中的永久动态队列。 它为它使用的每个集群发送方通道创建一个集群传输队列。

2. 对于您决定手动创建的集群传输队列，还有两个选项:

   a. 为您决定手动配置的每个集群发送方通道定义单独的传输队列。 在这种情况下，请将传输队列的 **CLCHNAME** 队列属性设置为集群发送方通道的名称。 选择要从此传输队列传输消息的集群发送方通道。

   b. 将一组集群发送方通道的消息流量组合到同一集群传输队列上; 请参阅 第 44 页的图 13。 在这种情况下，请将每个公共传输队列的 **CLCHNAME** 队列属性设置为通用集群发送方通道名称。 通用集群发送

方通道名称是用于对集群发送方通道名称进行分组的过滤器。 例如， SALES.* 对名称以 SALES.开头的所有集群发送方通道进行分组。 可以在 filter-string 中的任何位置放置多个通配符。 通配符为星号 “*”。 它表示从零到任意数目的字符。

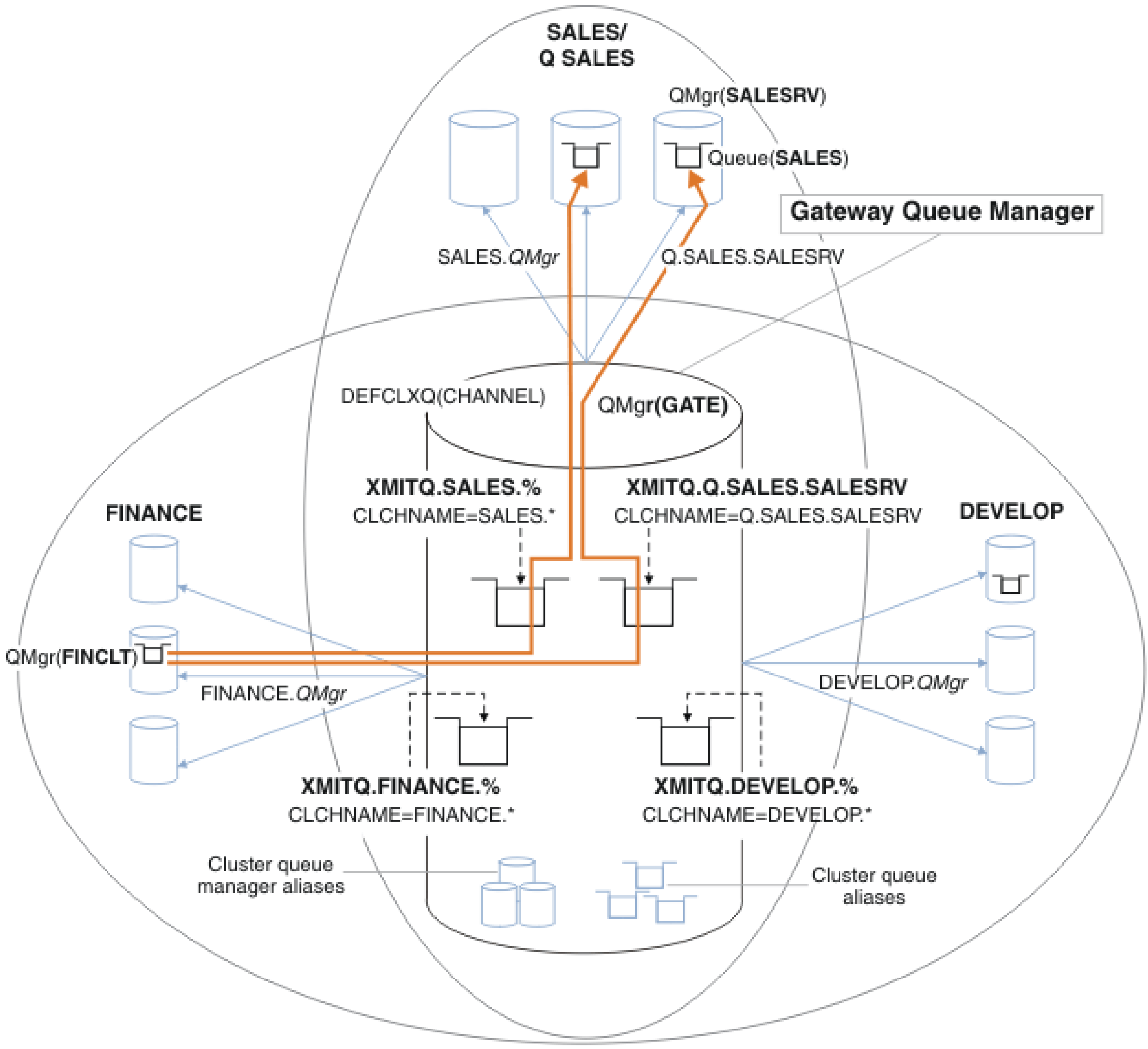


图 *13:* 不同部门 *IBM MQ* 集群的特定传输队列示例

过程

1. 选择要使用的缺省集群传输队列类型。
   - 选择单个集群传输队列，或者为每个集群连接选择单独的队列。

   保留缺省设置或运行 **MQSC** 命令:

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

2. 隔离不得与其他流共享集群传输队列的任何消息流。
   - 请参阅 第 47 页的『集群: 多个集群传输队列的示例配置』。 在此示例中，必须隔离的 SALES 队列是 SALESRV 上 SALES 集群的成员。 要隔离 SALES 队列，请创建新集群 Q.SALES，使 SALESRV 队列管理器成为成员，并修改 SALES 队列以属于 Q.SALES。
   - 向 SALES 发送消息的队列管理器也必须是新集群的成员。 如果使用集群队列别名和网关队列管理器，例如，在许多情况下，可以将更改限制为使网关队列管理器成为新集群的成员。

- 但是，将流从网关分离到目标不会将流从源队列管理器分离到网关。 但有时事实证明，这足以将流与网关分开，而不是流向网关。 如果不够，请将源队列管理器添加到新集群中。 如果希望消息通过网关传递，请将集群别名移动到新集群，并继续将消息发送到网关上的集群别名，而不是直接发送到目标队列管理器。

执行以下步骤以隔离消息流:

a) 配置流的目标，以便每个目标队列都是该队列管理器上特定集群中的唯一队列。

b) 为遵循系统命名约定创建的任何新集群创建集群发送方通道和集群接收方通道。

   - 请参阅 第 36 页的『集群: 重叠集群的特殊注意事项』。

c) 为每个将消息发送到目标队列的队列管理器上的每个隔离目标定义集群传输队列。

   - 集群传输队列的命名约定是使用以 XMITQ. 为前缀的集群通道名称属性 CLCHNAME 的值

3. 创建集群传输队列以满足监管或监视需求。

   - 典型的监管和监视需求会导致每个集群的传输队列或每个队列管理器的传输队列。 如果遵循集群通道 *ClusterName. QueueManagerName* 的命名约定，那么可以轻松创建用于选择队列管理器集群或队列管理器所属的所有集群的通用通道名称; 请参阅 第 47 页的『集群: 多个集群传输队列的示例配置』。
   - 通过将星号替换为百分号，扩展集群传输队列的命名约定以迎合通用通道名称。 例如，

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

相关概念

使用集群传输队列和集群发送方通道

第 25 页的『访问控制和多个集群传输队列』
在应用程序将消息放入远程集群队列时选择三种检查方式。 这些方式是针对集群队列进行远程检查，针对 SYSTEM.CLUSTER.TRANSMIT.QUEUE 进行本地检查，或者针对集群队列或集群队列管理器的本地概要文件进行检查。

第 31 页的『重叠集群』
重叠的集群提供了额外的管理功能。 使用名称列表来减少管理重叠集群所需的命令数。

相关任务

添加远程队列定义以隔离从网关队列管理器发送的消息

添加集群传输队列以隔离从网关队列管理器发送的集群消息流量

添加集群和集群传输队列以隔离从网关队列管理器发送的集群消息流量

将缺省值更改为单独的集群传输队列以隔离消息流量

使用网关队列管理器创建两个重叠的集群

配置集群之间的消息路径

如何选择要使用的集群传输队列类型

如何在不同的集群传输队列配置选项之间进行选择。

从 IBM WebSphere MQ 7.5 开始，您可以选择哪个集群传输队列与集群发送方通道相关联。

1. 您可以使所有集群发送方通道与单个缺省集群传输队列 SYSTEM.CLUSTER.TRANSMIT.QUEUE 相关联。 此选项是缺省值，并且是运行 IBM WebSphere MQ 7.1 或更低版本的队列管理器的唯一选项。
2. 您可以将所有集群发送方通道设置为自动与单独的集群传输队列相关联。 队列由队列管理器从模型队列 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 创建，命名为 SYSTEM.CLUSTER.TRANSMIT.*ChannelName*。 如果队列管理器属性 **DEFCLXQ** 设置为 CHANNEL，那么通道将使用其唯一命名的集群传输队列。

   注意: 如果要将专用 SYSTEM.CLUSTER.TRANSMIT.QUEUES 与从低于 IBM WebSphere MQ 7.5 的产品版本升级的队列管理器配合使用，请确保 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 将 SHARE/NOSHARE 选项设置为 **SHARE**。

3. 您可以设置要由单个集群传输队列提供服务的特定集群发送方通道。 通过创建传输队列并将其 **CLCHNAME** 属性设置为集群发送方通道的名称来选择此选项。

4. 您可以选择要由单个集群传输队列提供服务的集群发送方通道组。 通过创建传输队列并将其 **CLCHNAME** 属性设置为通用通道名称 (例如 *ClusterName.**) 来选择此选项。 如果通过遵循 第 36 页的『集群: 重叠集群的特殊注意事项』中的命名约定来命名集群通道，那么此名称将选择连接到集群 *ClusterName* 中的队列管理器的所有集群通道。

您可以将某些集群发送方通道的任一缺省集群传输队列选项与任意数量的特定和通用集群传输队列配置组合在一起。

最佳实践

在大多数情况下，对于现有 IBM MQ 安装，缺省配置是最佳选择。 集群队列管理器将集群消息存储在单个集群传输队列 SYSTEM.CLUSTER.TRANSMIT.QUEUE 上。 您可以选择将缺省值更改为将不同队列管理器和不同集群的消息存储在不同的传输队列上，或者定义您自己的传输队列。

在大多数情况下，对于新的 IBM MQ 安装，缺省配置也是最佳选择。 从缺省配置切换到针对每个集群发送方通道具有一个传输队列的备用缺省值的过程是自动的。 切换回也是自动的。 一个或另一个的选择并不关键，你可以逆转它。

选择其他配置的原因更多是与监管和管理有关，而不是与功能或性能有关。 如果有几个异常，那么配置多个集群传输队列对队列管理器的行为没有好处。 这将生成更多队列，并要求您修改已设置的监视和管理过程，这些过程将引用单个传输队列。 这就是为什么在平衡上，保留缺省配置是最佳选择，除非您有强大的治理或管理原因进行不同的选择。

这些异常都与 SYSTEM.CLUSTER.TRANSMIT.QUEUE 上存储的消息数增加时发生的情况相关。 如果您执行每个步骤来将一个目标的消息与另一个目标的消息分开，那么一个目标的通道和传递问题不应影响到另一个目标的传递。 但是，由于无法以足够快的速度将消息传递到一个目标，因此存储在 SYSTEM.CLUSTER.TRANSMIT.QUEUE 上的消息数可能会增加。 SYSTEM.CLUSTER.TRANSMIT.QUEUE 上针对一个目标的消息数可能会影响将消息传递到其他目标。

要避免由于填充单个传输队列而导致的问题，请将足够的容量构建到配置中。 然后，如果目标失败并且消息积压开始构建，那么您有时间来解决该问题。

如果通过中心队列管理器 (例如集群网关) 路由消息，那么它们共享公共传输队列 SYSTEM.CLUSTER.TRANSMIT.QUEUE。 如果网关队列管理器上存储在 SYSTEM.CLUSTER.TRANSMIT.QUEUE 上的消息数达到其最大深度，那么队列管理器将开始拒绝传输队列的新消息，直到其深度减小为止。 拥塞会影响通过网关路由的所有目标的消息。 消息备份正在向网关发送消息的其他队列管理器的传输队列。 该问题表现在写入队列管理器错误日志的消息中，消息吞吐量下降，以及从发送消息到消息到达其目标的时间之间的耗用时间较长。

拥堵对单个传输队列的影响会变得明显，即使在它已满之前也是如此。 如果您有混合的消息流量，有一些大的非持久消息和一些小的消息，那么随着传输队列的填满，传递小消息的时间会增加。 延迟是由于将通常不会写入磁盘的大型非持久消息写入磁盘。 如果您有时间关键消息流，与其他混合消息流共享集群传输队列，那么可能需要配置特殊消息路径以将其与其他消息流隔离; 请参阅 添加集群和集群传输队列以隔离从网关队列管理器发送的集群消息流量。

配置单独的集群传输队列的其他原因是为了满足监管要求，或者为了简化发送到不同集群目标的监视消息。 例如，您可能必须证明一个目标的消息从不与另一个目标的消息共享传输队列。

更改用于控制缺省集群传输队列的队列管理器属性 **DEFCLXQ**，以便为每个集群发送方通道创建不同的集群传输队列。 多个目标可以共享一个集群发送方通道，因此您必须规划集群以完全满足此目标。 将方法 添加集群和集群传输队列以隔离从网关队列管理器发送的集群消息流量 系统地应用于所有集群队列。 您的目标结果是没有集群目标与另一个集群目标共享集群发送方通道。 因此，集群目标的任何消息都不会与另一个目标的消息共享其集群传输队列。

为某些特定消息流创建单独的集群传输队列，可以轻松监视到该目标的消息流。 要使用新的集群传输队列，请定义该队列，将其与集群发送方通道相关联，然后停止并启动该通道。 此更改不必是永久性的。 您可以将消息流隔离一段时间，以监视传输队列，然后再次还原为使用缺省传输队列。

相关任务

集群: 多个集群传输队列的示例配置
在此任务中，您将应用步骤以将多个集群传输队列规划到三个重叠的集群。 这些要求是将消息流与所有其他消息流分离到一个集群队列，并将不同集群的消息存储在不同的集群传输队列上。

集群: 切换集群传输队列
规划如何使对现有生产队列管理器的集群传输队列所作的更改生效。

集群: 多个集群传输队列的示例配置
在此任务中，您将应用步骤以将多个集群传输队列规划到三个重叠的集群。 这些要求是将消息流与所有其他消息流分离到一个集群队列，并将不同集群的消息存储在不同的集群传输队列上。

关于此任务

此任务中的步骤显示如何应用 第 43 页的『集群: 规划如何配置集群传输队列』 中的过程并得出 第 47 页的图 14 中显示的配置。 它是具有网关队列管理器的三个重叠集群的示例，该集群配置有单独的集群传输队列。 第 49 页的『创建示例集群』中描述了用于定义集群的 MQSC 命令。

对于此示例，有两个需求。 一是将消息流从网关队列管理器分离到记录销售的销售应用程序。 二是查询有多少消息等待在任何时间点发送到不同的部门地区。 已定义 SALES， FINANCE 和 DEVELOP 集群。 集群消息当前从 SYSTEM.CLUSTER.TRANSMIT.QUEUE 转发。

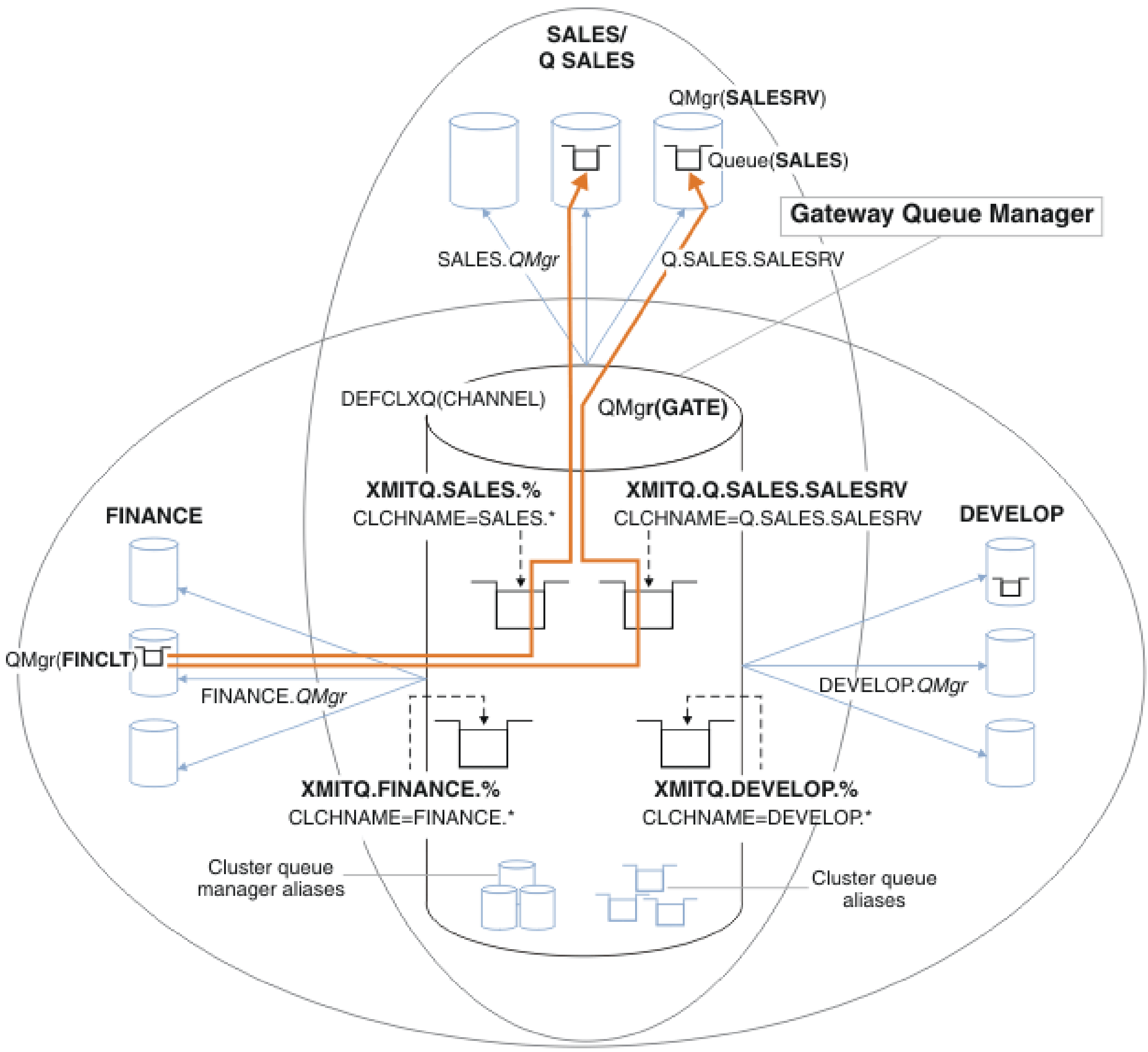


图 *14:* 不同部门 *IBM MQ* 集群的特定传输队列示例

修改集群的步骤如下所示: 请参阅 更改以隔离新集群中的销售队列并分隔网关集群传输队列 以获取定义。

过程

1. 第一个配置步骤是 “ 选择要使用的缺省集群传输队列类型 ”。

   决定是通过在 GATE 队列管理器上运行以下 **MQSC** 命令来创建单独的缺省集群传输队列。

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

   没有强烈的理由选择此缺省值，因为目的是手动定义集群传输队列。 该选项确实具有弱诊断值。 如果手动定义错误，并且消息在缺省集群传输队列中流下，那么会在创建永久动态集群传输队列时显示该消息。

2. 第二个配置步骤是 “ 隔离不得与其他流共享集群传输队列的任何消息流 ”。

   在这种情况下，从 SALESRV 上的队列 SALES 接收消息的销售应用程序需要隔离。 仅需要隔离来自网关队列管理器的消息。 三个分步骤是:

   a) “ 配置流的目标， 以便每个目标队列都是该队列管理器上特定集群中的唯一队列 ”.

      此示例需要将队列管理器 SALESRV 添加到销售部门中的新集群。 如果您有几个需要隔离的队列，那么可以决定为 SALES 队列创建特定集群。 集群名称的可能命名约定是将此类集群命名为 Q. *QueueName*，例如 Q.SALES。 如果要隔离大量队列，那么可能更实际的替代方法是在需要时创建隔离队列的集群。 集群名称可能为 QUEUES. *n*。

      在此示例中，新集群称为 Q.SALES。 要添加新集群，请参阅 更改以隔离新集群中的销售队列并分隔网关集群传输队列中的定义。 定义更改的摘要如下:

      i) 将 Q.SALES 添加到存储库队列管理器上的集群的名称列表。 在队列管理器 **REPOSNL** 参数中引用了名称列表。

      ii) 将 Q.SALES 添加到网关队列管理器上的集群的名称列表。 在网关队列管理器上的所有集群队列别名和集群队列管理器别名定义中都引用了名称列表。

      iii) 在队列管理器 SALESRV 上为其所属的两个集群创建名称列表，并更改 SALES 队列的集群成员资格:

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

      SALES 队列是两个集群的成员，仅用于转换。 新配置运行后，将从 SALES 集群中除去 SALES 队列; 请参阅 第 52 页的图 15。

   b) “ 为遵循系统命名约定创建的任何新集群创建集群发送方通道和集群接收方通道 ”.

      i) 将集群接收方通道 Q.SALES. *RepositoryQMgr* 添加到每个存储库队列管理器

      ii) 将集群发送方通道 Q.SALES. *OtherRepositoryQMgr* 添加到每个存储库队列管理器，以连接到其他存储库管理器。 启动这些通道。

      iii) 将集群接收方通道 Q.SALES.SALESRV 和 Q.SALES.GATE 添加到正在运行的任一存储库队列管理器。

      iv) 将集群发送方通道 Q.SALES.SALESRV 和 Q.SALES.GATE 添加到 SALESRV 和 GATE 队列管理器。 将集群发送方通道连接到您在其中创建集群接收方通道的存储库队列管理器。

   c) “ 为每个将消息发送到目标队列的队列管理器上的每个隔离目标定义集群传输队列 ”.

      在网关队列管理器上，为 Q.SALES.SALESRV 集群发送方通道定义集群传输队列 XMITQ.Q.SALES.SALESRV：

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

3. 第三个配置步骤是 “ 创建集群传输队列以满足监管或监视需求 ”。

   在网关队列管理器上定义集群传输队列:

   ```
   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
```

下一步做什么

切换到网关队列管理器上的新配置。

通过启动新通道并重新启动现在与不同传输队列相关联的通道来触发交换机。 或者，可以停止并启动网关队列管理器。

1. 停止网关队列管理器上的以下通道:

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

2. 在网关队列管理器上启动以下通道:

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

交换机完成后，从 SALES 集群中除去 SALES 队列; 请参阅 第 52 页的图 15。

相关概念

如何选择要使用的集群传输队列类型
如何在不同的集群传输队列配置选项之间进行选择。

相关任务

集群: 切换集群传输队列
规划如何使对现有生产队列管理器的集群传输队列所作的更改生效。

创建示例集群
用于创建示例集群并对其进行修改以隔离 SALES 队列和网关队列管理器上的单独消息的定义和指示信息。

关于此任务

用于创建 FINANCE， SALES 和 Q.SALES 集群的完整 **MQSC** 命令在 基本集群的定义， 用于隔离新集群中的销售队列并分隔网关集群传输队列的更改和 从销售集群中除去队列管理器 SALESRV 上的销售队列中提供。 定义中省略了 DEVELOP 集群，以保持定义较短。

过程

1. 创建 SALES 和 FINANCE 集群以及网关队列管理器。
   a) 创建队列管理器。

      对 第 49 页的表 4 中的每个队列管理器名称运行命令 `crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE` *QmgrName* 。

表 4: 队列管理器名称和端口号

描述	队列管理器名称	端口号
财务存储库	FINR1	1414
财务存储库	FINR2	1415
财务客户	FINCLT	1418
销售存储库	SALER1	1416
销售存储库	SALER2	1417

表 4: 队列管理器名称和端口号 (继续)		
描述	**队列管理器名称**	**端口号**
销售服务器	SALESRV	1419
网关	GATE	1420

b) 启动所有队列管理器

对 第 49 页的表 4 中的每个队列管理器名称运行命令 `strmqm` *QmgrName*。

c) 为每个队列管理器创建定义

Run the command: `runmqsc` *QmgrName* < *filename* where the files are listed in 基本集群的定义, and the file name matches the queue manager name.

基本集群的定义

finr1.txt

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

finr2.txt

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

finclt.txt

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

saler1.txt

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

saler2.txt

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

salesrv.txt

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

gate.txt

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

2. 通过运行样本请求程序来测试配置。
   a) 在 SALESRV 队列管理器上启动触发器监视器程序

      在 Windows 上，打开命令窗口并运行命令 `runmqtrm -m SALESRV`
   b) 运行样本请求程序，然后发送请求。

      在 Windows 上，打开命令窗口并运行命令 `amqsreq A.SALES FINCLT`

      将回传请求消息，并在 15 秒后完成样本程序。
3. 创建定义以隔离 `Q.SALES` 集群中的 SALES 队列，并在网关队列管理器上为 SALES 和 `FINANCE` 集群单独创建集群消息。

   Run the command: `runmqsc` *QmgrName* < *filename* where the files are listed in the following list, and the file name almost matches the queue manager name.

   用于隔离新集群中的销售队列并分隔网关集群传输队列的更改

   chgsaler1.txt

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

   chgsaler2.txt

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

   chgsalesrv.txt

```
DEFINE NAMELIST (CLUSTERS) NAMES(SALES, Q.SALES)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
```

```
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SAVESRV) CHLTYPE(CLUSRCVR) CONNAME('localhost(1419)')
CLUSTER(Q.SALES) REPLACE
ALTER QLOCAL (SALES) CLUSTER(' ') CLUSNL(CLUSTERS)
```

chggate.txt

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

4. 从 SALES 集群中除去 SALES 队列。

   在 第 52 页的图 15 中运行 **MQSC** 命令:

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

   图 *15:* 从销售集群中除去队列管理器 *SALESRV* 上的销售队列

5. 将通道切换到新的传输队列。

   需要停止并启动 GATE 队列管理器正在使用的所有通道。 要使用最少数量的命令执行此操作，请停止并启动队列管理器

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

下一步做什么

1. 重新运行样本请求程序以验证新配置是否有效; 请参阅步骤 第 51 页的『2』
2. 监视流经 GATE 队列管理器上所有集群传输队列的消息:
   a. 更改要开启队列监视的每个集群传输队列的定义。

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

   b. 检查队列管理器统计信息监视是否为 OFF，以最小化输出，并将监视时间间隔设置为较低的值以方便地执行多个测试。

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

   c. 重新启动 GATE 队列管理器。
   d. 运行样本请求程序几次以验证是否有相同数量的消息流经 SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV 和 SYSTEM.CLUSTER.TRANSMIT.QUEUE。 请求流经 SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV ，应答流经 SYSTEM.CLUSTER.TRANSMIT.QUEUE。

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

   e. 几个时间间隔内的结果如下所示:

```
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.
```

在第一个时间间隔内发送了一个请求和应答消息，在第二个时间间隔内发送了两个请求和应答消息。您可以推断请求消息已放置在 SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV 上，应答消息已放置在 SYSTEM.CLUSTER.TRANSMIT.QUEUE 上。

集群: 切换集群传输队列
规划如何使对现有生产队列管理器的集群传输队列所作的更改生效。

开始之前

如果减少切换过程必须传输到新传输队列的消息数，那么切换将更快地完成。 请阅读 将集群发送方通道切换到其他传输队列的过程如何工作，以了解在继续操作之前尝试清空传输队列的原因。

关于此任务

您可以选择两种方法使对集群传输队列的更改生效。

1. 让队列管理器自动进行更改。 这是缺省值。 当集群发送方通道下次启动时，队列管理器会切换具有暂挂传输队列更改的集群发送方通道。
2. 手动进行更改。 您可以在集群发送方通道停止时对其进行更改。 在集群发送方通道启动之前，可以将其从一个集群传输队列切换到另一个集群传输队列。

在决定选择两个选项中的哪一个时，您会考虑哪些因素，以及如何管理交换机?

过程

- 选项 1: 让队列管理器自动进行更改; 请参阅 第 55 页的『将活动集群发送方通道切换到另一组集群传输队列』。

  如果希望队列管理器为您进行切换，请选择此选项。

  描述此选项的另一种方法是，队列管理器在不强制通道停止的情况下切换集群发送方通道。 您可以选择强制通道停止，然后启动通道，以使交换机更快发生。 交换机在通道启动时启动，并在通道运行时运行，这与选项 2 不同。 在选项 2 中，交换机在通道停止时发生。

  如果通过让交换机自动发生来选择此选项，那么切换过程将在集群发送方通道启动时启动。 如果通道未停止，那么它将在变为不活动状态后启动 (如果有要处理的消息)。 如果通道已停止，请使用 START CHANNEL 命令将其启动。
  一旦在通道所服务的传输队列上没有针对集群发送方通道的剩余消息，交换机进程就会完成。 一旦出现这种情况，集群发送方通道的新到达消息将直接存储在新的传输队列上。 在此之前，消息存储在旧的传输队列上，切换过程将消息从旧的传输队列传输到新的传输队列。 集群发送方通道在整个切换过程中转发来自新集群传输队列的消息。
  交换机进程何时完成取决于系统的状态。 如果要在维护窗口中进行更改，请事先评估切换过程是否会及时完成。 它是否会及时完成取决于等待从旧传输队列传输的消息数是否达到零。

  第一种方法的优点是它是自动的。 一个缺点是，如果进行配置更改的时间仅限于维护窗口，那么您必须确信可以控制系统在维护窗口内完成切换过程。 如果无法确定，那么选项 2 可能是更好的选择。
- 选项 2: 手动进行更改; 请参阅 第 56 页的『将已停止的集群发送方通道切换到另一个集群传输队列』。

  如果要手动控制整个切换过程，或者要切换已停止或不活动的通道，请选择此选项。 如果要切换几个集群发送方通道，并且要在维护窗口期间执行切换，那么这是一个不错的选择。

  此选项的替代描述是表示在集群发送方通道停止时切换集群发送方通道。

如果选择此选项，那么您可以完全控制交换机的发生时间。

您可以确定在固定时间内，在维护窗口内完成切换过程。 切换所花费的时间取决于必须从一个传输队列传输到另一个传输队列的消息数。 如果消息不断到达，那么进程可能需要一段时间才能传输所有消息。

您可以选择在不从旧传输队列传输消息的情况下切换通道。 交换机为 “即时”。

当您重新启动集群发送方通道时，它将开始处理新分配给它的传输队列上的消息。

第二种方法的优点是您可以控制切换过程。 缺点是您必须识别要切换的集群发送方通道，运行必要的命令，并解决可能阻止集群发送方通道停止的任何不确定通道。

相关概念

如何选择要使用的集群传输队列类型
如何在不同的集群传输队列配置选项之间进行选择。

将集群发送方通道切换到其他传输队列的过程如何工作

相关任务

集群: 多个集群传输队列的示例配置
在此任务中，您将应用步骤以将多个集群传输队列规划到三个重叠的集群。 这些要求是将消息流与所有其他消息流分离到一个集群队列，并将不同集群的消息存储在不同的集群传输队列上。

将活动集群发送方通道切换到另一组集群传输队列
此任务为您提供了三个用于切换活动集群发送方通道的选项。 一个选项是让队列管理器自动进行切换，这不会影响正在运行的应用程序。 其他选项包括手动停止和启动通道，或者重新启动队列管理器。

开始之前

更改集群传输队列配置。 您可以更改 **DEFCLXQ** 队列管理器属性，或者添加或修改传输队列的 **CLCHNAME** 属性。

如果减少切换过程必须传输到新传输队列的消息数，那么切换将更快地完成。 请阅读 将集群发送方通道切换到其他传输队列的过程如何工作 ，以了解在继续操作之前尝试清空传输队列的原因。

关于此任务

使用任务中的步骤作为制定您自己的计划以进行集群传输队列配置更改的基础。

过程

1. 可选： 记录当前通道状态

   记录正在为集群传输队列提供服务的当前通道和已保存通道的状态。 以下命令显示与系统集群传输队列关联的状态。 添加您自己的命令以显示与已定义的集群传输队列相关联的状态。 使用约定 (例如 XMITQ.*ChannelName*) 来命名您定义的集群传输队列，以便于显示这些传输队列的通道状态。

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

2. 交换机传输队列。

   - 不执行任何操作。 当集群发送方通道在停止或不活动后重新启动时，队列管理器会切换这些集群发送方通道。

     如果您没有更改队列管理器配置的规则或顾虑，请选择此选项。 正在运行的应用程序不受更改影响。

   - 重新启动队列管理器。 将根据需要停止并自动重新启动所有集群发送方通道。

     选择此选项以立即启动所有更改。 在队列管理器关闭并重新启动时，正在运行的应用程序将被队列管理器中断。

   - 停止各个集群发送方通道并将其重新启动。

     选择此选项可立即更改几个通道。 正在运行的应用程序在停止消息通道和再次启动消息通道之间的消息传输中迂到短暂延迟。 集群发送方通道保持运行，但在您停止该通道期间除外。 在切换过程中，将

消息传递到旧的传输队列，通过切换过程传输到新的传输队列，并通过集群发送方通道从新的传输队列转发。

3. 可选： 在通道切换时监视通道

   显示交换机期间的通道状态和传输队列深度。 以下示例显示系统集群传输队列的状态。

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

4. 可选： 监视消息 AMQ7341 **通道** *ChannelName* **的传输队列已从队列** *QueueName* **切换到写入队列管理器错误日志的** *QueueName* 。

将已停止的集群发送方通道切换到另一个集群传输队列
如果选择手动进行更改，那么在集群发送方通道停止时对其进行更改，并在集群发送方通道启动前将其从一个集群传输队列切换到另一个集群传输队列。

开始之前

您可以进行一些配置更改，现在希望使这些更改生效，而不启动受影响的集群发送方通道。 或者，作为任务中的其中一个步骤进行所需的配置更改。

如果减少切换过程必须传输到新传输队列的消息数，那么切换将更快地完成。 请阅读 将集群发送方通道切换到其他传输队列的过程如何工作 ，以了解在继续操作之前尝试清空传输队列的原因。

关于此任务

此任务切换由已停止或不活动的集群发送方通道提供服务的传输队列。 您可以执行此任务，因为集群发送方通道已停止，并且您希望立即切换其传输队列。 例如，由于某种原因，集群发送方通道未启动，或者存在一些其他配置问题。 要解决此问题，您决定创建集群发送方通道，并将旧集群发送方通道的传输队列与您定义的新集群发送方通道相关联。

更有可能的情况是，您希望控制何时执行集群传输队列的重新配置。 要完全控制重新配置，请停止通道，更改配置，然后切换传输队列。

过程

1. 停止要切换的通道

   a) 停止要切换的任何正在运行或不活动的通道。 停止不活动的集群发送方通道会阻止它在您进行配置更改时启动。

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

2. 可选： 进行配置更改。

   例如，请参阅第 47 页的『集群: 多个集群传输队列的示例配置』。

3. 将集群发送方通道切换到新的集群传输队列。

   Multi 在 多平台上，发出以下命令:

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

   z/OS 在 z/OS 上，使用 CSQUTIL 命令的 SWITCH 函数来切换消息或监视正在发生的情况。 使用以下命令。

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

   有关更多信息，请参阅 SWITCH 函数。

runswchl 或 CSQUTIL SWITCH 命令将旧传输队列上的任何消息传输到新的传输队列。 当此通道的旧传输队列上的消息数达到零时，交换机将完成。 该命令是同步的。 此命令在切换过程中将进度消息写入窗口。

在传输阶段，将以集群发送方通道为目标的现有消息和新消息传输到新的传输队列。

由于集群发送方通道已停止，因此消息会在新的传输队列上构建。 将已停止的集群发送方通道与 第 55 页的『将活动集群发送方通道切换到另一组集群传输队列』中的步骤 第 55 页的『2』 进行对比。 在该步骤中，集群发送方通道正在运行，因此消息不一定会在新的传输队列上构建。

4. 可选： 在通道切换时监视通道

   在另一个命令窗口中，显示切换期间的传输队列深度。 以下示例显示系统集群传输队列的状态。

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

5. 可选： 监视消息 AMQ7341 **通道** *ChannelName* **的传输队列已从队列** *QueueName* **切换到写入队列管理器错误日志的** *QueueName* 。
6. 重新启动已停止的集群发送方通道。

   通道不会自动启动，因为您已停止这些通道，并将其置于 STOPPED 状态。

   ```
   START CHANNEL(ChannelName)
   ```

相关参考

伦斯湖

解析通道

停止通道

集群: 迁移和修改最佳实践

本主题提供有关规划和管理 IBM MQ 集群的指导。 此信息是基于测试和客户反馈的指南。

1. 第 57 页的『在集群中移动对象』 (在集群内移动对象的最佳实践，而不安装任何修订包或新版本的 IBM MQ)。
2. 第 58 页的『升级和维护安装』 (在应用维护或升级以及测试新体系结构时保持工作集群体系结构正常运行的最佳实践)。

在集群中移动对象

应用程序及其队列

当您必须将托管在一个队列管理器上的队列实例移动到另一个队列管理器上时，可以使用工作负载均衡参数来确保平稳过渡。

创建要在其中进行新托管的队列实例，但使用集群工作负载均衡设置继续向原始实例发送消息，直到应用程序准备好切换为止。 这是通过以下步骤实现的:

1. 将现有队列的 **CLWLRANK** 属性设置为高值，例如 5。
2. 创建队列的新实例，并将其 **CLWLRANK** 属性设置为零。
3. 完成新系统的任何进一步配置，例如，针对队列的新实例部署和启动使用应用程序。
4. 将新队列实例的 **CLWLRANK** 属性设置为高于原始实例，例如 9。
5. 允许原始队列实例处理系统中的任何排队消息，然后删除该队列。

移动整个队列管理器

如果队列管理器位于同一主机上，但 IP 地址正在更改，那么此过程如下所示:

- 当正确使用 DNS 时，可以帮助简化该过程。 有关通过设置 连接名称 (CONNAME) 通道属性使用 DNS 的信息，请参阅 ALTER CHANNEL。
- 如果移动完整存储库，请确保在进行更改之前至少有一个其他完整存储库正在平稳运行 (例如，通道状态没有问题)。

- 使用 SUSPEND QMGR 命令暂挂队列管理器以避免流量堆积。
- 修改计算机的 IP 地址。 如果 CLUSRCVR 通道定义使用 CONNAME 字段中的 IP 地址，请修改此 IP 地址项。 可能需要清空 DNS 高速缓存以确保到处都有可用的更新。
- 当队列管理器重新连接到完整存储库时，通道自动定义会自动解析自身。
- 如果队列管理器托管了完整存储库，并且 IP 地址发生更改，请务必确保尽快切换分区以将任何手动定义的 CLUSSDR 通道指向新位置。 在执行此切换之前，这些队列管理器可能只能与剩余的 (未更改的) 完整存储库联系，并且可能会看到有关不正确通道定义的警告消息。
- 使用 RESUME QMGR 命令恢复队列管理器。

如果必须将队列管理器移至新主机，那么可以复制队列管理器数据并从备份复原。 但是，除非没有其他选项，否则建议不要执行此过程; 最好在新机器上创建队列管理器，并如上一节中所述复制队列和应用程序。 这种情况提供了一种平滑的翻转/回滚机制。

如果您决心使用备份来移动完整的队列管理器，请遵循以下最佳实践:

- 将整个进程视为来自备份的队列管理器复原，应用您通常用于操作系统环境的系统恢复的任何进程。
- 迁移后使用 **REFRESH CLUSTER** 命令来废弃所有本地持有的集群信息 (包括任何不确定的自动定义通道)，并强制对其进行重建。

  注: 对于大型集群，使用 **REFRESH CLUSTER** 命令可能会对正在运行的集群造成干扰，此外每隔 27 天在集群对象向所有相关队列管理器自动发送状态更新时也可能有干扰。 请参阅在大型集群中刷新可能会影响集群的性能和可用性。

创建队列管理器并从集群中的现有队列管理器复制设置 (如本主题中所述) 时，切勿将两个不同的队列管理器视为实际相同。 特别是，不要为新队列管理器提供相同的队列管理器名称和 IP 地址。 尝试 "删除" 替换队列管理器是导致 IBM MQ 集群中出现问题的常见原因。 高速缓存期望接收包括 **QMID** 属性在内的更新，并且状态可能已损坏。

如果意外创建了两个同名的不同队列管理器，那么建议使用 RESET CLUSTER **QMID** 命令从集群中弹出不正确的条目。

升级和维护安装

避免所谓的大爆炸场景 (例如，停止所有集群和队列管理器活动，对所有队列管理器应用所有升级和维护，然后同时启动所有内容)。 集群设计为仍与多个版本的队列管理器共存，因此建议采用规划良好的分阶段维护方法。

具有备份计划:

- 您是否进行了备份?
- 避免立即使用新的集群功能: 请等待您确定所有队列管理器都已升级到新级别，并确定您不会将它们中的任何一个回滚。 在某些队列管理器仍处于较早级别的集群中使用新的集群功能可能会导致未定义的行为。 例如，在从 IBM WebSphere MQ 6.0 移至 IBM WebSphere MQ 7.1 时，如果队列管理器定义了集群主题，那么 IBM WebSphere MQ 6.0 队列管理器将无法理解该定义或无法发布此主题。

首先迁移完整存储库。 虽然他们可以转发他们不了解的信息，但他们不能坚持下去，因此除非绝对必要，否则并不是建议的方法。 有关更多信息，请参阅 队列管理器集群迁移。

集群：使用 *REFRESH CLUSTER* 最佳实践
使用 **REFRESH CLUSTER** 命令可废弃有关集群的所有本地保存的信息，并从集群中的完整存储库重建该信息。 您不应该使用此命令，除非在特殊情况下。 如果确实需要使用它，那么有关如何使用它有特殊注意事项。 此信息是基于测试和客户反馈的指南。

仅在确实需要时运行 REFRESH CLUSTER

IBM MQ 集群技术可确保对集群配置的任何更改 (例如对集群队列的更改) 自动为需要了解信息的集群成员所知。 没有必要采取进一步的行政步骤来实现这种信息传播。

如果此类信息未到达需要该信息的集群中的队列管理器，例如，当应用程序首次尝试将其打开时，集群中的另一个队列管理器不知道集群队列，那么意味着集群基础结构中存在问题。 例如，可能无法在队列管理器与

完整存储库队列管理器之间启动通道。 因此，必须对观察到的任何不一致情况进行调查。 如果可能，请在不使用 **REFRESH CLUSTER** 命令的情况下解决此情况。

在此产品文档中其他位置记录的极少数情况下，或者当 IBM 支持人员请求时，您可以使用 **REFRESH CLUSTER** 命令来废弃有关集群的所有本地保存的信息，并从集群中的完整存储库重建该信息。

在大型集群中刷新可能会影响集群的性能和可用性

使用 **REFRESH CLUSTER** 命令可能会对正在进行的集群造成干扰，例如，在处理队列管理器集群资源的重新传播时，为完整存储库创建突然增加的工作。 如果在大型集群 (即，数百个队列管理器) 中刷新，那么应避免在日常工作中使用该命令 (如果可能) ，并使用替代方法来更正特定不一致情况。 例如，如果未在集群中正确传播集群队列，那么更新集群队列定义的初始调查技术 (例如，更改其描述) 将在集群中重新传播队列配置。 此过程可帮助确定问题并可能解决临时不一致问题。

如果无法使用替代方法，并且必须在大型集群中运行 **REFRESH CLUSTER** ，那么应在非高峰时间或维护时段执行此操作，以避免影响用户工作负载。 您还应避免在单个批处理中刷新大型集群，而是按 第 59 页的『避免集群对象发送自动更新时出现性能和可用性问题』中所述错开活动。

避免集群对象发送自动更新时出现性能和可用性问题

在队列管理器上定义新集群对象之后，将从定义开始每 27 天生成此对象的更新，并将此更新发送到集群中的每个完整存储库，然后再发送到任何其他相关队列管理器。 向队列管理器发出 **REFRESH CLUSTER** 命令时，将在指定集群中本地定义的所有对象上重置此自动更新的时钟。

如果在单个批处理中刷新大型集群 (即，数百个队列管理器) ，或者在其他情况下 (例如，从配置备份重新创建系统) ，那么在 27 天后，所有这些队列管理器都将同时向完整存储库重新发布其所有对象定义。 这可能再次导致系统运行速度显着降低，甚至变为不可用，直到所有更新都完成为止。 因此，当您必须在大型集群中刷新或重新创建多个队列管理器时，应该将活动错开几个小时或几天，以便后续自动更新不会定期影响系统性能。

系统集群历史记录队列

执行 **REFRESH CLUSTER** 时，队列管理器将在刷新之前生成集群状态的快照，并将其存储在 `SYSTEM.CLUSTER.HISTORY.QUEUE` (SCHQ) (如果在队列管理器上定义) 上。 此快照仅用于 IBM 服务，以防以后发生系统问题。

缺省情况下，在启动时在分布式队列管理器上定义 SCHQ。 对于 z/OS 迁移，必须手动定义 SCHQ。

SCHQ 上的消息将在三个月后到期。

相关概念

第 88 页的『针对发布/预订集群的 REFRESH CLUSTER 注意事项』
发出 **REFRESH CLUSTER** 命令会导致队列管理器临时废弃本地保存的有关集群的信息，包括任何集群主题及其关联的代理预订。

运行 REFRESH CLUSTER 时发现的应用程序问题

相关参考

MQSC 命令参考：REFRESH CLUSTER

集群*:* 可用性，多实例和灾难恢复
本主题提供有关规划和管理 IBM MQ 集群的指导。 此信息是基于测试和客户反馈的指南。

IBM MQ 集群本身不是高可用性解决方案，但在某些情况下，可以使用它来提高使用 IBM MQ 的服务的可用性，例如，在不同的队列管理器上具有多个队列实例。 本部分提供了有关确保 IBM MQ 基础结构尽可能高可用性的指导信息，以便可以在此类体系结构中使用该基础结构。

注: 其他高可用性和灾难恢复解决方案可用于 IBM MQ，请参阅 配置高可用性，恢复和重新启动。

集群资源的可用性

通常建议维护两个完整存储库的原因是，丢失一个存储库对于集群的顺利运行并不重要。 即使两者都变得不可用，对于部分存储库持有的现有知识也有 60 天的宽限期，尽管在此事件中没有新的或先前未访问的资源 (例如，队列) 可用。

使用集群来提高应用程序可用性

通过使用队列和应用程序的多个实例，集群可以帮助设计高可用性应用程序 (例如请求/响应类型服务器应用程序)。 如果需要，优先级属性可以优先使用 "活动" 应用程序，除非例如队列管理器或通道变为不可用。 这对于在发生问题时快速切换以继续处理新消息非常强大。

但是，传递到集群中特定队列管理器的消息仅保留在该队列实例上，并且在恢复该队列管理器之前不可用于处理。 因此，为了实现真正的数据高可用性，您可能需要考虑其他技术，例如多实例队列管理器。

多实例队列管理器

软件高可用性 (多实例) 是用于保持现有消息可用的内置产品。 请参阅 将 IBM MQ 与高可用性配置配合使用， 创建多实例队列管理器以及以下部分以获取更多信息。 只要集群中的所有队列管理器至少正在运行 IBM WebSphere MQ 7.0.1，就可以使用此技术使集群中的任何队列管理器具有高可用性。 如果集群中的任何队列管理器处于先前级别，那么当它们故障转移到辅助 IP 时，可能会丢失与多实例队列管理器的连接。

正如本主题中先前所讨论的那样，只要配置了两个完整的存储库，它们几乎按其性质高度可用。 如果需要，可以将 IBM MQ 软件高可用性/多实例队列管理器用于完整存储库。 没有强烈的理由使用这些方法，事实上，对于临时中断，这些方法可能会在故障转移期间造成额外的性能成本。 建议不要使用软件 HA 而不是运行两个完整存储库，因为例如，在发生单个通道中断时，它不一定会故障转移，但可能会导致部分存储库无法查询集群资源。

灾难恢复

灾难恢复 (例如，从存储队列管理器数据的磁盘损坏时进行恢复) 很难很好地执行; IBM MQ 可以提供帮助，但它无法自动执行此操作。 IBM MQ 中唯一的 "true" 灾难恢复选项 (不包括任何操作系统或其他底层复制技术) 是从备份复原。 在以下情况下，需要考虑一些特定于集群的点:

- 测试灾难恢复方案时请小心。 例如，如果测试备份队列管理器的操作，那么在使它们在同一网络中联机时要小心，因为有可能意外地加入实时集群，并通过托管与实时集群队列管理器中相同的指定队列来开始 "窃取" 消息。
- 灾难恢复测试不得干扰正在运行的实时集群。 避免干扰的方法包括:
  - 在防火墙级别完成网络分离或分离。
  - z/OS 未启动通道启动或 z/OS **chinit** 地址空间。
  - 在发生实际灾难恢复方案之前，或者除非发生实际灾难恢复方案，否则不会向灾难恢复系统发放实时 TLS 证书。
- 在集群中复原队列管理器的备份时，备份可能与集群的其余部分不同步。 **REFRESH CLUSTER** 命令可以解析更新并与集群同步，但 **REFRESH CLUSTER** 命令必须用作最后的手段。 请参阅第 58 页的『集群：使用 REFRESH CLUSTER 最佳实践』。 查看任何内部流程文档和 IBM MQ 文档，以了解在使用该命令之前是否错过了简单的步骤。
- 对于任何恢复，应用程序都必须处理重放和数据丢失。 必须决定是将队列清除到已知状态，还是在其他地方有足够的信息来管理重放。

规划分布式发布/预订网络

您可以创建队列管理器网络，在该网络中，在一个队列管理器上创建的预订将接收由连接到网络中另一个队列管理器的应用程序发布的匹配消息。 要选择合适的拓扑，需要考虑您对手动控制，网络大小，更改频率，可用性和可伸缩性的需求。

开始之前

此任务假定您了解分布式发布/预订网络是什么以及它们的工作方式。 有关技术概述，请参阅 分布式发布/预订网络。

关于此任务

发布/预订网络有三种基本拓扑:

- 直接路由集群
- 主题主机路由集群
- 层次结构

对于前两个拓扑，起始点是 IBM MQ 集群配置。 可以使用或不使用集群创建第三个拓扑。 请参阅第 17 页的『规划分布式队列和集群』，以获取有关规划底层队列管理器网络的信息。

直接路由集群 是在集群已存在时要配置的最简单拓扑。 您在任何队列管理器上定义的任何主题都将自动在集群中的每个队列管理器上可用，发布将直接从发布应用程序连接的任何队列管理器路由到存在匹配预订的每个队列管理器。 这种简单的配置依赖于 IBM MQ 在集群中的每个队列管理器之间保持高级别的信息共享和连接。 对于小型和简单的网络 (即，少量队列管理器以及一组相当静态的发布者和订户) ，这是可以接受的。 但是，在更大或更动态的环境中使用时，开销可能会令人望而却步。 请参阅第 64 页的『发布/预订集群中的直接路由』。

主题主机路由集群 通过使集群中的任何队列管理器上定义的任何主题自动在集群中的每个队列管理器上可用，提供与直接路由集群相同的优势。 但是，主题主机路由集群要求您仔细选择托管每个主题的队列管理器，因为该主题的所有信息和发布都通过这些主题主机队列管理器传递。 这意味着系统不必维护所有队列管理器之间的通道和信息流。 但是，这也意味着发布可能不再直接发送到订户，而是可以通过主题主机队列管理器进行路由。 由于这些原因，系统上可能会有额外的负载，特别是在托管主题的队列管理器上，因此需要仔细规划拓扑。 对于包含许多队列管理器或托管一组动态发布者和订户 (即，频繁添加或除去的发布者或订户) 的网络，此拓扑特别有效。 可以定义其他主题主机以提高路径可用性并水平缩放发布工作负载。 请参阅第 69 页的『发布/预订集群中的主题主机路由』。

层次结构 需要设置最手动的配置，并且是要修改的最难的拓扑。 您必须手动配置层次结构中每个队列管理器与其直接关系之间的关系。 配置关系后，将发布 (与前两种拓扑一样) 路由到层次结构中其他队列管理器上的预订。 发布使用层次结构关系进行路由。 这允许将非常特定的拓扑配置为满足不同的需求，但这也可能导致发布需要许多 “中继段” 通过中间队列管理器来访问预订。 对于发布，始终只有一个通过层次结构的路由，因此每个队列管理器的可用性至关重要。 通常只有在无法配置单个集群的情况下，才会首选层次结构; 例如，跨多个组织时。 请参阅第 89 页的『在发布/预订层次结构中进行路由』。

必要时，可将上述三种拓扑组合起来，以解决特定的地形要求。 有关示例，请参阅 组合多个集群的主题空间。

要为分布式发布/预订网络选择合适的拓扑，您需要考虑以下广泛问题:

- 您的网络将有多大?
- 您需要对其配置进行多少手动控制?
- 无论在主题和预订方面，还是在队列管理器方面，系统都将具有多大的动态?
- 您的可用性和可伸缩性需求是什么?
- 所有队列管理器可以直接相互连接吗?

过程

- 估算网络需要的大小。
  a) 估算需要多少主题。
  b) 估算您期望拥有的发布者和订户数量。
  c) 估算将有多少队列管理器参与发布/预订活动。

     另请参阅第 78 页的『发布/预订集群: 最佳实践』，特别是以下部分:

     – 如何调整系统大小
     – 限制发布/预订活动中涉及的集群队列管理器数的原因
     – 如何决定要集群的主题

  如果您的网络将具有许多队列管理器，并处理许多发布者和订户，那么您可能需要使用主题主机路由集群或层次结构。 直接路由集群几乎无需手动配置，可以成为小型或静态网络的良好解决方案。
- 请考虑您需要对每个主题，发布者或订户托管的队列管理器进行多少手动控制。
  a) 请考虑某些队列管理器的能力是否低于其他队列管理器。
  b) 请考虑到某些队列管理器的通信链路是否比其他队列管理器更脆弱。
  c) 确定您期望主题具有许多出版物和少量订户的案例。
  d) 确定您期望主题具有许多订户和少量出版物的案例。

在所有拓扑中，发布将传递到其他队列管理器上的预订。 在直接路由的集群中，这些发布将采用最短的预订路径。 在主题主机路由集群或层次结构中，您可以控制发布所采用的路径。 如果队列管理器的功能不同，或者具有不同的可用性和连接级别，那么您可能希望将特定工作负载分配给特定队列管理器。 您可以使用主题主机路由集群或层次结构来执行此操作。

在所有拓扑中，尽可能将发布应用程序与预订放在同一队列管理器上，从而最大程度地减少开销并实现性能最大化。 对于主题主机路由集群，请考虑将发布者或订户放在托管该主题的队列管理器上。 这将除去队列管理器之间的任何额外 "中继段" ，以将发布传递给订户。 在主题具有许多发布者和很少的订户，或者具有许多订户和很少的发布者的情况下，此方法特别有效。 例如，请参阅 使用集中发布程序或订户进行主题主机路由。

另请参阅第 78 页的『发布/预订集群: 最佳实践』，特别是以下部分:

– 如何决定要集群的主题
– 发布程序和预订位置

• 请考虑网络活动的动态程度。

  a) 估算在不同主题上添加和除去订户的频率。

  每当从队列管理器添加或除去预订，并且该预订是该特定主题字符串的第一个或最后一个预订时，都会将该信息传达给拓扑中的其他队列管理器。 在直接路由的集群和层次结构中，此预订信息将传播到拓扑中的每个队列管理器，无论它们是否具有主题上的发布程序。 如果拓扑由许多队列管理器组成，那么这可能是显着的性能开销。 在主题主机路由集群中，此信息仅传播到那些托管映射到预订的主题字符串的集群主题的队列管理器。

  另请参阅第 78 页的『发布/预订集群: 最佳实践』的 预订更改和动态主题字符串 部分。

  注: 在非常动态的系统中，许多唯一主题字符串的集合正在快速且持续地进行更改，因此最好将模型切换到 "随时随地发布" 方式。 请参阅 发布/预订网络中的预订性能。

  b) 请考虑队列管理器在拓扑中的动态程度。

  层次结构要求将拓扑中队列管理器中的每个更改手动插入或从层次结构中除去，并在层次结构中的较高级别更改队列管理器时小心谨慎。 层次结构中的队列管理器通常也使用手动配置的通道连接。 您必须维护这些连接，在从层次结构中添加和除去队列管理器时添加和除去通道。

  在发布/预订集群中，队列管理器会自动连接到首次加入集群时所需的任何其他队列管理器，并自动了解主题和预订。

• 考虑您的路由可用性和发布流量可伸缩性需求。

  a) 决定是否需要始终具有从发布队列管理器到预订队列管理器的可用路由，即使队列管理器不可用时也是如此。

  b) 请考虑您需要网络的可伸缩性。 确定发布流量级别是否过高，无法通过单个队列管理器或通道进行路由，以及该级别的发布流量是必须由单个主题分支处理，还是可以跨多个主题分支进行传播。

  c) 请考虑是否需要维护消息排序。

由于直接路由的集群将消息从发布队列管理器直接发送到预订队列管理器，因此您无需考虑沿路由的中间队列管理器的可用性。 同样，不考虑对中间队列管理器进行缩放。 但是，如前所述，自动维护集群中所有队列管理器之间的通道和信息流的开销会显着影响性能，尤其是在大型或动态环境中。

可以针对各个主题调整主题主机路由集群。 您可以确保在不同的队列管理器上定义具有大量发布工作负载的主题树的每个分支，并且确保每个队列管理器具有足够的性能，并且可用于该主题树的该分支的预期工作负载。 您还可以通过在多个队列管理器上定义每个主题来进一步提高可用性和水平缩放。 这允许系统围绕不可用的主题主机队列管理器进行路由，并使它们之间的工作负载均衡发布流量。 但是，在多个队列管理器上定义给定主题时，还会引入以下约束:

– 在发布之间丢失消息排序。
– 不能使用保留发布。 请参阅第 87 页的『发布/预订集群中保留的发布的设计注意事项』。

无法通过多个路径配置层次结构中的路由的高可用性或可伸缩性。

另请参阅第 78 页的『发布/预订集群: 最佳实践』的 发布流量 部分。

• 根据这些计算，使用提供的链接来帮助您决定是使用主题主机路由集群，直接路由集群，层次结构还是这些拓扑的混合。

下一步做什么

您现在已准备好配置分布式发布/预订网络。

相关任务

配置队列管理器集群

配置分布式队列

配置发布/预订集群

将队列管理器连接到发布/预订层次结构

设计发布/预订集群

有两种基本发布/预订集群拓扑: 直接路由 和 主题主机路由。 每一个都有不同的好处。 设计发布/预订集群时，请选择最符合预期网络需求的拓扑。

有关两个发布/预订集群拓扑的概述，请参阅 发布/预订集群。 要帮助您评估网络需求，请参阅 第 60 页的『规划分布式发布/预订网络』 和 第 78 页的『发布/预订集群: 最佳实践』。

通常，这两种集群拓扑都具有以下优点:

- 基于点到点集群拓扑的简单配置。
- 自动处理加入和离开集群的队列管理器。
- 通过添加额外的队列管理器并在这些队列管理器和发布程序之间分发额外的预订和发布程序，简化其他预订和发布程序的扩展。

但是，随着需求变得更加具体，这两种拓扑具有不同的优点。

直接路由的发布/预订集群

通过直接路由，集群中的任何队列管理器都会将发布从已连接的应用程序直接发送到具有匹配预订的集群中的任何其他队列管理器。

直接路由的发布/预订集群提供以下优势:

- 发往同一集群中特定队列管理器上的预订的消息将直接传输到该队列管理器，并且不需要通过中间队列管理器。 与主题主机路由拓扑或分层拓扑相比，这可以提高性能。
- 由于所有队列管理器都直接相互连接，因此此拓扑的路由基础结构中没有单点故障。 如果一个队列管理器不可用，那么集群中其他队列管理器上的预订仍能够从可用队列管理器上的发布程序接收消息。
- 配置非常简单，尤其是在现有集群上。

使用直接路由的发布/预订集群时要考虑的事项:

- 集群中的所有队列管理器都可以识别集群中的所有其他队列管理器。
- 集群中用于托管集群主题的一个或多个预订的队列管理器会自动创建集群中所有其他队列管理器的集群发送方通道，即使这些队列管理器未在任何集群主题上发布消息也是如此。
- 队列管理器上对集群主题下的主题字符串的第一个预订会导致将消息发送到集群中的每个其他队列管理器。 同样，要删除的主题字符串上的最后一个预订也会生成一条消息。 在集群主题下使用的单个主题字符串越多，预订的更改速率越高，发生的队列间管理器通信越多。
- 即使队列管理器既不发布也不预订这些主题，集群中的每个队列管理器也会保留其已通知的已预订主题字符串的知识。

由于上述原因，定义了直接路由主题的集群中的所有队列管理器都将产生额外的开销。 集群中的队列管理器越多，开销越大。 同样，预订的主题字符串越多，其更改速率越大，开销越大。 这可能会导致在大型或动态直接路由的发布/预订集群中的小型系统上运行的队列管理器负载过多。 请参阅 直接路由发布/预订性能 以获取更多信息。

当您知道集群无法容纳直接路由的集群发布/预订的开销时，可以改为使用 主题主机路由的发布/预订。 或者，在极端情况下，可以通过在集群中的每个队列管理器上将队列管理器属性 **PSCLUS** 设置为 `DISABLED` 来完全禁用集群发布/预订功能。 请参阅第 85 页的『禁止集群发布/预订』。 这将阻止创建任何集群主题，从而确保网络不会产生与集群发布/预订相关联的任何开销。

主题主机路由的发布/预订集群

通过主题主机路由，以管理方式定义集群主题的队列管理器将成为发布的路由器。 来自集群中非托管队列管理器的发布通过托管队列管理器路由到具有匹配预订的集群中的任何队列管理器。

主题主机路由的发布/预订集群在直接路由的发布/预订集群上提供以下额外优势:

- 只有定义了主题主机路由主题的队列管理器才会了解集群中的所有其他队列管理器。
- 仅主题主机队列管理器需要能够连接到集群中的所有其他队列管理器，并且通常仅连接到存在预订的那些队列管理器。 因此，在队列管理器之间运行的通道明显较少。
- 托管一个或多个集群主题预订的集群队列管理器将自动创建集群发送方通道，仅针对托管映射到预订主题字符串的集群主题的队列管理器。
- 队列管理器上对集群主题下主题字符串的第一个预订会导致将消息发送到集群主题所在的集群中的队列管理器。 同样，要删除的主题字符串上的最后一个预订也会生成一条消息。 在集群主题下使用的单个主题字符串越多，预订的更改速率越高，队列管理器之间的通信就越多，但仅在预订主机和主题主机之间进行。
- 对物理配置进行更多控制。 通过直接路由，所有队列管理器都必须参与发布/预订集群，从而增加其开销。 通过主题主机路由，只有主题主机队列管理器知道其他队列管理器及其预订。 您可以显式选择主题主机队列管理器，因此可以确保这些队列管理器在足够的设备上运行，并且可以将功能较差的系统用于其他队列管理器。

使用主题主机路由发布/预订集群时要考虑的事项:

- 当发布者或订户不在主题主管队列管理器上时，将在发布队列管理器与预订队列管理器之间引入额外的 “中继段” 。 额外的 “中继段” 导致的等待时间可能意味着主题主机路由的效率低于直接路由。
- 在大型集群上，主题主机路由可缓解直接路由的显着性能和扩展问题。
- 您可以选择在单个队列管理器上定义所有主题，或者在极少数队列管理器上定义所有主题。 如果执行此操作，请确保主题主机队列管理器托管在具有良好连接的强大系统上。
- 可以在多个队列管理器上定义同一主题。 这会提高主题的可用性，还会提高可伸缩性，因为 IBM MQ 工作负载会在该主题的所有主机上均衡某个主题的出版物。 但是，请注意，在多个队列管理器上定义同一主题会丢失该主题的消息顺序。
- 通过在不同的队列管理器上托管不同的主题，可以提高可伸缩性，而不会丢失消息顺序。

相关任务

配置发布/预订集群

调整分布式发布/预订网络

分布式发布/预订故障诊断

相关参考

发布/预订集群方案

发布/预订集群中的直接路由

来自任何发布队列管理器的发布将直接路由到具有匹配预订的集群中的任何其他队列管理器。

有关如何在发布/预订层次结构和集群中的队列管理器之间路由消息的简介，请参阅 分布式发布/预订网络。

直接路由的发布/预订集群的行为如下所示:

- 所有队列管理器自动知道所有其他队列管理器。
- 所有具有集群主题预订的队列管理器都会创建到集群中所有其他队列管理器的通道，并将其预订通知给这些队列管理器。
- 应用程序发布的消息将从其连接到的队列管理器路由到存在匹配预订的每个队列管理器。

下图显示了当前未用于发布/预订或点到点活动的队列管理器集群。 请注意，集群中的每个队列管理器仅与完整存储库队列管理器连接。

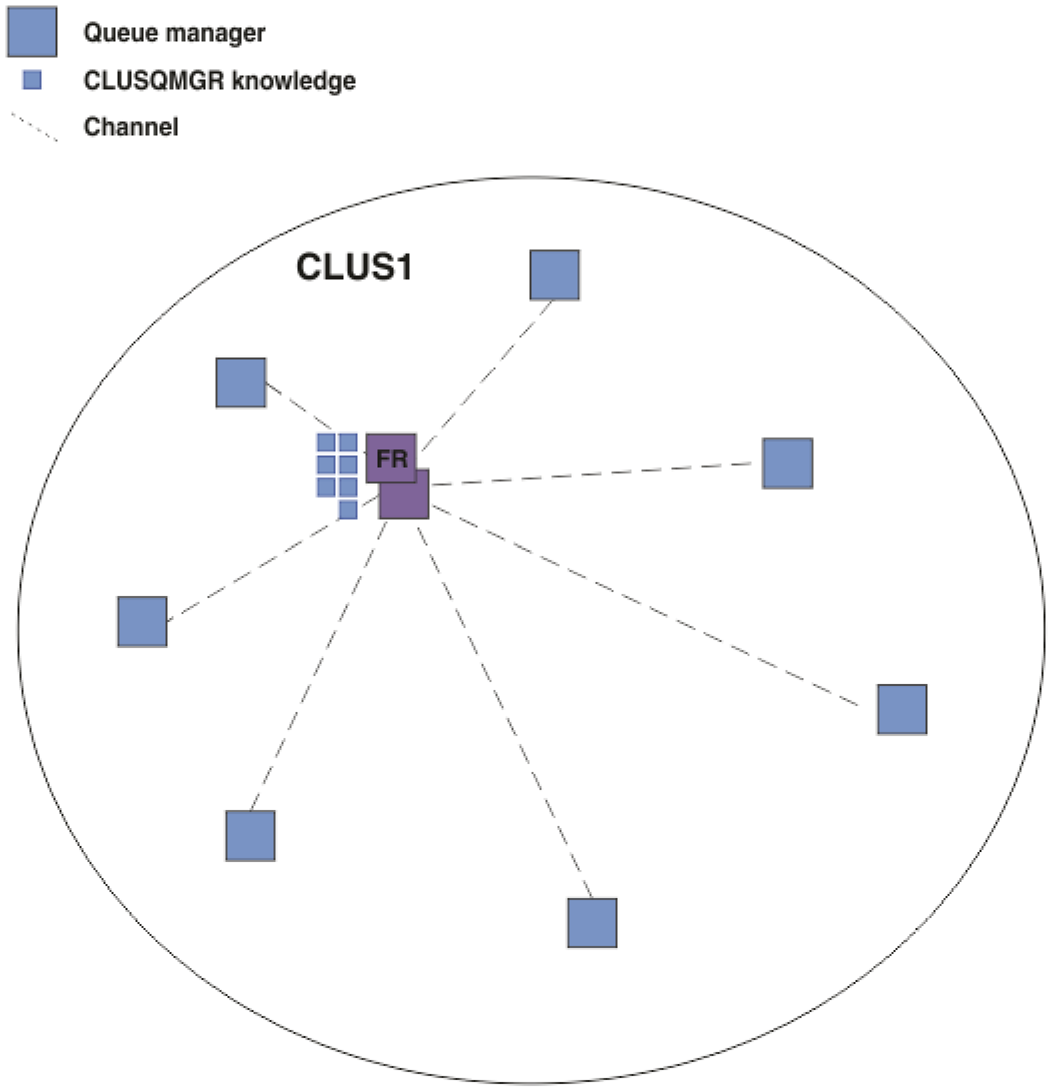


图 *16:* 队列管理器集群

要使发布在直接路由集群中的队列管理器之间流动，请按 配置发布/预订集群中所述对主题树的分支进行集群，并指定 直接路由 (缺省值)。

在直接路由的发布/预订集群中，在集群中的任何队列管理器上定义主题对象。 执行此操作时，完整存储库队列管理器会自动将对象的知识以及集群中所有其他队列管理器的知识推送到集群中的所有队列管理器。 这在任何队列管理器引用主题之前发生:

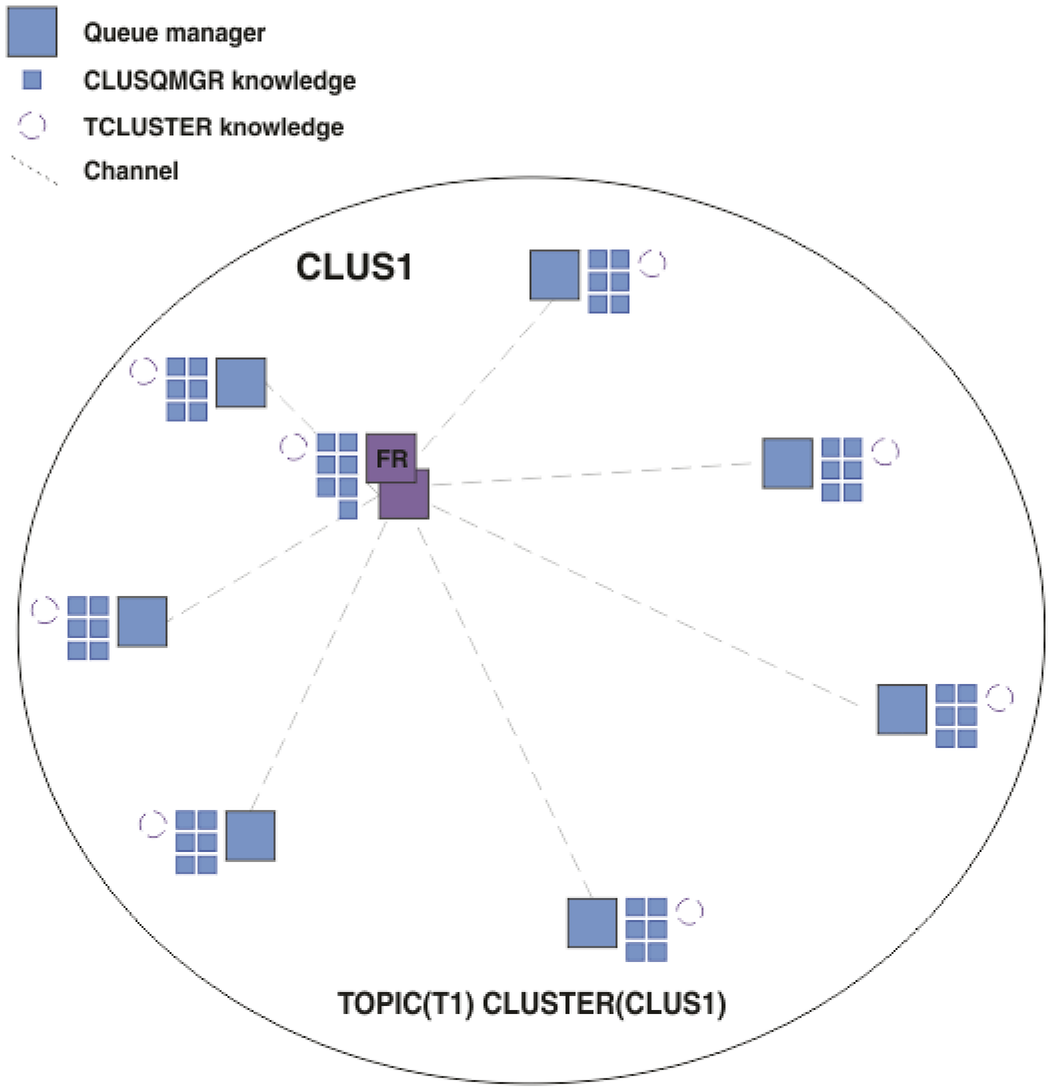


图 *17:* 直接路由的发布/预订集群

创建预订时，托管该预订的队列管理器会向集群中的每个队列管理器建立通道，并发送该预订的详细信息。此分布式预订知识由每个队列管理器上的 代理预订 表示。 在集群中与该代理预订的主题字符串相匹配的任何队列管理器上生成发布时，将建立从发布者队列管理器到托管预订的每个队列管理器的集群通道，并将消息发送到每个队列管理器。

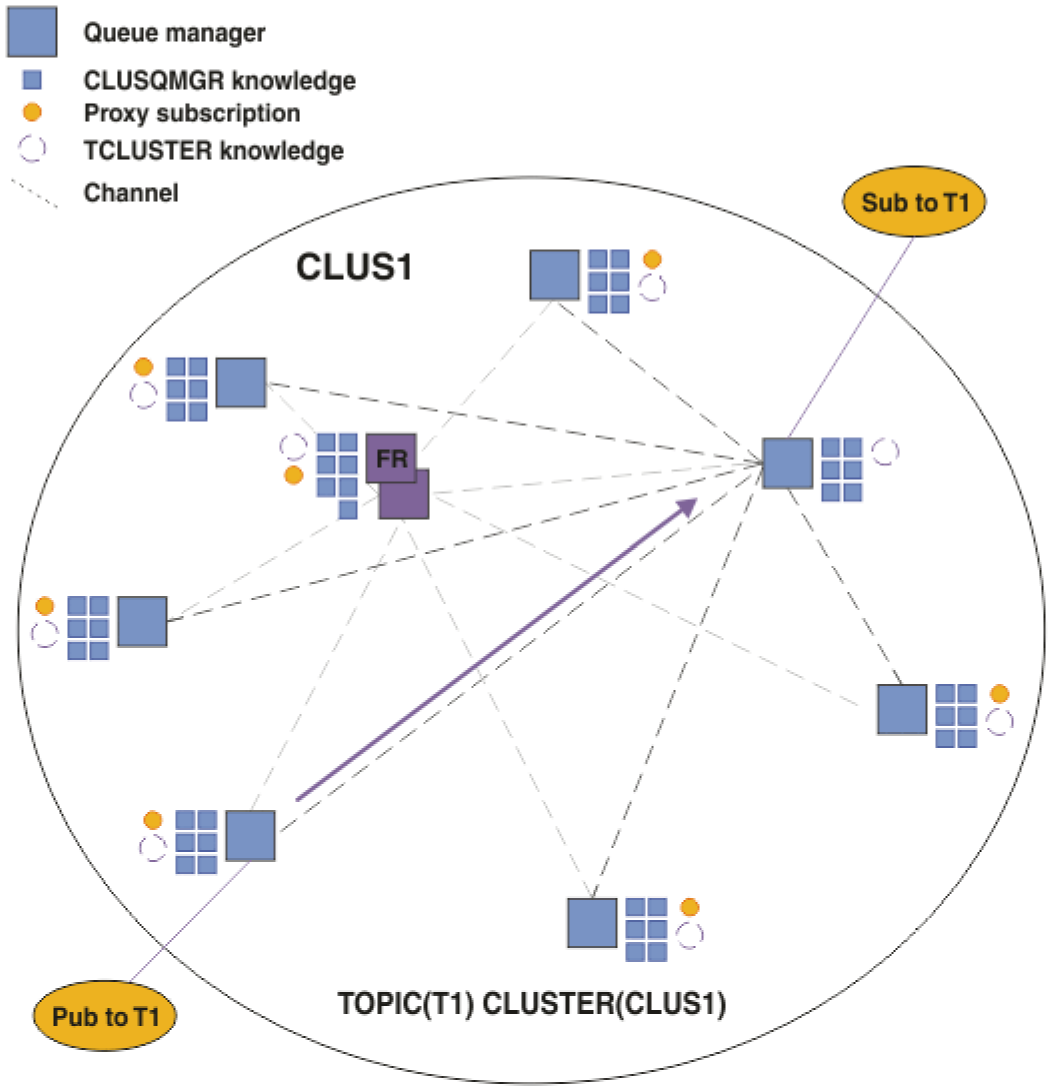


图 *18:* 具有集群主题的发布者和订户的直接路由发布/预订集群

将发布直接路由到预订托管队列管理器可简化配置，并最大程度地缩短将发布交付到预订的等待时间。

但是，根据预订和发布程序的位置，集群可以快速完全互连，每个队列管理器都与其他每个队列管理器直接连接。 这在您的环境中可能是可接受的，也可能是不可接受的。 同样，如果要预订的主题字符串集频繁更改，那么在所有队列管理器之间传播该信息的开销也会很大。 直接路由的发布/预订集群中的所有队列管理器都必须能够处理这些开销。

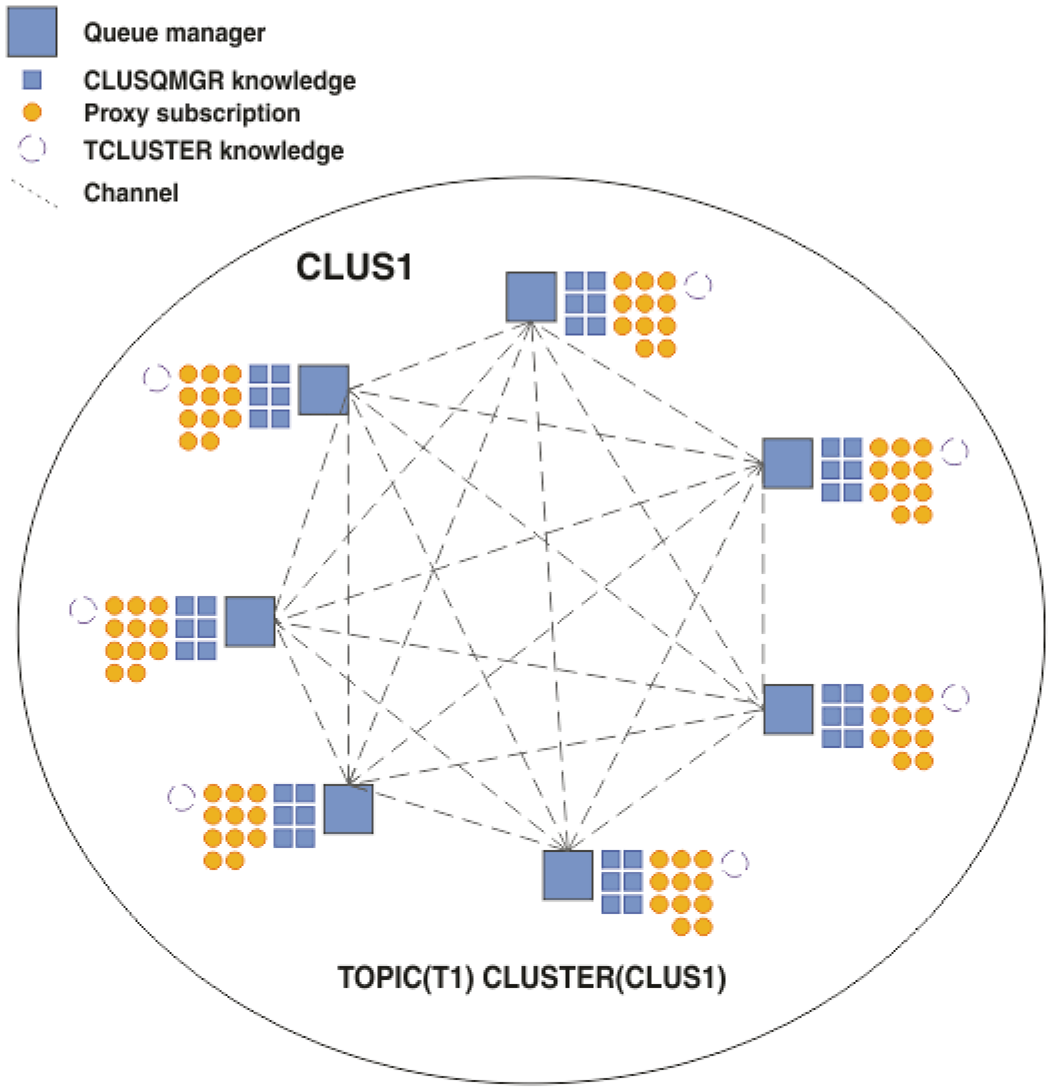


图 *19:* 完全互连的直接路由发布/预订集群

摘要和其他注意事项

直接路由的发布/预订集群无需手动干预即可创建或管理，并在发布者和订户之间提供直接路由。 对于某些配置，它通常是最合适的拓扑，尤其是具有很少队列管理器的集群，或者具有可接受的高队列管理器连接且预订不经常更改的集群。 但是，这也会对您的系统施加某些约束:

- 每个队列管理器上的负载与集群中的队列管理器总数成比例。 因此，在更大的集群中，单个队列管理器和整个系统可能会迂到性能问题。
- 缺省情况下，预订的所有集群主题字符串都将在整个集群中传播，并且发布仅传播到预订了关联主题的远程队列管理器。 因此，对预订集的快速更改可能成为限制因素。 您可以更改此缺省行为，而是将所有发布传播到所有队列管理器，这将除去对代理预订的需求。 这将减少预订知识流量，但可能会增加发布流量以及每个队列管理器建立的通道数。 请参阅 发布/预订网络中的预订性能。

  注: 类似的限制也适用于层次结构。
- 由于发布/预订队列管理器的互连性质，代理预订在网络中的所有节点之间传播需要时间。 远程发布不一定会立即开始预订，因此在预订新主题字符串之后可能不会发送早期发布。 您可以通过将所有发布传播到所有队列管理器来除去预订延迟所导致的问题，这将除去对代理预订的需求。 请参阅 发布/预订网络中的预订性能。

  注: 此限制也适用于层次结构。

在使用直接路由之前，请探索 第 69 页的『发布/预订集群中的主题主机路由』和 第 89 页的『在发布/预订层次结构中进行路由』中详细描述的替代方法。

发布/预订集群中的主题主机路由

来自集群中非托管队列管理器的发布通过托管队列管理器路由到具有匹配预订的集群中的任何队列管理器。

有关如何在发布/预订层次结构和集群中的队列管理器之间路由消息的简介，请参阅 分布式发布/预订网络。

要了解主题主机路由的行为和优点，最好首先了解 第 64 页的『发布/预订集群中的直接路由』。

主题主机路由的发布/预订集群的行为如下所示:

- 集群受管主题对象是在集群中的各个队列管理器上手动定义的。 这些称为 主题主机队列管理器。
- 在集群队列管理器上进行预订时，将创建从预订主机队列管理器到主题主机队列管理器的通道，并且仅在托管主题的队列管理器上创建代理预订。
- 当应用程序将信息发布到主题时，已连接的队列管理器始终将该发布转发到主管该主题的一个队列管理器，该队列管理器将该主题传递到集群中具有与该主题的匹配预订的所有队列管理器。

以下示例更详细地说明了此过程。

使用单个主题主机进行主题主机路由

要使发布在主题主机路由集群中的队列管理器之间流动，请按 配置发布/预订集群中所述对主题树的分支进行集群，并指定 主题主机路由。

在集群中的多个队列管理器上定义主题主机路由主题对象有多种原因。 但是，为了简单起见，我们从单个主题主机开始。

下图显示了当前未用于发布/预订或点到点活动的队列管理器集群。 请注意，集群中的每个队列管理器仅与完整存储库队列管理器连接。

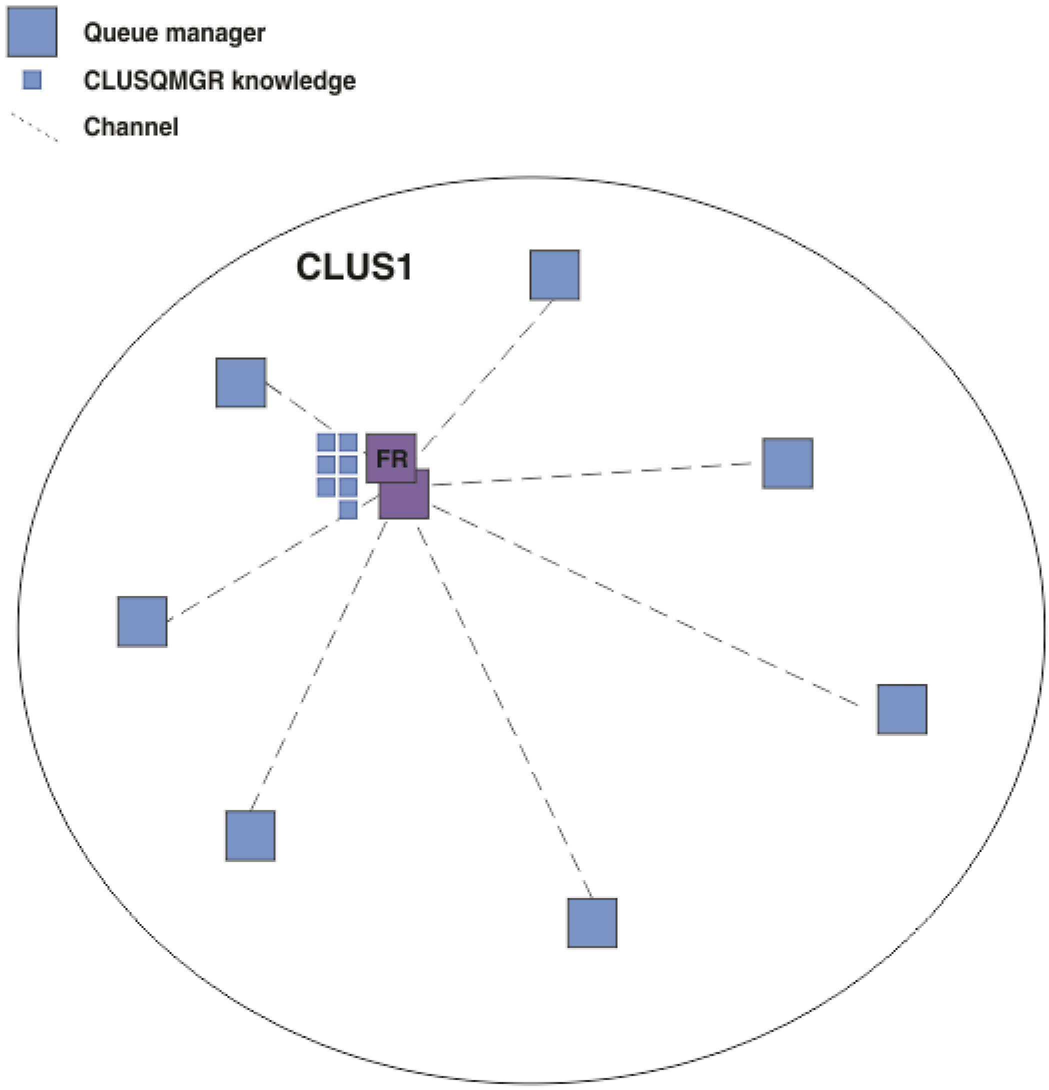


图 *20:* 队列管理器集群

在主题主机路由的发布/预订集群中，您在集群中的特定队列管理器上定义主题对象。 然后，发布/预订流量流经该队列管理器，使其成为集群中的关键队列管理器并增加其工作负载。 由于这些原因，建议不要使用完整存储库队列管理器，而要使用集群中的另一个队列管理器。 在主机队列管理器上定义主题对象时，完整存储库队列管理器会将该对象及其主机的知识自动推送到集群中的所有其他队列管理器。 请注意，与 直接路由不同，不会向每个队列管理器讲述集群中的每个其他队列管理器。

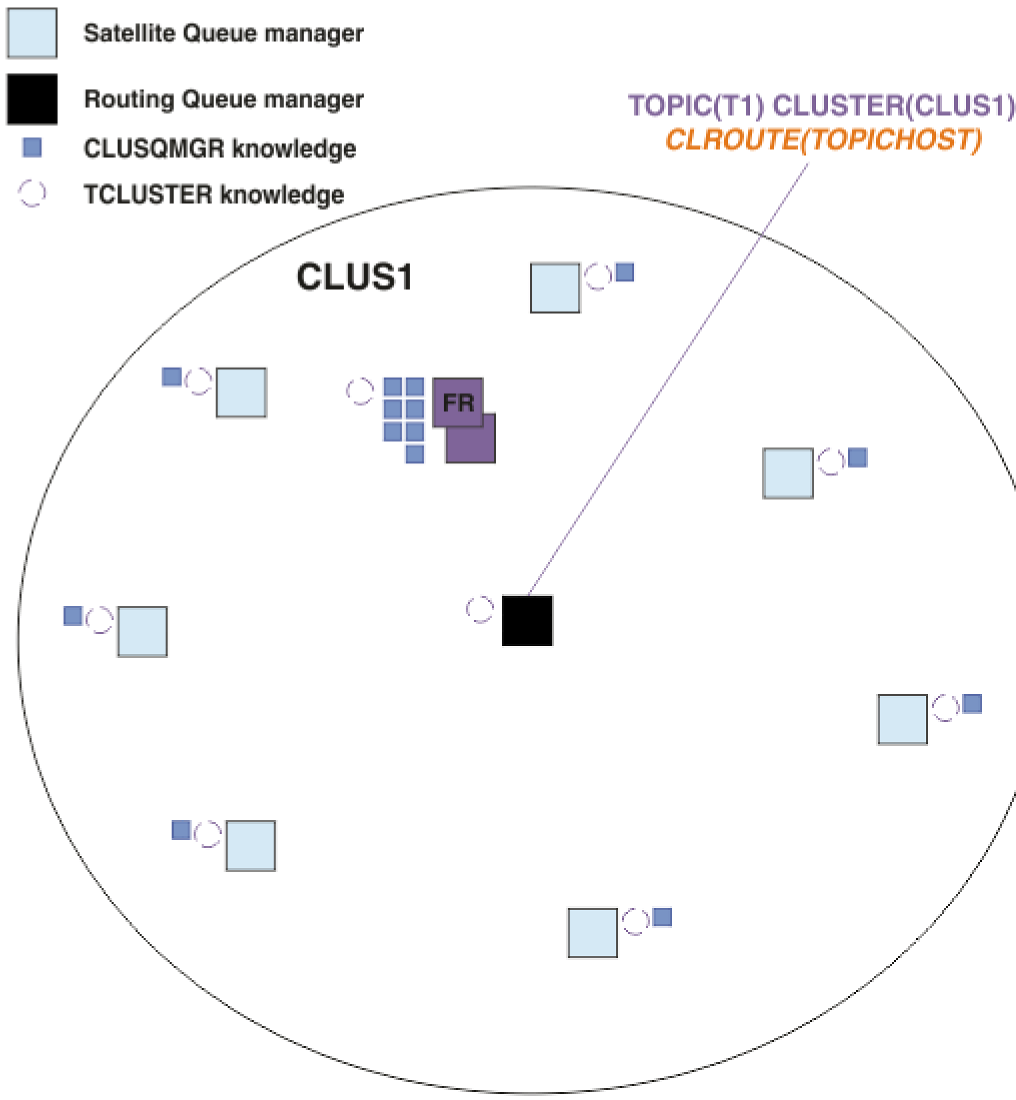


图 *21:* 在一个主题主机上定义了一个主题的主题主机路由发布/预订集群

在队列管理器上创建预订时，将在预订队列管理器与主题主机队列管理器之间创建通道。 预订队列管理器仅连接到主题主机队列管理器，并发送预订的详细信息 (以 代理预订的形式)。 主题主机队列管理器不会将此预订信息转发到集群中的任何其他队列管理器。

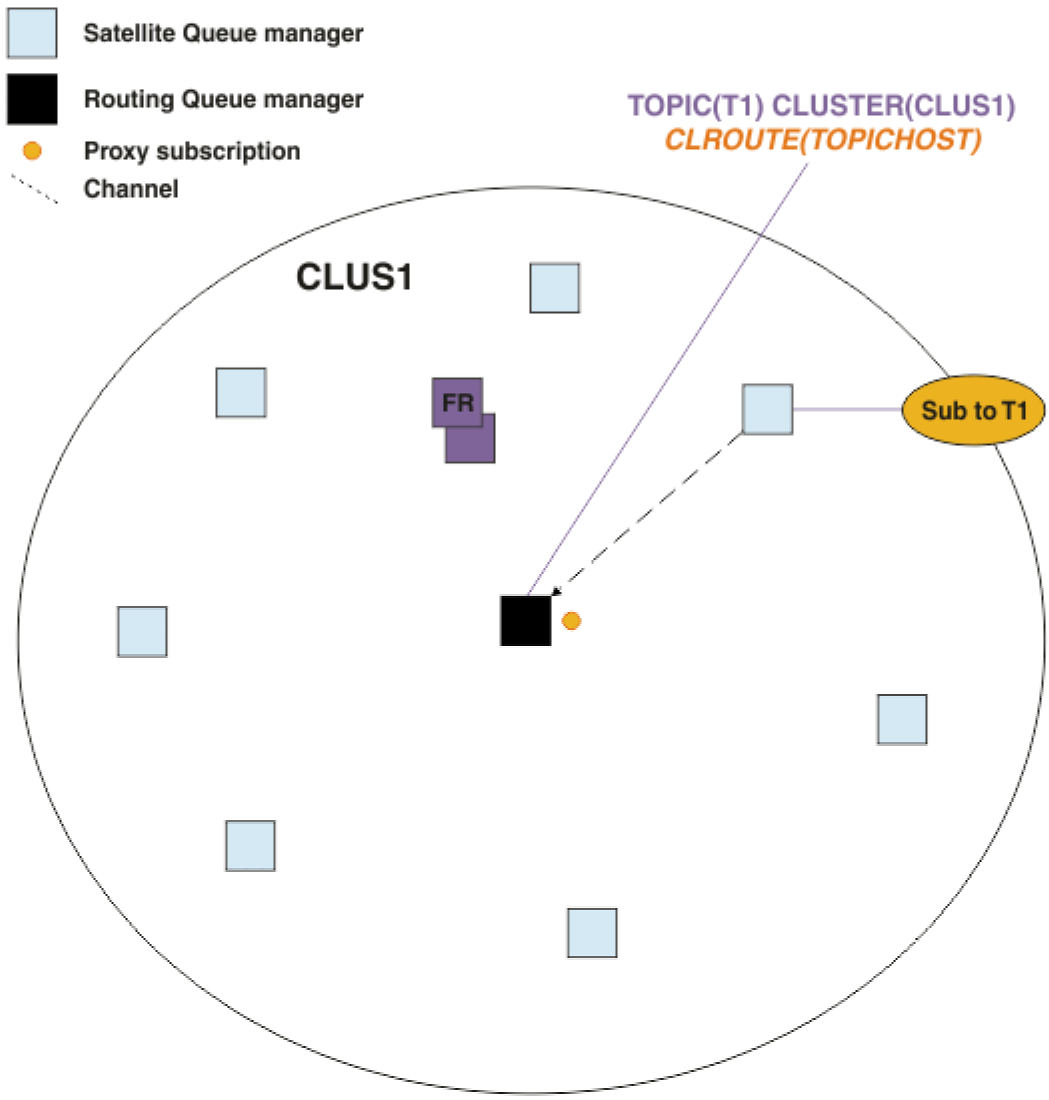


图 *22:* 主题主机路由的发布/预订集群，其中一个主题主机上定义了一个主题，一个订户

当发布应用程序连接到另一个队列管理器并发布消息时，将在发布队列管理器与主题主机队列管理器之间创建通道，并将消息转发到该队列管理器。 发布队列管理器不知道集群中其他队列管理器上的任何预订，因此即使集群中没有该主题的订户，也会将消息转发到主题主机队列管理器。 发布队列管理器仅连接到主题主机队列管理器。 发布通过主题主机路由到预订队列管理器 (如果存在)。

将直接满足与发布程序相同的队列管理器上的预订，而不首先将消息发送到主题主机队列管理器。

请注意，由于每个主题主机队列管理器都具有关键作用，因此您必须选择可处理主题托管的负载，可用性和连接需求的队列管理器。

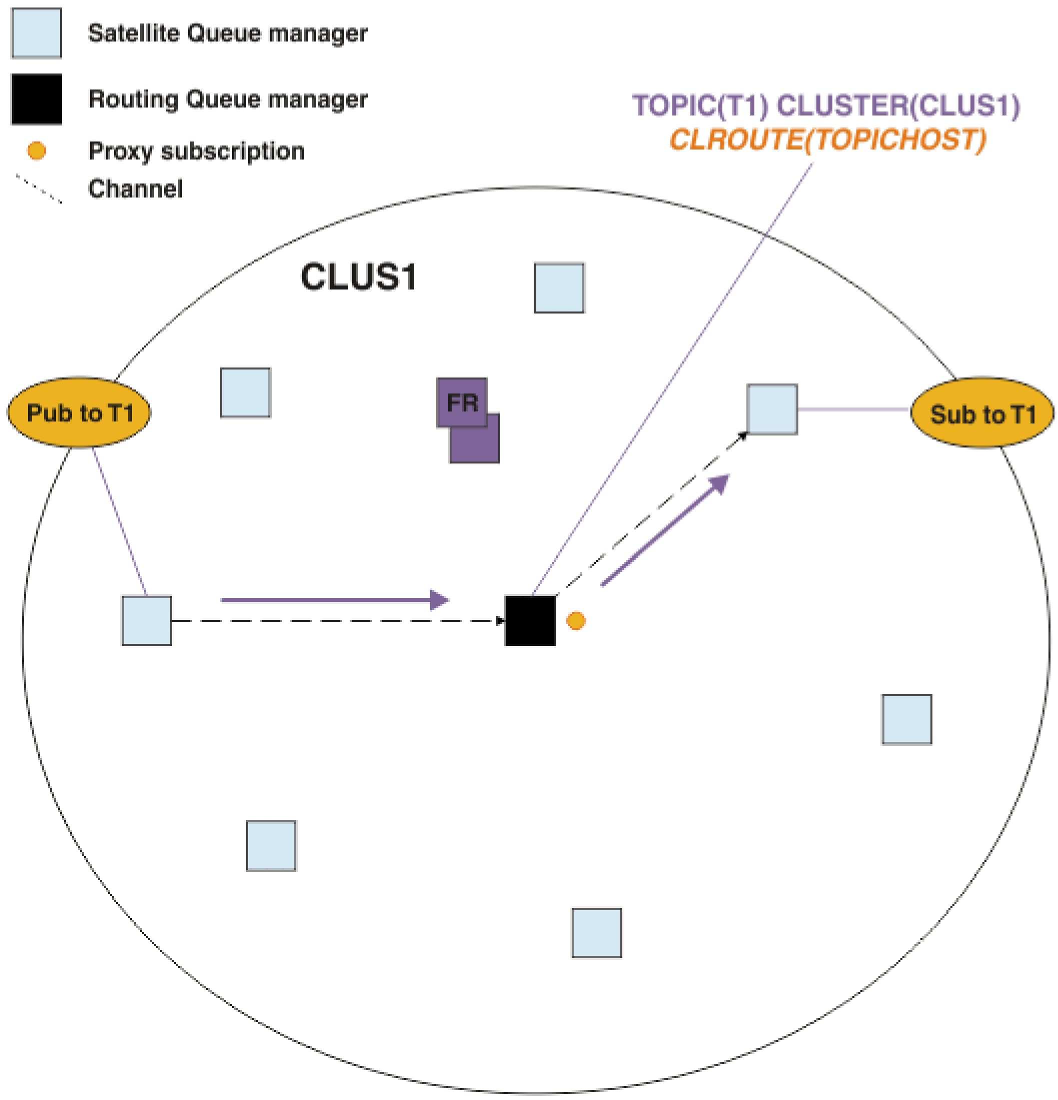


图 *23:* 具有一个主题，一个订户和一个发布者的主题主机路由发布/预订集群

在多个队列管理器之间划分主题树

路由主题主管队列管理器仅负责与配置其受管主题对象的主题树分支相关的预订知识和发布消息。 如果集群中的不同发布/预订应用程序使用不同的主题，那么可以配置不同的队列管理器来托管主题树的不同集群分支。 这允许通过减少集群中每个主题主机队列管理器上的发布流量，预订知识和通道来进行扩展。 您应该将此方法用于主题树的不同高容量分支:

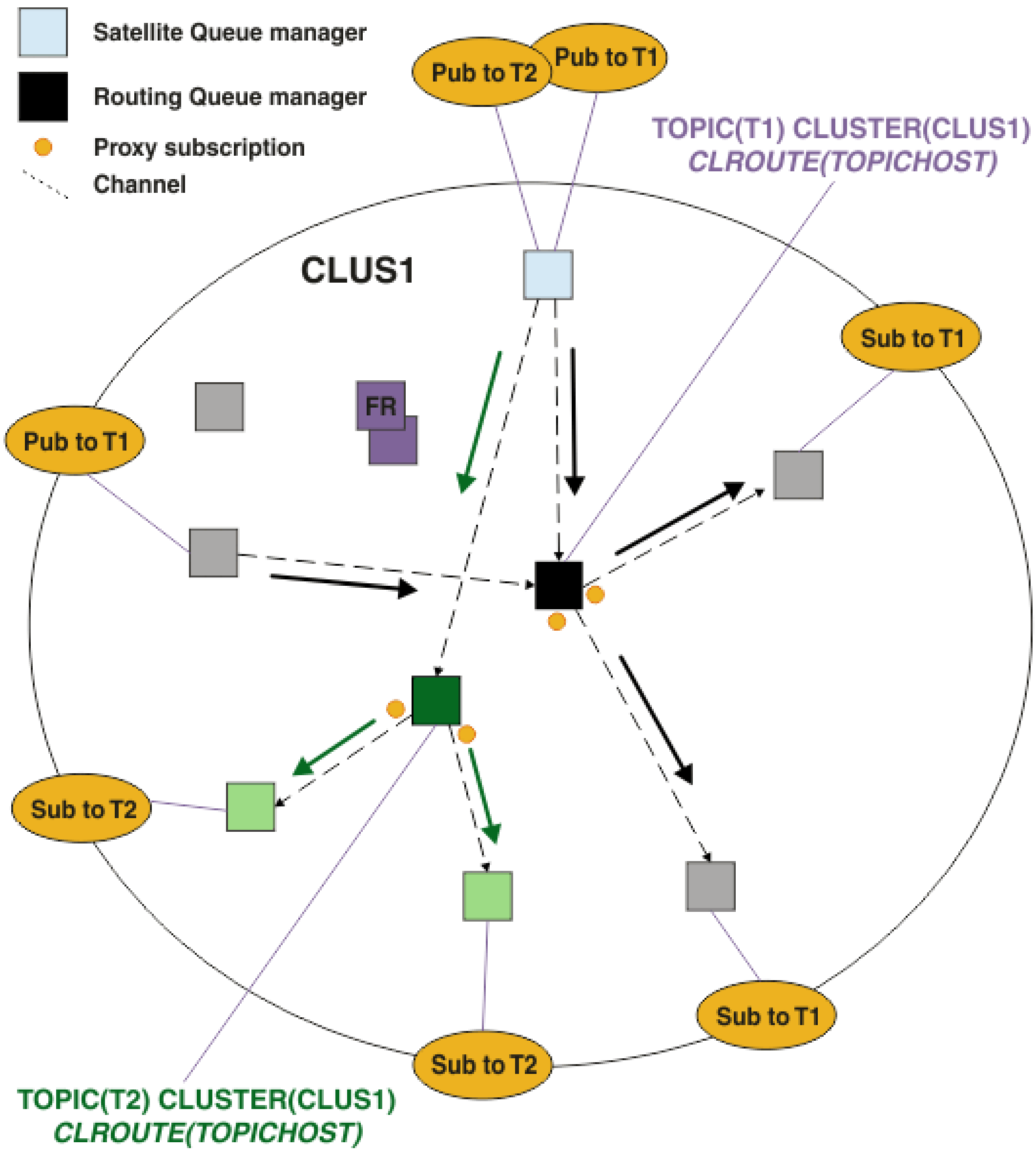


图 *24:* 主题主机路由的发布/预订集群，包含两个主题，每个主题在一个主题主机上定义

例如，使用 主题树中描述的主题，如果主题 T1 配置了主题字符串 /USA/Alabama，而主题 T2 配置了主题字符串 /USA/Alaska，那么将通过托管 T1 的队列管理器来路由发布到 /USA/Alabama/Mobile 的消息。 并且将通过托管 T2 的队列管理器来路由发布到 /USA/Alaska/Juneau 的消息。

注: 通过在主题树中使用比集群点更高的通配符，无法使单个预订跨主题树的多个集群分支。 请参阅 通配符预订。

使用单个主题的多个主题主机进行主题主机路由

如果单个队列管理器负责主题的路由，并且该队列管理器变得不可用或无法处理工作负载，那么发布将不会迅速流向预订。

如果您需要比仅在一个队列管理器上定义主题时获得更大的弹性，可伸缩性和工作负载均衡，那么可以在多个队列管理器上定义主题。 发布的每条消息都通过单个主题主机进行路由。 存在多个匹配的主题主机定义时，将选择其中一个主题主机。 所作的选择与对集群队列的选择相同。 这允许将消息路由到可用的主题主机，从而避免任何不可用的消息，并允许在多个主题主机队列管理器和通道之间均衡消息负载。 但是，当您将多个主题主机用于集群中的同一主题时，不会维护跨多条消息的排序。

下图显示了在两个队列管理器上定义了相同主题的主题主机路由集群。 在此示例中，预订队列管理器以代理预订的形式向两个主题主机队列管理器发送有关预订主题的信息:

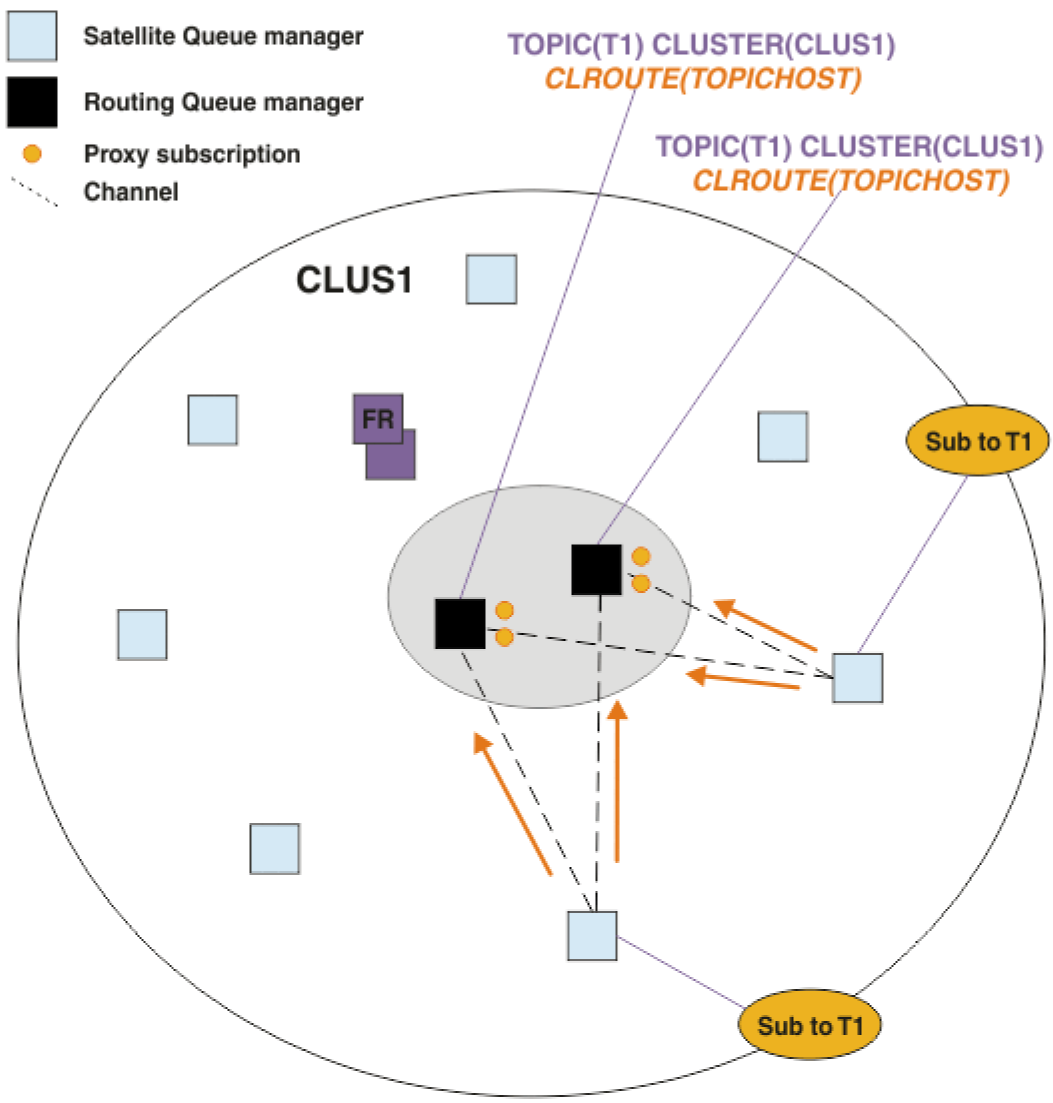


图 *25:* 在多主题主机发布/预订集群中创建代理预订

从非主管队列管理器进行发布时，队列管理器会将发布的副本发送到该主题的主题主机队列管理器的 一个 。 系统根据 集群工作负载管理算法的缺省行为来选择主机。 在典型系统中，这近似于每个主题主机队列管理器之间的循环分布。 来自同一发布应用程序的消息之间没有亲缘关系; 这等同于使用集群绑定类型 NOTFIXED。

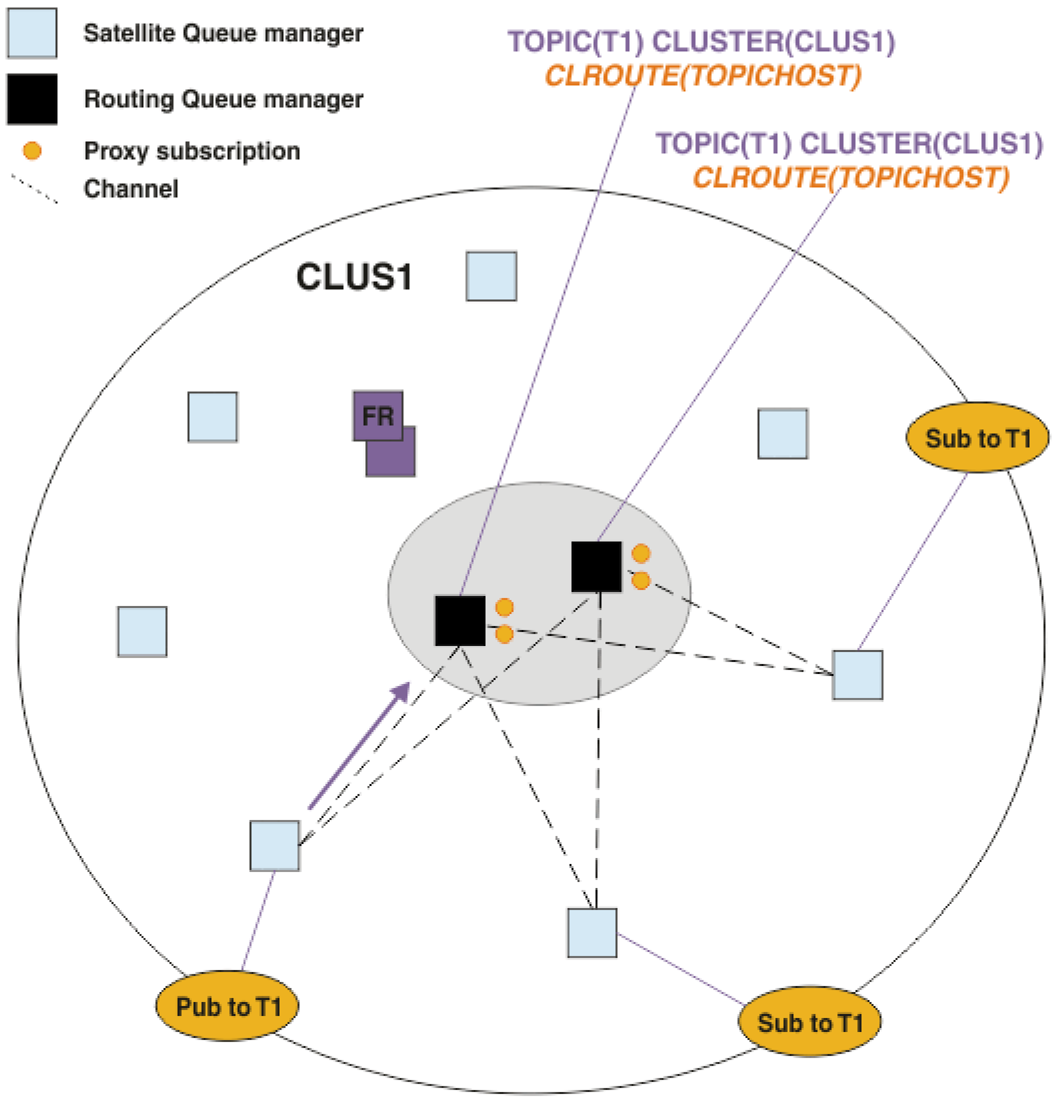


图 *26:* 在多主题主机发布/预订集群中接收发布

然后，将所选主题主机队列管理器的入站发布转发到已注册匹配代理预订的所有队列管理器:

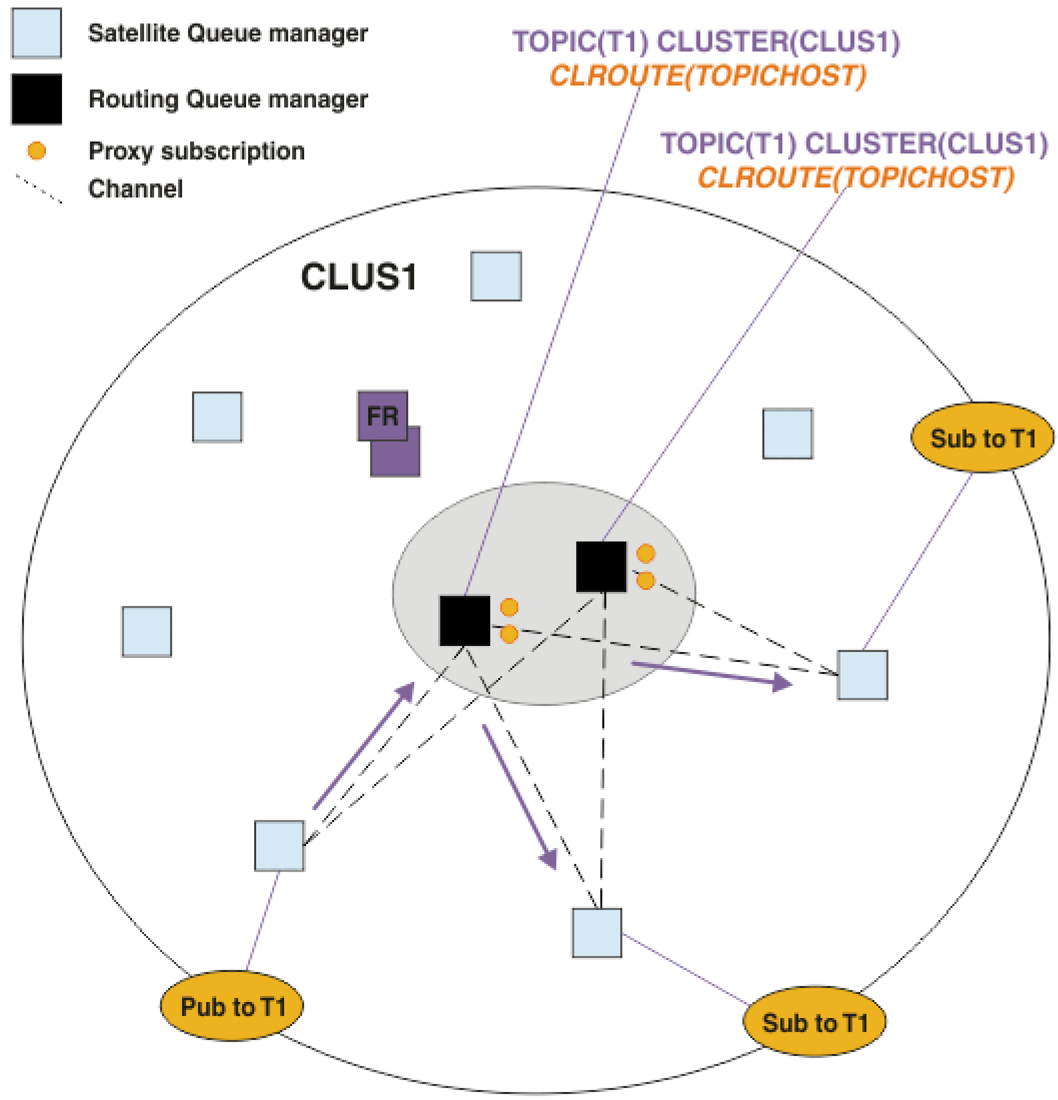


图 *27:* 将发布路由到多主题主机发布/预订集群中的订户

使预订和发布程序成为主题主机队列管理器的本地预订和发布程序

以上示例显示了未托管受管路由主题对象的队列管理器上的发布者和订户之间的路由。 在这些拓扑中，消息需要多个 中继段 才能访问预订。

如果不需要额外的中继段，那么可能需要将密钥发布程序连接到主题托管队列管理器。 但是，如果一个主题有多个主题主机，并且只有一个发布程序，那么所有发布程序流量都将通过发布程序所连接的主题主机队列管理器进行路由。

同样，如果存在密钥预订，那么这些预订可能位于主题主机队列管理器上。 但是，如果路由主题有多个主机，那么只有一部分发布内容将避免额外的中继段，其余部分将首先通过其他主题主机队列管理器进行路由。

此处进一步描述了这些拓扑: 使用集中发布程序或订户进行主题主机路由。

注: 如果在将发布程序或预订与路由主题主机共存时更改路由主题配置，那么需要进行特殊规划。 例如，请参阅 向主题主机路由集群添加额外主题主机。

摘要和其他注意事项

主题主机路由的发布/预订集群使您能够精确控制哪些队列管理器托管每个主题，这些队列管理器将成为主题树的该分支的 路由 队列管理器。 此外，没有预订或发布者的队列管理器无需与主题主机队列管理器连接,

而有预订的队列管理器无需与不托管主题的队列管理器连接。 此配置可以显着减少集群中队列管理器之间的连接数以及在队列管理器之间传递的信息量。 在只有一部分队列管理器在执行发布/预订工作的大型集群中尤其如此。 此配置还使您能够对集群中各个队列管理器上的负载进行一些控制，因此 (例如) 您可以选择在功能更强大且弹性更强的系统上托管高度活动的主题。 对于某些配置 (尤其是较大的集群) ，它通常是比 直接路由更合适的拓扑。

但是，主题主机路由还将对您的系统施加一些约束：

- 系统配置和维护需要比直接路由更多的规划。 您需要决定在主题树中要对哪些点建立集群，以及集群中主题定义的位置。
- 与直接路由主题一样，在定义新的主题主机路由主题时，该信息会推送到完整存储库队列管理器，并从中定向到集群的所有成员。 此事件将导致通道从完整存储库启动到集群的每个成员（如果尚未启动）。
- 发布始终会从非主机队列管理器发送至主机队列管理器，即使集群中没有预订也是如此。 因此，通常期望存在预订时，或全局连接和知晓的开销大于额外发布流量的风险时，应使用路由的主题。

  注: 如前所述，使发布者成为主题主机的本地发布者可以降低此风险。
- 在非主机队列管理器上发布的消息不会直接进入托管预订的队列管理器，而是会始终通过主题主机队列管理器进行路由。 此方法可能会增加集群的总开销，并且可能会增加消息等待时间而降低性能。

  注: 如前所述，使预订或发布程序成为主题主机的本地预订或发布程序可降低此风险。
- 使用单一主题主机队列管理器将为发布到某个主题的所有消息引入单一故障点。 您可以通过定义多个主题主机来移除此单一故障点。 然而，拥有多个主机将影响预订收到已发布消息的顺序。
- 主题主机队列管理器将产生额外消息负载，因为它们需要处理来自多个队列管理器的发布流量。 可以减轻此负载：将多个主题主机用于单个主题（在此情况下，不会维护消息顺序），或使用不同的队列管理器来托管主题树的不同分支的路由主题。

在使用主题主机路由之前，请探索 第 64 页的『发布/预订集群中的直接路由』和 第 89 页的『在发布/预订层次结构中进行路由』中详细描述的替代方法。

发布/预订集群: 最佳实践

使用集群主题使在队列管理器之间扩展发布/预订域变得简单，但如果无法完全理解机制和影响，可能会导致问题。 有两个模型用于信息共享和发布路由。 实施最能满足个人业务需求的模型，并在所选集群上执行最佳操作。

以下部分中的最佳实践信息并不提供一刀切的解决方案，而是共享用于解决常见问题的常见方法。 它假定您基本了解 IBM MQ 集群以及发布/预订消息传递，并且您熟悉 分布式发布/预订网络 和 第 63 页的『设计发布/预订集群』中的信息。

当您将集群用于点到点消息传递时，集群中的每个队列管理器都将以需要知道的方式工作。 即，仅当连接到集群的应用程序请求使用其他集群资源 (例如，集群中的其他队列管理器和集群队列) 时，它才会发现这些资源。 将发布/预订消息传递添加到集群时，将引入更多级别的信息共享以及集群队列管理器之间的连接。 为了能够遵循发布/预订集群的最佳实践，您需要充分了解这种行为变化的影响。

为了使您能够根据您的精确需求构建最佳体系结构，在发布/预订集群中有两个用于信息共享和发布路由的模型: 直接路由 和 主题主机路由。 要做出正确的选择，您需要了解两个模型，以及每个模型满足的不同需求。 以下部分与 第 60 页的『规划分布式发布/预订网络』一起讨论了这些需求:

- 第 78 页的『限制发布/预订活动中涉及的集群队列管理器数的原因』
- 第 79 页的『如何确定要集群的主题』
- 第 79 页的『如何调整系统大小』
- 第 80 页的『发布者和预订位置』
- 第 80 页的『发布流量』
- 第 81 页的『预订更改和动态主题字符串』

限制发布/预订活动中涉及的集群队列管理器数的原因

在集群中使用发布/预订消息传递时，存在容量和性能注意事项。 因此，最好仔细考虑跨队列管理器的发布/预订活动的需求，并将其限制为仅需要该活动的队列管理器的数量。 在确定需要发布和预订主题的最小队列管理器集合后，可以使这些队列管理器成为仅包含它们且不包含其他队列管理器的集群的成员。

如果已建立的集群已正常用于点到点消息传递，那么此方法特别有用。 将现有大型集群转换为发布/预订集群时，最好先为发布/预订工作创建单独的集群，在此工作中可以尝试使用应用程序，而不是使用当前集群。 您可以使用已存在于一个或多个点到点集群中的现有队列管理器子集，并使此子集成为新的发布/预订集群的成员。 但是，新集群的完整存储库队列管理器不得是任何其他集群的成员; 这将隔离现有集群完整存储库中的额外负载。

如果无法创建新集群，并且必须将现有大型集群转换为发布/预订集群，请不要使用直接路由模型。 主题主机路由模型通常在更大的集群中表现更好，因为它通常限制发布/预订信息共享以及与正在主动执行发布/预订工作的队列管理器集合的连接，集中在托管主题的队列管理器上。 此情况的例外情况是，在托管主题定义的队列管理器上调用了预订信息的手动刷新，此时主题主机队列管理器将连接到集群中的每个队列管理器。 请参阅 代理预订的再同步。

如果您确定集群由于其大小或当前负载而无法用于发布/预订，那么最好防止此集群意外变为发布/预订集群。 使用 **PSCLUS** 队列管理器属性可阻止任何人在集群中的任何队列管理器上添加集群主题。 请参阅 第 85 页的『禁止集群发布/预订』。

如何确定要集群的主题

请务必仔细选择将哪些主题添加到集群中: 这些主题的主题树越高，其使用范围就越广。 这可能会导致传播的预订信息和发布数超过必需的数量。 如果主题树有多个不同的分支 (其中一些需要集群，而另一些不需要集群) ，请在需要集群的每个分支的根上创建受管主题对象，并将这些对象添加到集群中。 例如，如果分支 /A， /B 和 /C 需要集群，请为每个分支定义单独的集群主题对象。

注: 系统会阻止您在主题树中嵌套集群主题定义。 仅允许您在每个子分支的主题树中的一个点上对主题进行聚类。 例如，无法为 /A 和 /A/B 定义集群主题对象。 嵌套集群主题可能会导致混淆哪个集群对象适用于哪个预订，尤其是在预订使用通配符时。 这在使用主题主机路由时更为重要，其中路由决策由主题主机的分配精确定义。

如果必须将集群主题添加到主题树的高处，但集群点以下的树的某些分支不需要集群行为，那么可以使用预订和发布范围属性来降低预订和发布共享级别以用于更多主题。

如果不考虑所看到的行为，那么不应将主题根节点放入集群中。 尽可能使全局主题变得明显，例如，在主题字符串中使用高级限定符: /global 或 /cluster。

还有一个原因是不想使根主题节点集群化。 这是因为每个队列管理器都具有根节点 SYSTEM.BASE.TOPIC 主题对象的本地定义。 当此对象在集群中的一个队列管理器上进行集群时，将使所有其他队列管理器了解此对象。 但是，当存在同一对象的本地定义时，其属性将覆盖集群对象。 这将导致这些队列管理器像未对主题进行集群一样工作。 要解决此问题，您需要对 SYSTEM.BASE.TOPIC 的每个定义进行集群。 您可以对直接路由定义执行此操作，但不能对主题主机路由定义执行此操作，因为这会导致每个队列管理器都成为主题主机。

如何调整系统大小

发布/预订集群通常导致集群通道模式与集群中的点到点消息传递模式不同。 点对点模型是 "选择性加入" 模型，但发布/预订集群具有更滥用的预订扇出性质，尤其是在使用直接路由主题时。 因此，请务必确定发布/预订集群中的哪些队列管理器将使用集群通道连接到其他队列管理器，以及在何种情况下连接到其他队列管理器。

下表列出了在正常运行的发布/预订集群中每个队列管理器期望的集群发送方和接收方通道的典型集合，这取决于发布/预订集群中的队列管理器角色。

表 *5:* 每个路由方法的集群发送方和接收方通道。

队列管理器角色	直接集群接收方	直接集群发送方	主题集群接收方	主题集群发送方
完整存储库	AllQmgrs	AllQmgrs	AllQmgrs	AllQMgrs
主题定义的主机	不适用	不适用	AllSubs+AllPubs (1)	AllSubs (1)
已创建预订	AllPubs (1)	AllQMgrs	AllHosts	AllHosts
已连接发布程序	AllSubs (1)	AllSubs (1)	AllHosts	AllHosts
无发布者或订户	AllSubs (1)	无 (1)	无 (2)	无 (2)

关键字:

AllQmgrs
这是与集群中的每个队列管理器之间的通道。

AllSubs
与已创建预订的每个队列管理器之间的通道。

AllPubs
与已连接发布应用程序的每个队列管理器之间的通道。

AllHosts
在其中配置了集群主题对象定义的每个队列管理器之间的通道。

None
仅出于发布/预订消息传递的目的，集群中没有与其他队列管理器之间的通道。

注意:

1. 如果从此队列管理器刷新代理预订，那么可能会自动创建与集群中所有其他队列管理器之间的通道。
2. 如果从此队列管理器刷新代理预订，那么可能会自动创建用于托管集群主题定义的集群中任何其他队列管理器的通道。

上表显示，与直接路由相比，主题主机路由通常使用的集群发送方和接收方通道要少得多。 如果通道连接是集群中某些队列管理器所关注的问题，那么出于容量或建立特定通道 (例如，通过防火墙) 的能力的原因，主题主机路由是首选解决方案。

发布者和预订位置

集群发布/预订使在一个队列管理器上发布的消息能够传递到集群中任何其他队列管理器上的预订。 对于点到点消息传递，在队列管理器之间传输消息的成本可能对性能不利。 因此，您应该考虑在发布消息的队列管理器上创建对主题的预订。

在集群中使用主题主机路由时，还应考虑与主题托管队列管理器相关的预订和发布程序的位置。 当发布者未连接到作为集群主题的主机的队列管理器时，发布的消息将始终发送到主题托管队列管理器。 同样，在不是集群主题的主题主机的队列管理器上创建预订时，从集群中的其他队列管理器发布的消息总是首先发送到主题主管队列管理器。 更具体地说，如果预订位于托管该主题的队列管理器上，但有一个或多个其他队列管理器也托管同一主题，那么来自其他队列管理器的部分发布将通过这些其他主题托管队列管理器进行路由。 请参阅 使用集中发布程序或订户的主题主机路由 ，以获取有关设计主题主机路由发布/预订集群以最大限度缩短发布程序与预订之间的距离的更多信息。

发布流量

由连接到集群中的一个队列管理器的应用程序发布的消息将使用集群发送方通道传输到其他队列管理器上的预订。

使用直接路由时，发布的消息在队列管理器之间采用最短路径。 即，它们直接从发布队列管理器到具有预订的每个队列管理器。 不会将消息传输到没有主题预订的队列管理器。 请参阅 发布/预订网络中的代理预订。

如果集群中任何一个队列管理器与另一个队列管理器之间的发布消息速率较高，那么这两个点之间的集群通道基础结构必须能够保持该速率。 这可能涉及调整正在使用的通道和传输队列。

当您使用主题主机路由时，在不是主题主机的队列管理器上发布的每条消息都会传输到主题主机队列管理器。 这与集群中其他位置是否存在一个或多个预订无关。 这引入了在规划时要考虑的其他因素:

- 首次向主题主机队列管理器发送每个发布的额外等待时间是否可接受?
- 每个主题主机队列管理器能否维持入站和出站发布速率? 请考虑一个具有许多不同队列管理器上的发布程序的系统。 如果它们都将其消息发送到一组非常小的主题托管队列管理器，那么这些主题主机可能会成为处理这些消息并将其路由到预订队列管理器的瓶颈。
- 是否预期相当一部分已发布的消息将没有匹配的订户? 如果是这样，并且发布此类消息的速率很高，那么最好使发布者的队列管理器成为主题主机。 在此情况下，不会将集群中不存在任何预订的任何已发布消息传输到任何其他队列管理器。

还可以通过引入多个主题主机来缓解这些问题，以在它们之间传播发布负载:

- 如果有多个不同的主题 (每个主题都有一定比例的发布流量) ，请考虑将它们托管在不同的队列管理器上。
- 如果无法将主题分隔到不同的主题主机上，请考虑在多个队列管理器上定义相同的主题对象。 这将导致出版物的工作负载在每个出版物之间进行均衡，以便进行路由。 但是，仅当不需要发布消息排序时，此操作才适用。

预订更改和动态主题字符串

另一个考虑因素是对传播代理预订的系统性能的影响。 通常，当在该队列管理器上创建特定集群主题字符串 (而不仅仅是配置的主题对象) 的第一个预订时，队列管理器会将代理预订消息发送到集群中的某些其他队列管理器。 同样，在删除特定集群主题字符串的最后一个预订时，将发送代理预订删除消息。

对于直接路由，具有预订的每个队列管理器都会将这些代理预订发送到集群中的每个其他队列管理器。 对于主题主机路由，具有预订的每个队列管理器仅将代理预订发送到主管该集群主题的定义的每个队列管理器。 因此，使用直接路由时，集群中的队列管理器越多，跨这些队列管理器维护代理预订的开销就越高。 而对于主题主机路由，集群中的队列管理器数不是一个因素。

在这两种路由模型中，如果发布/预订解决方案由许多要预订的唯一主题字符串组成，或者集群中的队列管理器上的主题经常被预订和取消预订，那么将在该队列管理器上看到由于不断生成消息分发和删除代理预订而导致的大量开销。 通过直接路由，由于需要将这些消息发送到集群中的每个队列管理器，这就更加复杂了。

如果预订的更改速率过高而无法容纳 (即使在主题主机路由系统中) ，请参阅 发布/预订网络中的预订性能 以获取有关减少代理预订开销的方法的信息。

定义集群主题

集群主题是定义了 **cluster** 属性的管理主题。 有关集群主题的信息会推送给集群的所有成员，并与本地主题相结合以创建横跨多个队列管理器的主题空间的一些部分。 这支持将一个队列管理器中的某个主题上发布的消息传递给集群中其他队列管理器的预订。

在队列管理器上定义集群主题时，会将集群主题定义发送到完整存储库队列管理器。 然后，完整存储库会将集群主题定义传播到集群中的所有队列管理器，从而使相同的集群主题可用于集群中任何队列管理器处的发布者和订户。 您在其中创建集群主题的队列管理器称为集群主题主机。 集群主题可供集群中的任何队列管理器使用，但是对集群主题的任何修改必须在定义该主题的队列管理器（主机）上执行，此时会通过完整存储库将修改传播到集群的所有成员。

使用直接路由时，集群主题定义的位置不会直接影响系统的行为，因为集群中的所有队列管理器都以相同方式使用主题定义。 因此，您应该在任何将成为集群成员的队列管理器上定义主题，只要需要该主题，并且该主题位于足够可靠的系统上，以便定期与完整存储库队列管理器联系。

使用主题主机路由时，集群主题定义的位置非常重要，因为集群中的其他队列管理器会创建此队列管理器的通道，并向其发送预订信息和发布。 要选择托管主题定义的最佳队列管理器，您需要了解主题主机路由。 请参阅 第 69 页的『发布/预订集群中的主题主机路由』。

如果您具有集群主题和本地主题对象，那么本地主题优先。 请参阅 第 83 页的『具有相同名称的多个集群主题定义』。

有关用于显示集群主题的命令的信息，请参阅相关信息。

集群主题继承

通常，在集群发布/预订拓扑中发布和预订应用程序期望工作方式相同，无论它们连接到集群中的哪个队列管理器。 这就是集群管理的主题对象传播到集群中每个队列管理器的原因。

受管主题对象从主题树中更高的其他受管主题对象继承其行为。 当没有为主题参数设置显式值时，将发生此继承。

在集群发布/预订的情况下，考虑此类继承很重要，因为这会引入发布者和订户根据其连接到的队列管理器的不同行为方式的可能性。 如果集群主题对象保留从更高主题对象继承的任何参数，那么该主题在集群中的不同队列管理器上的行为可能不同。 类似地，在主题树中的集群主题对象下定义的本地定义主题对象将意味着那些较低级别的主题仍处于集群状态，但本地对象可能会以与集群中的其他队列管理器不同的方式更改其行为。

通配符预订

当对在集群主题对象处或下方解析的主题字符串进行本地预订时，将创建代理预订。 如果在主题层次结构中进行的通配符预订高于任何集群主题，那么不会在集群周围为匹配的集群主题发送代理预订，因此不会从集群的其他成员接收任何发布。 但是，它会从本地队列管理器接收发布。

但是，如果另一个应用程序预订解析为集群主题或低于集群主题的主题字符串，那么将生成代理预订并将发布内容传播到此队列管理器。 原件到达时，更高的通配符预订将被视为这些出版物的合法接收方并接收副本。 如果不需要此行为，请在集群主题上设置 **WILDCARD(BLOCK)** 。 这将使原始通配符不被视为合法预订，并使其停止接收集群主题或其子主题上的任何发布 (本地发布或来自集群中的其他位置)。

相关概念

处理管理主题

使用预订

相关参考

DISPLAY TOPIC

DISPLAY TPSTATUS

显示子项

集群主题属性

当主题对象设置了集群名称属性时，将在集群中的所有队列管理器之间传播主题定义。 每个队列管理器都使用传播的主题属性来控制发布/预订应用程序的行为。

主题对象具有许多适用于发布/预订集群的属性。 部分控制发布和预订应用程序的一般行为，部分控制如何在集群中使用主题。

必须以集群中所有队列管理器都可以正确使用集群主题对象定义的方式进行配置。

例如，如果要用于受管预订 (MDURMDL 和 MNDURMDL) 的模型队列 设置为非缺省队列名称，必须在将创建受管预订的所有队列管理器上定义指定的模型队列。

同样，如果任何属性设置为 ASPARENT，那么主题的行为将依赖于主题树中的较高节点 (请参阅 管理主题对象) 在集群中的每个单独队列管理器上。 在从不同的队列管理器发布或预订时，这可能会导致不同的行为。

与集群中的发布/预订行为直接相关的主要属性如下所示:

CLROUTE

此参数控制连接了发布程序的队列管理器与存在匹配预订的队列管理器之间的消息路由。

- 您可以将路由配置为在这些队列管理器之间直接执行，或者通过托管集群主题定义的队列管理器执行。 有关更多详细信息，请参阅 发布/预订集群 。
- 在设置 **CLUSTER** 参数时，无法更改 **CLROUTE** 。 要更改 **CLROUTE**，请首先将 **CLUSTER** 属性设置为空白。 这将阻止使用该主题的应用程序以集群方式执行操作。 这反过来会导致将发布传递到预订的中断，因此您在进行更改时也应该停顿发布/预订消息传递。

PROXYSUB

此参数控制何时进行代理预订。

- FIRSTUSE 是缺省值，会导致发送代理预订以响应分布式发布/预订拓扑中队列管理器上的本地预订，并在不再需要时取消。 有关您可能希望将此属性从缺省值 FIRSTUSE 更改为缺省值的原因的详细信息，请参阅 各个代理预订转发和 处处发布 。
- 要启用 所有位置发布，请将高级主题对象的 **PROXYSUB** 参数设置为 FORCE 。 这将导致单个通配符代理预订与主题树中此主题对象下的所有主题匹配。

注: 在大型或繁忙的发布/预订集群中设置 **PROXYSUB(FORCE)** 属性可能会导致系统资源负载过高。 **PROXYSUB(FORCE)** 属性将传播到每个队列管理器，而不仅仅是定义主题的队列管理器。 这将导致集群中的每个队列管理器创建通配符代理预订。

在集群中的任何队列管理器上发布的此主题的消息副本将直接或通过主题主机队列管理器 (具体取决于 **CLROUTE** 设置) 发送到集群中的每个队列管理器。

当直接路由主题时，每个队列管理器都会创建集群发送方通道到其他每个队列管理器。 当主题是主题主机路由时，将从集群中的每个队列管理器创建到每个主题主机队列管理器的通道。

有关在集群中使用 **PROXYSUB** 参数的更多信息，请参阅 直接路由发布/预订性能。

PUBSCOB 和 SUBSCOPE

这些参数确定此队列管理器是将发布传播到拓扑 (发布/预订集群或层次结构) 中的队列管理器，还是将作用域限制为仅其本地队列管理器。 您可以使用 MQPMO_SCOPE_QMGR 和 MQSO_SCOPE_QMGR 以编程方式执行等效作业。

PUBSCOPE

如果使用 **PUBSCOPE(QMGR)**定义了集群主题对象，那么该定义将与集群共享，但基于该主题的发布的作用域仅为本地，并且不会将其发送到集群中的其他队列管理器。

SUBSCOPE

如果使用 **SUBSCOPE(QMGR)**定义了集群主题对象，那么该定义将与集群共享，但基于该主题的预订的作用域仅为本地，因此不会将代理预订发送到集群中的其他队列管理器。

这两个属性通常一起用于隔离队列管理器，使其不与特定主题上的集群其他成员进行交互。 队列管理器既不向集群的其他成员发布这些主题的出版物，也不从集群的其他成员接收这些主题的出版物。 如果在子主题上定义了主题对象，那么此情境不会阻止发布或预订。

在主题的本地定义上将 **SUBSCOPE** 设置为 QMGR 不会阻止集群中的其他队列管理器将其代理预订传播到队列管理器 (如果他们正在将该主题的集群版本与 **SUBSCOPE(ALL)**配合使用)。 但是，如果本地定义还将 **PUBSCOPE** 设置为 QMGR ，那么不会从此队列管理器向这些代理预订发送发布。

相关概念

发布作用域

预订作用域

具有相同名称的多个集群主题定义

您可以在集群中的多个队列管理器上定义相同的指定集群主题对象，在某些情况下，这将启用特定行为。 当存在多个同名的集群主题定义时，大多数属性应该匹配。 如果不匹配，那么将根据不匹配的重要性来报告错误或警告。

通常，如果多个集群主题定义的属性不匹配，那么将发出警告，并且集群中的每个队列管理器将使用其中一个主题对象定义。 每个队列管理器使用的定义不是确定性的，或者在集群中的队列管理器之间是一致的。 这种不匹配应该尽快解决。

在集群设置或维护期间，有时需要创建多个不相同的集群主题定义。 但是，这仅作为临时措施有用，因此被视为潜在的错误情况。

检测到不匹配时，会将以下警告消息写入每个队列管理器的错误日志:

- Multi 在 多平台 上，AMQ9465 和 AMQ9466。
- z/OS 在 z/OS 上，CSQX465I 和 CSQX466I。

可以通过查看主题状态而不是主题对象定义 (例如，使用 **DISPLAY TPSTATUS**) 来确定每个队列管理器上任何主题字符串的所选属性。

在某些情况下，配置属性中的冲突严重到足以停止正在创建的主题对象，或者导致不匹配的对象被标记为无效且未在集群中传播 (请参阅 DISPLAY TOPIC 中的 **CLSTATE**)。 当集群路由属性 (**CLROUTE**) 中存在冲突时，会发生这些情况 主题定义。 此外，由于主题主机路由定义之间一致性的重要性，将拒绝进一步的不一致，如本文的后续部分中所述。

如果在定义对象时检测到冲突，那么将拒绝配置更改。 如果稍后由完整存储库队列管理器检测到，那么会将以下警告消息写入队列管理器错误日志:

- Multi 在 多平台上: AMQ9879
- z/OS 在 z/OS 上: CSQX879E。

在集群中定义同一主题对象的多个定义时，本地定义的定义优先于任何远程定义的定义。 因此，如果定义中存在任何差异，那么托管多个定义的队列管理器的行为将彼此不同。

定义与另一个队列管理器中的集群主题同名的非集群主题的影响

可以定义未在集群中的队列管理器上集群的受管主题对象，并同时定义与其他队列管理器上的集群主题定义相同的指定主题对象。 在这种情况下，本地定义的主题对象优先于所有同名的远程定义。

这将在从此队列管理器中使用时阻止主题的集群行为。 即，预订可能不会接收来自远程发布程序的发布，并且来自发布程序的消息可能不会传播到集群中的远程预订。

在配置此类系统之前应仔细考虑，因为这可能会导致混淆行为。

注: 如果单个队列管理器需要阻止发布和预订在集群中传播，即使主题已在其他位置进行集群，另一种方法是将发布和预订作用域设置为仅本地队列管理器。 请参阅 第 82 页的『集群主题属性』。

直接路由集群中的多个集群主题定义

对于直接路由，通常不会在多个集群队列管理器上定义相同的集群主题。 这是因为直接路由使该主题在集群中的所有队列管理器上都可用，无论该主题是在哪个队列管理器上定义的。 此外，添加多个集群主题定义会显着增加系统活动和管理复杂性，随着复杂性的增加，出现人为错误的可能性会更大:

- 每个定义都会导致将另一个集群主题对象推送到集群中的其他队列管理器 (包括其他集群主题主机队列管理器)。
- 集群中特定主题的所有定义必须相同，否则难以确定队列管理器正在使用哪个主题定义。

由于集群主题定义由完整的存储库队列管理器以及其部分集群存储库中的所有其他队列管理器进行高速缓存，因此唯一的主机队列管理器对于该主题在整个集群中正常运行并不重要。 有关更多信息，请参阅 使用直接路由的主题主机队列管理器的可用性。

对于可能需要在第二个队列管理器上临时定义集群主题的情况，例如，要从集群中除去该主题的现有主机时，请参阅 将集群主题定义移至其他队列管理器。

如果您需要更改集群主题定义，请在定义该集群主题定义的队列管理器上谨慎进行修改。 尝试从另一个队列管理器对其进行修改时，可能会意外地创建具有冲突主题属性的主题的第二个定义。

主题主机路由集群中的多个集群主题定义

当使用 主题主机的集群路由定义集群主题时，该主题将在集群中的所有队列管理器之间传播，就像针对 直接 路由的主题一样。 此外，该主题的所有发布/预订消息传递都将通过定义该主题的队列管理器进行路由。 因此，集群中主题的位置和定义数变得很重要 (请参阅 第 69 页的『发布/预订集群中的主题主机路由』)。

为了确保足够的可用性和可伸缩性，如果可能，具有多个主题定义是很有用的。 请参阅 使用主题主机路由的主题主机队列管理器的可用性。

在集群中添加或除去 主题主机 路由主题的其他定义时，应考虑配置更改时的消息流。 如果在更改时集群中正在将消息发布到主题，那么需要一个登台过程来添加或除去主题定义。 请参阅 将集群主题定义移动到其他队列管理器 和 将额外的主题主机添加到主题主机路由的集群。

如前所述，多个定义的属性应该匹配，但 **PUB** 参数可能例外，如下一节中所述。 当通过主题主机队列管理器路由发布时，更重要的是使多个定义保持一致。 因此，如果为主题主机集群路由配置了一个或多个主题定义，那么将拒绝在主题字符串或集群名称中检测到的不一致。

注: 如果尝试在主题树中的另一个主题上方或下方配置集群主题定义，其中配置了现有集群主题定义以进行主题主机路由，那么也会拒绝集群主题定义。 这可防止在与通配符预订相关的发布路由中出现歧义。

PUB 参数的特殊处理

PUB 参数用于控制应用程序何时可以发布到主题。 对于集群中的主题主机路由，它还可以控制使用哪些主题主机队列管理器来路由发布。 因此，允许它在集群中具有同一主题对象的多个定义，以及 PUB 参数的不同设置。

如果主题的多个远程集群定义具有此参数的不同设置，那么当满足以下条件时，该主题允许将发布发送并传递到预订:

- 在将发布程序连接到的队列管理器上未定义匹配的主题对象，该对象设置为 `PUB(DISABLED)`。

- 集群中的一个或多个多主题定义设置为 PUB(ENABLED)，或者一个或多个多主题定义设置为 PUB(ASPARENT)，并且在主题树中的较高位置将连接发布程序和定义的预订的本地队列管理器设置为 PUB(ENABLED)。

对于主题主机路由，当消息由连接到不是主题主机的队列管理器的应用程序发布时，仅将消息路由到未显式将 **PUB** 参数设置为 DISABLED 的主题托管队列管理器。 因此，您可以使用 PUB(DISABLED) 设置来停顿通过特定主题主机的消息流量。 您可能希望执行此操作以准备维护或除去队列管理器，或者出于 向主题主机路由集群添加额外主题主机中描述的原因。

集群主题主机队列管理器的可用性
设计发布/预订集群以将主题主机队列管理器变为不可用时集群将无法处理主题流量的风险降至最低。 主题主机队列管理器变为不可用的影响取决于集群是使用主题主机路由还是直接路由。

使用直接路由的主题主机队列管理器的可用性

对于直接路由，通常不会在多个集群队列管理器上定义相同的集群主题。 这是因为直接路由使该主题在集群中的所有队列管理器上都可用，无论该主题是在哪个队列管理器上定义的。请参阅 直接路由集群中的多个集群主题定义。

在集群中，每当集群对象 (例如，集群队列或集群主题) 的主机长时间不可用时，集群的其他成员将最终使这些对象的知识到期。 对于集群主题，如果集群主题主机队列管理器变为不可用，那么其他队列管理器将继续以直接集群方式处理主题的发布/预订请求 (即，向远程队列管理器上的预订发送发布)，至少从主题托管队列管理器上次与完整存储库队列管理器进行通信的时间开始 60 天。 如果定义了集群主题对象的队列管理器不再可用，那么最终将删除其他队列管理器上的高速缓存主题对象，并且该主题将还原为本地主题，在这种情况下，预订将停止接收来自连接到远程队列管理器的应用程序的发布。

使用 60 天时间段来恢复您在其中定义集群主题对象的队列管理器，几乎不需要采取特殊措施来保证集群主题主机保持可用状态 (但是请注意，在不可用的集群主题主机上定义的任何预订都不会保持可用状态)。 60 天的时间足以迎合技术问题，仅因行政错误就有可能被超过。 为了降低这种可能性，如果集群主题主机不可用，那么集群的所有成员将每小时写入错误日志消息，并声明其高速缓存的集群主题对象未刷新。 通过确保定义了集群主题对象的队列管理器正在运行来响应这些消息。 如果无法使集群主题主机队列管理器再次可用，请在集群中的另一个队列管理器上定义具有完全相同的属性的同一集群主题定义。

使用主题主机路由的主题主机队列管理器的可用性

对于主题主机路由，主题的所有发布/预订消息传递都通过定义该主题的队列管理器进行路由。 因此，考虑集群中这些队列管理器的持续可用性非常重要。 如果主题主机变为不可用，并且该主题不存在其他主机，那么从发布者到集群中不同队列管理器上的订户的流量将立即停止该主题。 如果其他主题主机可用，那么集群队列管理器将通过这些主题主机来路由新的发布流量，从而提供消息路由的持续可用性。

对于直接主题，在 60 天后，如果第一个主题主机仍然不可用，那么将从集群中除去对该主题主机的主题的了解。 如果这是集群中此主题的最后一个剩余定义，那么所有其他队列管理器将停止将发布转发到任何主题主机以进行路由。

因此，为了确保足够的可用性和可伸缩性，如果可能，可以在至少两个集群队列管理器上定义每个主题。 这提供了防止任何给定主题主机队列管理器变为不可用的保护。 另请参阅 主题主机路由集群中的多个集群主题定义。

如果无法配置多个主题主机 (例如，因为需要保留消息排序)，并且无法仅配置一个主题主机 (因为单个队列管理器的可用性不得影响发布到集群中所有队列管理器中的预订的流)，请考虑将该主题配置为直接路由的主题。 这避免了对整个集群的单个队列管理器的依赖，但仍要求每个单独的队列管理器都可用，以便它处理本地托管的预订和发布程序。

禁止集群发布/预订

将第一个直接路由的集群主题引入到集群中会强制集群中的每个队列管理器了解其他每个队列管理器，并可能导致它们相互创建通道。 如果不需要这样做，那么您应该改为配置主题主机路由发布/预订。 如果存在直接路由的集群主题可能会危及集群的稳定性，那么由于每个队列管理器的缩放问题，您可以通过在集群中的每个队列管理器上将 **PSCLUS** 设置为 DISABLED 来完全禁用集群发布/预订功能。

如第 64 页的『发布/预订集群中的直接路由』中所述，当您将直接路由的集群主题引入到集群中时，将自动通知所有部分存储库该集群的所有其他成员。 集群主题还可能在所有其他节点 (例如，指定了

PROXYSUB(FORCE)) 创建预订，并导致从队列管理器启动大量通道，即使没有本地预订也是如此。 这会使集群中的每个队列管理器立即产生额外负载。 对于包含许多队列管理器的集群，这可能会导致性能显着降低。 因此，必须仔细规划对集群的直接路由发布/预订的引入。

当您知道集群无法容纳直接路由的发布/预订的开销时，可以改为使用主题主机路由的发布/预订。 有关差异的概述，请参阅 第 63 页的『设计发布/预订集群』。

如果您希望完全禁用集群的发布/预订功能，那么可以通过在集群中的每个队列管理器上将队列管理器属性 **PSCLUS** 设置为 DISABLED 来执行此操作。 此设置通过修改队列管理器功能的三个方面来禁用集群中的直接路由和主题主机路由发布/预订:

- 此队列管理器的管理员不再能够将 Topic 对象定义为集群对象。
- 将拒绝来自其他队列管理器的入局主题定义或代理预订，并且将记录一条警告消息以通知管理员不正确的配置。
- 当完整存储库接收到主题定义时，它们不再自动与所有其他部分存储库共享有关每个队列管理器的信息。

虽然 **PSCLUS** 是集群中每个单独队列管理器的参数，但它并非旨在选择性地禁用集群中队列管理器子集中的发布/预订。 如果以这种方式选择性地禁用，那么您将看到频繁的错误消息。 这是因为如果在启用了 **PSCLUS** 的队列管理器上集群某个主题，那么会不断看到和拒绝代理预订和主题定义。

因此，您应该在集群中的每个队列管理器上将 **PSCLUS** 设置为 DISABLED 。 但是，在实践中，此状态可能难以实现和维护，例如，队列管理器可以随时加入和离开集群。 至少，必须确保在所有完整存储库队列管理器上将 **PSCLUS** 设置为 DISABLED 。 如果执行此操作，并且随后在集群中的 ENABLED 队列管理器上定义了集群主题，那么这不会导致完整存储库通知每个其他队列管理器的每个队列管理器，因此将保护集群免受所有队列管理器中的潜在缩放问题影响。 在此场景中，将在完整存储库队列管理器的错误日志中报告集群主题的源。

如果队列管理器参与一个或多个发布/预订集群以及一个或多个点到点集群，那么必须在该队列管理器上将 **PSCLUS** 设置为 ENABLED 。 因此，将点到点集群与发布预订集群重叠时，应该在每个集群中使用一组单独的完整存储库。 此方法允许有关每个队列管理器的主题定义和信息仅在发布/预订集群中流动。

在将 **PSCLUS** 从 ENABLED 更改为 DISABLED 时，为了避免配置不一致，此队列管理器所属的任何集群中都不能存在任何集群主题对象。 在将 **PSCLUS** 更改为 DISABLED 之前，必须删除任何此类主题 (即使是远程定义的主题)。

有关 **PSCLUS** 的更多信息，请参阅 ALTER QMGR (PSCLUS)。

相关概念

直接路由的发布/预订集群性能

发布/预订和多个集群

单个队列管理器可以是多个集群的成员。 此排列有时称为 重叠集群。 通过这种重叠，可以从多个集群访问集群队列，并且可以将点到点消息流量从一个集群中的队列管理器路由到另一个集群中的队列管理器。 发布/预订集群中的集群主题不提供相同的功能。 因此，在使用多个集群时，必须清楚地了解它们的行为。

与队列不同，不能将主题定义与多个集群相关联。 集群主题的作用域仅限于为其定义主题的同一集群中的那些队列管理器。 这允许将发布仅传播到同一集群中的那些队列管理器上的预订。

队列管理器的主题树

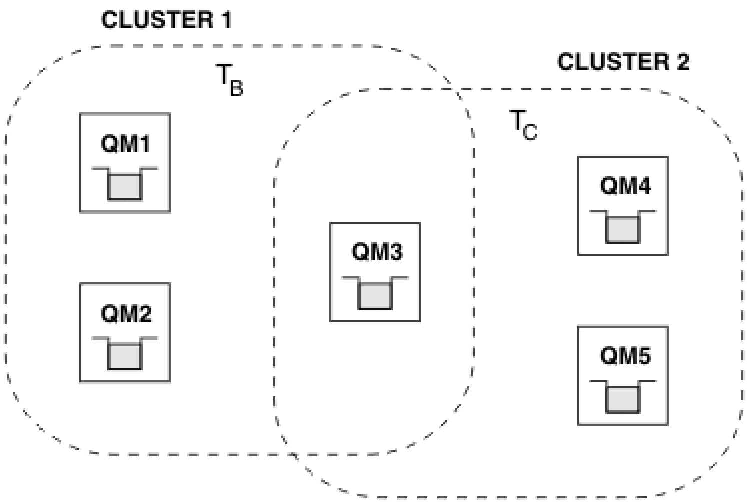


图 *28:* 重叠集群*:* 两个集群各预订不同的主题

当队列管理器是多个集群的成员时，将使其了解其中每个集群中定义的所有集群主题。 例如，在上图中，QM3 同时识别 T_B 和 T_C 受管集群主题对象，而 QM1 仅识别 T_B。 QM3 将这两个主题定义应用于其本地主题，因此对于某些主题具有不同的行为 QM1 。 因此，不同集群中的集群主题互不干扰非常重要。 当一个集群主题在另一个集群中的另一个集群主题 (例如，它们的主题字符串为 `/Sport` 和 `/Sport/Football`) 的上方或下方定义时，甚至对于这两个集群中的同一主题字符串，都可能发生干扰。 另一种干扰形式是在不同集群中使用相同的对象名定义受管集群主题对象时，但针对不同的主题字符串。

如果进行了此类配置，那么将发布传递到匹配的预订将非常依赖于发布者和订户相对于集群的相对位置。 因此，您不能依赖此类配置，您应该对其进行更改以除去干扰主题。

使用发布/预订消息传递来规划重叠集群拓扑时，可以通过将主题树和集群主题对象名称视为跨拓扑中所有重叠集群来避免任何干扰。

集成多个发布/预订集群

如果需要发布/预订消息传递以跨不同集群中的队列管理器，那么有两个选项可用:

- 通过使用发布/预订层次结构配置将集群连接在一起。 请参阅 组合多个集群的主题空间。
- 创建覆盖现有集群的其他集群，并包含需要发布或预订特定主题的所有队列管理器。

使用后一个选项时，您应该仔细考虑集群的大小以及最有效的集群路由机制。 请参阅第 63 页的『设计发布/预订集群』。

发布/预订集群中保留的发布的设计注意事项

在设计发布/预订集群以使用保留发布时，需要考虑一些限制。

注意事项

注意事项 *1:* 以下集群队列管理器始终存储保留发布的最新版本:

- 发布者的队列管理器
- 在主题主机路由集群中，主题主机 (前提是该主题只有一个主题主机，如本文下一节所述)
- 具有与保留发布的主题字符串匹配的预订的所有队列管理器

注意事项 *2:* 队列管理器在没有预订时不会接收更新的保留发布。 因此，存储在不再预订该主题的队列管理器上的任何保留发布将变得过时。

注意事项 *3:* 在创建任何预订时，如果存在主题字符串的保留发布的本地副本，那么本地副本将传递到预订。 如果您是任何给定主题字符串的第一个订户，那么还会从下列其中一个集群成员交付匹配的保留发布内容:

- 在直接路由的集群中，发布者的队列管理器
- 在主题主机路由集群中，给定主题的主题主机

将保留发布从主题主机或发布队列管理器传递到预订队列管理器与 MQSUB 调用异步。 因此，如果您使用 MQSUBRQ 调用，那么可能会错过最新的保留发布，直到后续调用 MQSUBRQ 为止。

影响

对于任何发布/预订集群，在进行第一个预订时，本地队列管理器可能正在存储保留发布的旧副本，这是传递到新预订的副本。 本地队列管理器上存在预订意味着这将在下次更新保留发布时解决。

对于主题主机路由的发布/预订集群，如果您为给定主题配置了多个主题主机，那么新订户可能会从主题主机接收最新保留发布，或者他们可能会从另一个主题主机接收旧的保留发布 (最新的已丢失)。 对于主题主机路由，通常为给定主题配置多个主题主机。 但是，如果您期望应用程序使用保留发布，那么应该仅为每个主题配置一个主题主机。

对于任何给定的主题字符串，您应该只使用单个发布程序，并确保发布程序始终使用相同的队列管理器。 如果不执行此操作，那么不同的保留发布可能在同一主题的不同队列管理器上处于活动状态，从而导致意外行为。 由于分发了多个代理预订，因此可能会接收到多个保留发布。

如果您仍然关注使用旧发布的订户，请考虑在创建每个保留发布时设置消息到期时间。

您可以使用 **CLEAR TOPICSTR** 命令从发布/预订集群中除去保留发布。 在某些情况下，您可能需要在发布/预订集群的多个成员上发出该命令，如 **CLEAR TOPICSTR** 中所述。

通配符预订和保留发布

如果您使用的是通配符预订，那么传递到发布/预订集群的其他成员的相应代理预订将在第一个通配符之前从主题分隔符进行通配符。 请参阅 通配符和集群主题。

因此，所使用的通配符可能与更多主题字符串和更多保留发布相匹配，而不是与预订应用程序相匹配。

这将增加保留发布所需的存储量，因此您需要确保主管队列管理器具有足够的存储容量。

相关概念
保留的发布内容
个人代理订阅转发和发布随处可见

针对发布/预订集群的 *REFRESH CLUSTER* 注意事项

发出 **REFRESH CLUSTER** 命令会导致队列管理器临时废弃本地保存的有关集群的信息，包括任何集群主题及其关联的代理预订。

从发出 **REFRESH CLUSTER** 命令到队列管理器完全了解集群发布/预订的必要信息所花费的时间取决于集群的大小，可用性以及完整存储库队列管理器的响应能力。

在刷新处理期间，会中断发布/预订集群中的发布/预订流量。 对于大型集群，使用 **REFRESH CLUSTER** 命令可能会在集群进行中时中断集群，并且在集群对象自动向所有相关队列管理器发送状态更新后的 27 天时间间隔内再次中断集群。 请参阅在大型集群中刷新可能会影响集群的性能和可用性。 由于这些原因，只有在 IBM 支持中心的指导下，才必须在发布/预订集群中使用 **REFRESH CLUSTER** 命令。

对集群的中断可能会在外部显示为以下症状:

- 此队列管理器上集群主题的预订未从连接到集群中其他队列管理器的发布程序接收发布。
- 发布到此队列管理器上的集群主题的消息不会传播到其他队列管理器上的预订。
- 在此期间创建的此队列管理器上的集群主题预订未一致地将代理预订发送给集群的其他成员。
- 在此期间删除的此队列管理器上的集群主题预订无法一致地从集群的其他成员中除去代理预订。
- 在消息传递中暂停 10 秒或更长时间。
- **MQPUT** 失败，例如， MQRC_PUBLICATION_FAILURE。
- 放置在死信队列上的发布，原因为 MQRC_UNKNOWN_REMOTE_Q_MGR

由于这些原因，需要先停顿发布/预订应用程序，然后再发出 **REFRESH CLUSTER** 命令。

在发布/预订集群中的队列管理器上发出 **REFRESH CLUSTER** 命令后，请等待直到成功刷新所有集群队列管理器和集群主题，然后再同步代理预订，如 代理预订再同步中所述。 当所有代理预订都已正确再同步时，请重新启动发布/预订应用程序。

如果 **REFRESH CLUSTER** 命令需要很长时间才能完成，请通过查看 SYSTEM.CLUSTER.COMMAND.QUEUE 的 CURDEPTH 来对其进行监视。

相关概念

第 58 页的『集群：使用 REFRESH CLUSTER 最佳实践』

使用 **REFRESH CLUSTER** 命令可废弃有关集群的所有本地保存的信息，并从集群中的完整存储库重建该信息。 您不应该使用此命令，除非在特殊情况下。 如果确实需要使用它，那么有关如何使用它有特殊注意事项。 此信息是基于测试和客户反馈的指南。

运行 REFRESH CLUSTER 时发现的应用程序问题

相关参考

MQSC 命令参考：REFRESH CLUSTER

在发布/预订层次结构中进行路由

如果分布式队列管理器拓扑是发布/预订层次结构，并且在队列管理器上进行了预订，那么缺省情况下将在层次结构中的每个队列管理器上创建代理预订。 然后，在任何队列管理器上接收到的发布将通过层次结构路由到托管匹配预订的每个队列管理器。

有关如何在发布/预订层次结构和集群中的队列管理器之间路由消息的简介，请参阅 分布式发布/预订网络。

在分布式发布/预订层次结构中的队列管理器上预订主题时，队列管理器将管理将预订传播到已连接的队列管理器的过程。 代理预订 流至网络中的所有队列管理器。 代理预订向队列管理器提供将发布转发到托管该主题的预订的那些队列管理器所需的信息。 发布/预订层次结构中的每个队列管理器仅知道其直接关系。 放入一个队列管理器的发布通过其直接关系发送给那些具有预订的队列管理器。 下图对此进行了说明，其中 订户 *1* 在 亚洲 队列管理器 (1) 上注册特定主题的预订。 *Asia* 队列管理器上此预订的代理预订将转发到网络中的所有其他队列管理器 (2,3, 4)。

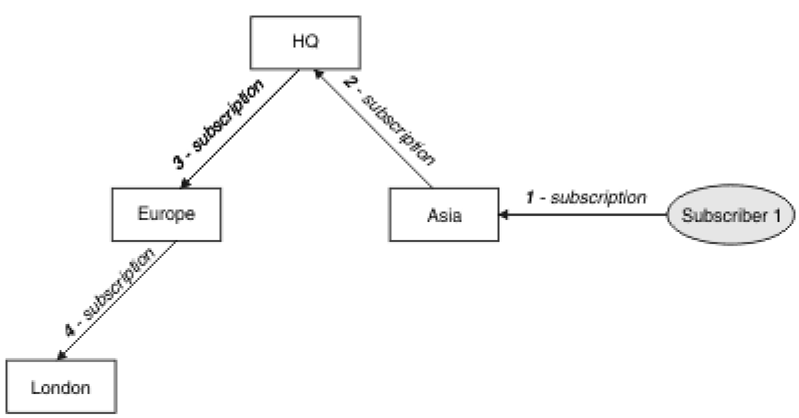


图 *29:* 通过队列管理器网络传播预订

队列管理器将合并在其上创建的所有预订，无论是从本地应用程序还是从远程队列管理器。 除非代理预订已存在，否则它将为具有其邻居的预订的主题创建代理预订。 下图对此进行了说明，其中 *Subscriber 2* 在 *HQ* 队列管理器 (5) 上注册了与 第 89 页的图 29 中相同的主题的预订。 此主题的预订将转发到 亚洲 队列管理器，以便它知道网络 (6) 上的其他位置存在预订。 不会将预订转发到 欧洲 队列管理器，因为已注册此主题的预订; 请参阅 第 89 页的图 29 中的步骤 3。

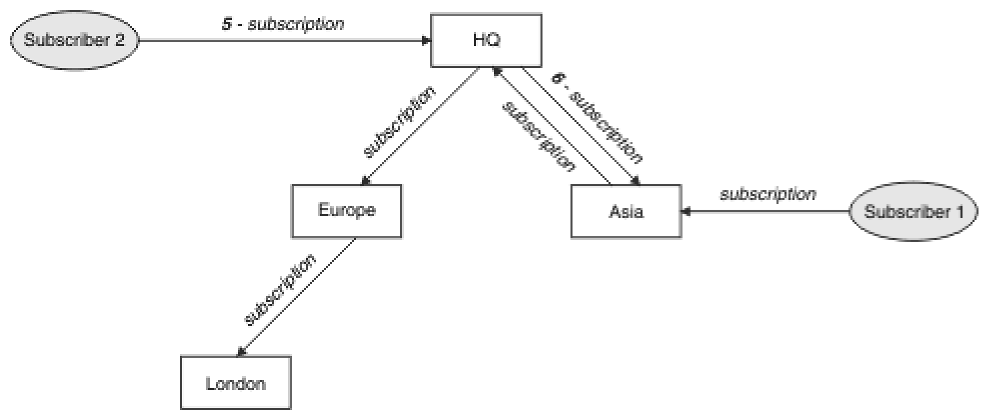


图 *30:* 多个预订

当应用程序将信息发布到主题时，缺省情况下，接收队列管理器会将其转发到具有该主题的有效预订的所有队列管理器。 它可以通过一个或多个中间队列管理器将其转发。 下图对此进行了说明，在该图中，发布者将发布内容 (与 第 90 页的图 30 中的主题相同) 发送到 欧洲 队列管理器 (7)。 此主题的预订从 *HQ* 到 欧洲，因此该出版物将转发到 *HQ* 队列管理器 (8)。 但是，不存在从 伦敦 到 欧洲 (仅从 欧洲 到 伦敦) 的预订，因此不会将该出版物转发到 伦敦 队列管理器。 *HQ* 队列管理器将发布直接发送到 *Subscriber 2* 和 *Asia* 队列管理器 (9)。 该出版物将从 亚洲 (10) 转发到 订户 *1* 。

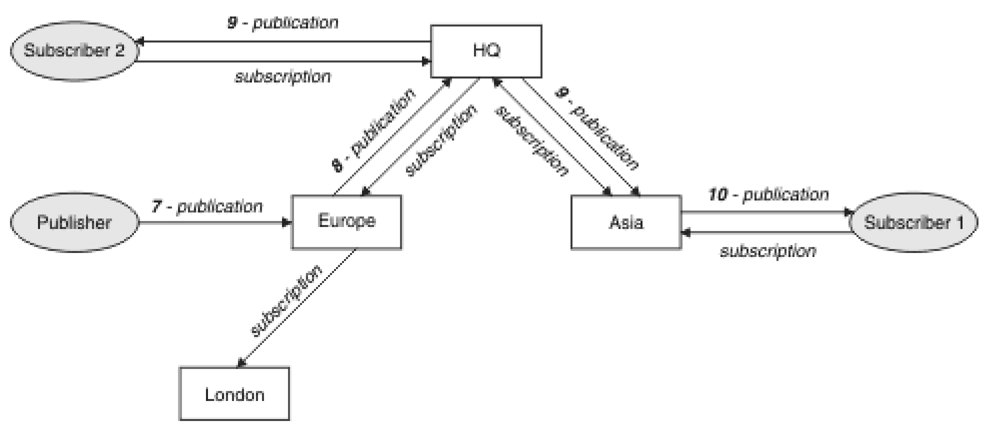


图 *31:* 通过队列管理器网络传播发布

当队列管理器将任何发布或预订发送到另一个队列管理器时，它会在消息中设置自己的用户标识。 如果您正在使用发布/预订层次结构，并且如果将入局通道设置为在消息中放入具有用户标识权限的消息，那么必须对发送队列管理器的用户标识进行授权。 请参阅 将缺省用户标识与队列管理器层次结构配合使用。

注: 如果改为使用发布/预订集群，那么授权由集群处理。

摘要和其他注意事项

发布/预订层次结构使您能够精确控制队列管理器之间的关系。 创建后，无需手动干预即可进行管理。 但是，这也会对您的系统施加某些约束:

- 层次结构中的较高节点 (尤其是根节点) 必须托管在强大，高可用性和高性能的设备上。 这是因为预期会有更多发布流量流经这些节点。
- 层次结构中每个非叶队列管理器的可用性会影响网络将消息从发布者流向其他队列管理器上的订户的能力。
- 缺省情况下，预订的所有主题字符串都在整个层次结构中传播，发布仅传播到预订了关联主题的远程队列管理器。 因此，对预订集的快速更改可能成为限制因素。 您可以更改此缺省行为，而是将所有发布传播到所有队列管理器，这将除去对代理预订的需求。 请参阅 发布/预订网络中的预订性能。

注: 类似的限制也适用于直接路由的集群。

- 由于发布/预订队列管理器的互连性质，代理预订在网络中的所有节点之间传播需要时间。 远程发布不一定会立即开始预订，因此在预订新主题字符串之后可能不会发送早期发布。 您可以通过将所有发布传播到所有队列管理器来除去预订延迟所导致的问题，这将除去对代理预订的需求。 请参阅 发布/预订网络中的预订性能。

  注: 此限制也适用于直接路由集群。

- 对于发布/预订层次结构，添加或除去队列管理器需要对该层次结构进行手动配置，并仔细考虑这些队列管理器的位置及其对其他队列管理器的依赖。 除非要添加或除去位于层次结构底部的队列管理器，因此在它们下面没有更多分支，否则还必须在层次结构中配置其他队列管理器。

在使用发布/预订层次结构作为路由机制之前，请探索 第 64 页的『发布/预订集群中的直接路由』 和 第 69 页的『发布/预订集群中的主题主机路由』中详细描述的替代方法。

分布式发布/预订系统队列

队列管理器使用四个系统队列进行发布/预订消息传递。 仅出于问题确定和容量规划目的，您需要了解它们是否存在。

请参阅 在发布/预订网络中均衡生产者和使用者， 以获取有关如何监视这些队列的指导信息。

表 6: *Multiplatforms* 版上的发布/预订系统队列

系统队列	**用途**
SYSTEM.INTER.QMGR.CONTROL	IBM MQ 分布式发布/预订控制队列
SYSTEM.INTER.QMGR.FANREQ	IBM MQ 分布式发布/预订内部代理预订扇出进程输入队列
SYSTEM.INTER.QMGR.PUBS	IBM MQ 分布式发布/预订发布
SYSTEM.HIERARCHY.STATE	IBM MQ 分布式发布/预订层次结构关系状态

z/OS 在 z/OS 上，通过在 CSQINP2 初始化输入数据集中包含 CSQ4INSX， CSQ4INSR 和 CSQ4INSG 样本，在创建队列管理器时设置必要的系统对象。 有关更多信息，请参阅 任务 13: 定制初始化输入数据集。

发布/预订系统队列的属性显示在 第 91 页的表 7 中。

表 7: 发布/预订系统队列的属性

属性	**缺省值**
DEFPSIST	Yes
DEFSOPT	SHARED
MAXMSGL	Multi 在 多平台上 :ALTER QMGR 命令的 MAXMSGL 参数值 z/OS 在 z/OS 上: 4194304 (即， 4 MB)
MAXDEPTH	999999999
SHARE	不适用
z/OS z/OS STGCLASS	此属性仅在 z/OS 平台上使用

注: 包含应用程序放入的消息的唯一队列是 `SYSTEM.INTER.QMGR.PUBS`。 **MAXDEPTH** 设置为此队列的最大值，以允许在中断或过度装入期间临时构建已发布的消息。 如果队列管理器正在无法包含该队列深度的系统上运行，那么应该对其进行调整。

相关任务

分布式发布/预订故障诊断

分布式发布/预订系统队列错误

当分布式发布/预订队列管理器队列不可用时，可能会发生错误。 这将影响在发布/预订网络中传播预订知识，并将其发布到远程队列管理器上的预订。

如果扇出请求队列 SYSTEM.INTER.QMGR.FANREQ 不可用，那么创建预订可能会生成错误，当需要将代理预订传递到直接连接的队列管理器时，会将错误消息写入队列管理器错误日志。

如果层次结构关系状态队列 SYSTEM.HIERARCHY.STATE 不可用，那么会将错误消息写入队列管理器错误日志，并将发布/预订引擎置于 COMPAT 方式。 要查看发布/预订方式，请使用命令 DISPLAY QMGR PSMODE。

如果任何其他 SYSTEM.INTER.QMGR 队列不可用，那么会将一条错误消息写入队列管理器错误日志，并且尽管功能未禁用，但发布/预订消息可能会在此队列管理器或远程队列管理器上的队列上构建。

如果发布/预订系统队列或到父，子或发布/预订集群队列管理器的必需传输队列不可用，那么将发生以下结果:

- 未交付发布，发布应用程序可能会接收到错误。 有关发布应用程序何时接收到错误的详细信息，请参阅 **DEFINE TOPIC** 命令的以下参数: **PMSGDLV** ， **NPMSGDLV** 和 **USEDLQ** 。
- 接收到的队列间管理器发布将回退到输入队列，然后重新尝试。 如果达到回退阈值，那么会将未交付的发布放在死信队列上。 队列管理器错误日志将包含问题的详细信息。
- 未交付的代理预订将回退到扇出请求队列，然后再次尝试。 如果达到回退阈值，那么未交付的代理预订不会交付到任何已连接的队列管理器，并且会放置在死信队列上。 队列管理器错误日志将包含问题的详细信息，包括所需的任何必需纠正管理操作的详细信息。
- 层次结构关系协议消息失败，连接状态标记为 ERROR。 要查看连接状态，请使用命令 **DISPLAY PUBSUB**。

相关任务

分布式发布/预订故障诊断

Multi 规划 Multiplatforms 版上的存储和性能需求

您必须为 IBM MQ 系统设置切合实际且可实现的存储和性能目标。 使用这些链接可了解影响平台上的存储和性能的因素。

根据您要在其中使用 IBM MQ 的系统以及要使用的组件不同，需求也会有所不同。

有关受支持的硬件和软件环境的最新信息，请参阅 IBM MQ 的系统需求。

IBM MQ 将队列管理器数据存储在文件系统中。 使用以下链接来了解有关规划和配置目录结构以与 IBM MQ 配合使用的信息:

- 第 96 页的『多平台上的规划文件系统支持』
- 第 97 页的『Multiplatforms 版上共享文件系统的需求』
- 第 105 页的『在 Multiplatforms 版上共享 IBM MQ 文件』
- Linux AIX 第 107 页的『AIX and Linux 系统上的目录结构』
- Windows 第 116 页的『Windows 系统上的目录结构』
- IBM i 第 119 页的『IBM i 上的目录结构』

使用以下链接以获取有关 AIX and Linux 上的系统资源，共享内存和进程优先级的信息:

- Linux AIX 第 123 页的『IBM MQ 和 UNIX System V IPC 资源』
- AIX 第 123 页的『AIX 上的共享内存』
- Linux AIX 第 123 页的『IBM MQ 和 UNIX 进程优先级』

使用以下链接以获取有关日志文件的信息:

- 第 122 页的『在 Multiplatforms 版上选择循环或线性日志记录』
- 计算日志大小

相关概念

第 124 页的『在 z/OS 上规划 IBM MQ 环境』
规划 IBM MQ 环境时，必须考虑数据集，页集， Db2，耦合设施的资源需求以及日志记录和备份设施的需求。 使用本主题来规划运行 IBM MQ 的环境。

相关任务

第 5 页的『规划 IBM MQ 体系结构』
规划 IBM MQ 环境时，请考虑 IBM MQ 为单个和多个队列管理器体系结构以及点到点和发布/预订消息传递样式提供的支持。 还要规划资源需求以及日志记录和备份工具的使用。

相关参考

AIX and Linux 上的硬件和软件需求
Windows 上的硬件和软件需求

Multi Multiplatforms 版上的磁盘空间需求

IBM MQ 的存储需求取决于您安装的组件以及所需的工作空间量。

您选择安装的可选组件 (包括所需的任何必备组件) 需要磁盘存储器。 总存储需求还取决于您使用的队列数，队列上消息的数量和大小以及消息是否持久。 您还需要磁盘，磁带或其他介质上的归档容量以及您自己的应用程序的空间。

下表显示在不同平台上安装产品的各种组合时所需的大致磁盘空间。 (值向上取整为最接近的 5 MB， 其中 MB 为 1,048,576 字节。)

- LTS 第 93 页的『Long Term Support 的磁盘空间需求』
- CD 第 94 页的『Continuous Delivery 的磁盘空间需求』

Long Term Support 的磁盘空间需求

LTS V 9.2.0

表 8: IBM MQ Multiplatforms for Long Term Support 的磁盘空间需求

平台	客户端安装第 94 页的『1』	服务器安装 第 94 页的『2』	完全安装 第 94 页的『3』
AIX AIX	325 MB	370 MB	1690 MB
IBM i IBM i (请参阅 IBM i 的其他说明)	480 MB	840 MB	1815 MB
Linux Linux for x86-64	245 MB	270 MB	1835 MB
Linux Linux on POWER Systems - Little Endian	150 MB	170 MB	1205 MB
Linux Linux for IBM Z	235 MB	260 MB	1240 MB
Windows Windows (64 位安装) 第 94 页的『4』	280 MB	390 MB	1755 MB

注意:

1. 客户机安装包含以下组件:
    - 运行时
    - 客户机
2. 服务器安装包含以下组件:
    - 运行时
    - 服务器
3. 完整安装包含所有可用组件。
4. Windows 并非此处列出的所有组件都是 Windows 系统上的可安装功能部件; 它们的功能有时包含在其他功能部件中。 请参阅 Windows 系统的 IBM MQ 功能部件。

IBM i 的其他说明: IBM i

1. 在 IBM i 上，无法将本机客户机与服务器分开。 表中的服务器图用于不带 Java 的 5724H72*BASE 以及英语语言装入 (2924)。 有 22 种可能的唯一语言负载。
2. 该表中的图适用于没有 Java 的本机客户机 5725A49 *BASE。
3. 可以将 Java 和 JMS 类添加到服务器和客户机绑定。 如果要包含这些功能部件，请添加 110 MB。
4. 将样本源添加到客户机或服务器将额外添加 10 MB。
5. 向 Java 和 JMS 类添加样本将添加额外的 5 MB。

Continuous Delivery 的磁盘空间需求

CD

表 9: *IBM MQ Multiplatforms for Continuous Delivery* 的磁盘空间需求

平台/CD 发行版	客户端安装 第 96 页的『1』	服务器安装 第 96 页的『2』	完全安装 第 96 页的『3』
AIX AIX			
V 9.2.0 IBM MQ 9.2.0	150 MB	205 MB	1410 MB
V 9.2.1 IBM MQ 9.2.1	325 MB	370 MB	1690 MB
V 9.2.2 IBM MQ 9.2.2	325 MB	370 MB	1690 MB
V 9.2.3 IBM MQ 9.2.3	325 MB	370 MB	1690 MB
V 9.2.4 IBM MQ 9.2.4	330 MB	375 MB	1690 MB
V 9.2.5 IBM MQ 9.2.5	330 MB	375 MB	1760 MB
Linux Linux for x86-64 (64 位)			
V 9.2.0 IBM MQ 9.2.0	130 MB	185 MB	1645 MB
V 9.2.1 IBM MQ 9.2.1	245 MB	270 MB	1835 MB

表 9: *IBM MQ Multiplatforms for Continuous Delivery* 的磁盘空间需求 *(继续)*			
平台/CD 发行版	**客户端安装** 第 96 页的『1』	**服务器安装** 第 96 页的『2』	**完全安装** 第 96 页的『3』
V 9.2.2 IBM MQ 9.2.2	245 MB	270 MB	1835 MB
V 9.2.3 IBM MQ 9.2.3	245 MB	270 MB	1835 MB
V 9.2.4 IBM MQ 9.2.4	245 MB	270 MB	1870 MB
V 9.2.5 IBM MQ 9.2.5	265 MB	295 MB	2010 MB
Linux **Linux on POWER Systems - Little Endian**			
V 9.2.0 IBM MQ 9.2.0	95 MB	135 MB	1100 MB
V 9.2.1 IBM MQ 9.2.1	150 MB	170 MB	1205 MB
V 9.2.2 IBM MQ 9.2.2	150 MB	170 MB	1205 MB
V 9.2.3 IBM MQ 9.2.3	150 MB	170 MB	1205 MB
V 9.2.4 IBM MQ 9.2.4	150 MB	170 MB	1210 MB
V 9.2.5 IBM MQ 9.2.5	170 MB	190 MB	1350 MB
Linux **Linux for IBM Z**			
V 9.2.0 IBM MQ 9.2.0	130 MB	175 MB	1130 MB
V 9.2.1 IBM MQ 9.2.1	235 MB	265 MB	1255 MB
V 9.2.2 IBM MQ 9.2.2	235 MB	265 MB	1255 MB
V 9.2.3 IBM MQ 9.2.3	235 MB	265 MB	1255 MB
V 9.2.4 IBM MQ 9.2.4	235 MB	265 MB	1300 MB
V 9.2.5 IBM MQ 9.2.5	255 MB	290 MB	1435 MB
Windows **Windows (64 位安装)** 第 96 页的『4』			

表 9: *IBM MQ Multiplatforms for Continuous Delivery* 的磁盘空间需求 *(*继续*)*			
平台/CD 发行版	**客户端安装**第 96 页的『1』	**服务器安装** 第 96 页的『2』	**完全安装** 第 96 页的『3』
V 9.2.0 IBM MQ 9.2.0	230 MB	365 MB	1565 MB
V 9.2.1 IBM MQ 9.2.1	280 MB	390 MB	1900 MB
V 9.2.2 IBM MQ 9.2.2	280 MB	390 MB	1900 MB
V 9.2.3 IBM MQ 9.2.3	280 MB	390 MB	1900 MB
V 9.2.4 IBM MQ 9.2.4	280 MB	390 MB	1950 MB
V 9.2.5 IBM MQ 9.2.5	290 MB	410 MB	2095 MB

注意:

1. 客户机安装包含以下组件:
   - 运行时
   - 客户机
2. 服务器安装包含以下组件:
   - 运行时
   - 服务器
3. 完整安装包含所有可用组件。
4. Windows 并非此处列出的所有组件都是 Windows 系统上的可安装功能部件; 它们的功能有时包含在其他功能部件中。 请参阅 Windows 系统的 IBM MQ 功能部件。

相关概念

IBM MQ 组件和功能

Multi 多平台上的规划文件系统支持

队列管理器数据存储在文件系统中。 队列管理器使用文件系统锁定来防止多实例队列管理器的多个实例同时处于活动状态。

共享文件系统

共享文件系统使多个系统能够同时访问同一物理存储设备。 如果多个系统直接访问同一物理存储设备，而没有某种强制锁定和并行控制的方法，那么将发生损坏。 操作系统为本地文件系统提供对本地进程的锁定和并行控制; 网络文件系统为分布式系统提供锁定和并行控制。

历史上，网络文件系统的执行速度不够快，或者提供了足够的锁定和并行控制，无法满足日志记录消息的要求。 如今，联网的文件系统可以提供良好的性能，实现可靠的网络文件系统协议，如 *RFC 3530* ，网络文件系统 *(NFS)* 版本 *4* 协议，满足可靠记录消息的需求。

共享文件系统和 IBM MQ

多实例队列管理器的队列管理器数据存储在共享网络文件系统中。 在 AIX, Linux, and Windows 系统上，队列管理器的数据文件和日志文件必须放在共享网络文件系统中。 IBM i 在 IBM i 上，将使用日记帐而不是日志文件，并且无法共享日记帐。 IBM i 上的多实例队列管理器使用日志复制或可切换日志，以使日志在不同队列管理器实例之间可用。

IBM MQ 使用锁定来防止同一多实例队列管理器的多个实例同时处于活动状态。 同一锁定还可确保两个单独的队列管理器不能无意中使用同一组队列管理器数据文件。 一次只能有一个队列管理器实例具有其锁定。 因此， IBM MQ 支持存储在作为共享文件系统访问的联网存储器上的队列管理器数据。

由于并非网络文件系统的所有锁定协议都是健全的，并且由于可能针对性能而非数据完整性配置了文件系统，因此您必须运行 **amqmfsck** 命令来测试网络文件系统是否将正确控制对队列管理器数据和日志的访问。 此命令仅适用于 UNIX， Linux 和 IBM i 系统。 在 Windows 上，只有一个受支持的网络文件系统，不需要 **amqmfsck** 命令。

相关任务

第 98 页的『在 Multiplatforms 版上验证共享文件系统行为』
运行 **amqmfsck** 以检查 AIX， Linux 或 IBM i 上的共享文件系统是否满足存储多实例队列管理器的队列管理器数据的需求。 (Windows 配置的唯一要求是它将 SMB 3 用于共享存储器供应。)

Multi Multiplatforms 版上共享文件系统的需求

共享文件系统必须提供数据写完整性，保证对文件的独占访问权，并在无法可靠地使用 IBM MQ 时释放锁定。

共享文件系统必须满足的需求

共享文件系统必须满足三个基本要求才能可靠地使用 IBM MQ:

1. 数据写入完整性

   数据写入完整性有时称为 清空时写入磁盘。 队列管理器必须能够与成功落实到物理设备的数据同步。 在事务系统中，您需要确保在继续其他处理之前已安全地落实某些写操作。

   更具体地说， IBM MQ for AIX or Linux 平台使用 *O_SYNC* 打开选项和 `fsync()` 系统调用来显式强制写入可恢复介质，并且写操作依赖于正确运行的这些选项。

   注意: Linux 您应该使用 async 选项安装文件系统，该选项仍然支持同步写入选项，并提供比 sync 选项更好的性能。

   但是，请注意，如果已从 Linux 导出文件系统，那么仍必须使用 sync 选项导出文件系统。

2. 保证对文件的独占访问权

   为了使多个队列管理器同步，需要有一种队列管理器获取对文件的互斥锁定的机制。

3. 发生故障时释放锁定

   如果队列管理器发生故障，或者如果与文件系统发生通信故障，那么队列管理器锁定的文件需要解锁并可供其他进程使用，而无需等待队列管理器重新连接到文件系统。

共享文件系统必须满足这些要求才能使 IBM MQ 可靠运行。 如果没有，那么在多实例队列管理器配置中使用共享文件系统时，队列管理器数据和日志会损坏。

对于 Microsoft Windows 上的多实例队列管理器，联网存储器必须由 Microsoft Windows 网络所使用的服务器消息块 (SMB) 协议访问。 服务器消息块 (SMB) 客户机不满足 IBM MQ 在 Microsoft Windows 以外的平台上锁定语义的要求，因此在 Microsoft Windows 以外的平台上运行的多实例队列管理器不得使用服务器消息块 (SMB) 作为其共享文件系统。

对于其他受支持平台上的多实例队列管理器，必须通过符合 Posix 的网络文件系统协议来访问存储器，并支持基于租赁的锁定。 网络文件系统 4 满足此要求。 不能将较旧的文件系统 (例如，网络文件系统版本 3) 与多实例队列管理器配合使用，这些系统没有在发生故障后释放锁定的可靠机制。

检查共享文件系统是否满足需求

您必须检查计划使用的共享文件系统是否满足这些需求。 您还必须检查是否正确配置了文件系统以实现可靠性。 共享文件系统有时会提供配置选项，以牺牲可靠性来提高性能。

有关更多信息，请参阅 测试 IBM MQ 多实例队列管理器文件系统的语句。

在正常情况下， IBM MQ 使用属性高速缓存正常运行，不需要禁用高速缓存，例如通过在 NFS 安装上设置 NOAC。 当多个文件系统客户机争用对文件系统服务器上同一文件的写访问权时，属性高速缓存可能会导致问题，因为每个客户机使用的高速缓存属性可能与服务器上的那些属性不同。 以此方式访问的文件的一个示例是多实例队列管理器的队列管理器错误日志。 队列管理器错误日志可能由活动队列管理器实例和备用队列管理器实例写入，并且高速缓存的文件属性可能导致在发生文件回滚之前，错误日志的大小大于预期。

要帮助检查文件系统，请运行任务 验证共享文件系统行为。 此任务检查共享文件系统是否满足需求 2 和 3。 您需要验证共享文件系统文档中的需求 1，或者通过尝试将数据记录到磁盘。

在写入磁盘时，磁盘故障会导致错误， IBM MQ 会将这些错误报告为 "首次故障数据捕获" 错误。 您可以针对操作系统运行文件系统检查程序，以检查共享文件系统是否存在任何磁盘故障。 例如：

- Linux AIX 在 AIX and Linux 上，文件系统检查程序称为 fsck。
- Windows 在 Windows 平台上，文件系统检查程序称为 CHKDSK 或 SCANDISK。

NFS 服务器安全性

注意:

- 不能对用于存放 IBM MQ 安装目录的安装点使用 **nosuid** 或 **noexec** 选项。 这是因为 IBM MQ 包含 setuid/setgid 可执行程序，并且不能阻止这些程序正常运行。
- 将队列管理器数据仅放在网络文件系统 (NFS) 服务器上时，可以将以下三个选项与 mount 命令配合使用，以确保系统安全，而不会对队列管理器的运行造成有害影响:

  诺埃克
  : 通过使用此选项，可以阻止二进制文件在 NFS 上运行，这将阻止远程用户在系统上运行不需要的代码。

  诺苏伊德
  : 通过使用此选项，可防止使用 set-user-identifier 和 set-group-identifier 位，这将阻止远程用户获取更高的特权。

  节点 v
  : 通过使用此选项，可以停止字符并阻止使用或定义特殊设备，这将阻止远程用户走出 chroot 监狱。

Linux IBM i AIX 在 *Multiplatforms* 版上验证共享文件系统行为

运行 **amqmfsck** 以检查 AIX， Linux 或 IBM i 上的共享文件系统是否满足存储多实例队列管理器的队列管理器数据的需求。 (Windows 配置的唯一要求是它将 SMB 3 用于共享存储器供应。)

开始之前

您需要一个具有联网存储器的服务器，以及另外两个已安装 IBM MQ 的连接到该服务器的服务器。 您必须具有管理员 (root) 权限才能配置文件系统，并且必须是 IBM MQ 管理员才能运行 **amqmfsck**。

关于此任务

第 97 页的『Multiplatforms 版上共享文件系统的需求』 描述了将共享文件系统与多实例队列管理器配合使用的文件系统需求。 IBM MQ 技术说明 针对 IBM MQ 多实例队列管理器文件系统的测试语句 列出了 IBM 已使用其进行测试的共享文件系统。 本任务中的过程描述了如何测试文件系统以帮助您评估未列出的文件系统是否保持数据完整性。

多实例队列管理器的故障转移可由硬件或软件故障触发，包括阻止队列管理器写入其数据或日志文件的网络问题。 主要是您有兴趣在文件服务器上引起故障。 但是，您还必须导致 IBM MQ 服务器失败，以测试是否成功释放了任何锁定。 要对共享文件系统充满信心，请测试以下所有故障以及特定于您的环境的任何其他故障:

1. 在文件服务器上关闭操作系统，包括同步磁盘。
2. 正在停止文件服务器上的操作系统而不同步磁盘。
3. 按每个服务器上的重置按钮。
4. 从每个服务器拔出网络电缆。
5. 从每个服务器拔出电源线。
6. 关闭每个服务器。

在要用于共享队列管理器数据和日志的网络存储器上创建目录。 目录所有者必须是 IBM MQ 管理员，换言之，是 AIX and Linux 上 mqm 组的成员。 运行测试的用户必须具有 IBM MQ 管理员权限。

使用在 在 Linux 上创建多实例队列管理器 或 在 IBM i 上使用日志镜像和 NetServer 创建多实例队列管理器 中导出和安装文件系统的示例来帮助您配置文件系统。 不同的文件系统要求执行不同的配置步骤。 请阅读文件系统文档。

注: 与 **amqmfsck** 并行运行 IBM MQ MQI client 样本程序 **amqsfhac** 以演示队列管理器在故障期间维护消息完整性。

过程

在每个检查中，当文件系统检查程序正在运行时，将导致先前列表中的所有故障。 如果您打算与 **amqmfsck** 同时运行 **amqsfhac** ，请执行与此任务并行的任务 第 103 页的『运行 amqsfhac 以测试消息完整性』 。

1. 将导出的目录安装在两个 IBM MQ 服务器上。

   在文件系统服务器上，创建共享目录 `shared` 和子目录以保存多实例队列管理器 `qmdata` 的数据。 有关在 Linux 上为多实例队列管理器设置共享目录的示例，请参阅 在 Linux 上创建多实例队列管理器
2. 检查基本文件系统行为。

   在一个 IBM MQ 服务器上，运行不带参数的文件系统检查程序。

   在 IBM MQ 服务器 1 上:

   ```
   amqmfsck /shared/qmdata
   ```
3. 检查是否同时从两个 IBM MQ 服务器写入同一目录。

   在两个 IBM MQ 服务器上，使用 `-c` 选项同时运行文件系统检查程序。

   在 IBM MQ 服务器 1 上:

   ```
   amqmfsck -c /shared/qmdata
   ```

   在 IBM MQ 服务器 2 上:

   ```
   amqmfsck -c /shared/qmdata
   ```
4. 检查是否正在等待和释放两个 IBM MQ 服务器上的锁定。

   在两个 IBM MQ 服务器上，使用 `-w` 选项同时运行文件系统检查程序。

   在 IBM MQ 服务器 1 上:

   ```
   amqmfsck -w /shared/qmdata
   ```

   在 IBM MQ 服务器 2 上:

   ```
   amqmfsck -w /shared/qmdata
   ```
5. 检查数据完整性。

   a) 格式化测试文件。

在要测试的目录中创建大文件。 文件已格式化，以便后续阶段可以成功完成。 该文件必须足够大，以便有足够的时间来中断第二阶段以模拟故障转移。 尝试缺省值 262144 页 (1 GB)。 程序会自动减少慢速文件系统上的此缺省值，以便在大约 60 秒内完成格式化

在 IBM MQ 服务器 1 上:

```
amqmfsck -f /shared/qmdata
```

服务器通过以下消息进行响应:

```
Formatting test file for data integrity test.

Test file formatted with 262144 pages of data.
```

b) 在导致失败时，使用文件系统检查程序将数据写入测试文件。

同时在两台服务器上运行测试程序。 在将要经历故障的服务器上启动测试程序，然后在将要经历故障的服务器上启动测试程序。 导致您正在调查的故障。

第一个测试程序停止，并显示错误消息。 第二个测试程序获取对测试文件的锁定，并从第一个测试程序离开的位置开始将数据写入测试文件。 让第二个测试程序运行到完成。

<table>
<caption>表 *10:* 同时在两台服务器上运行数据完整性检查</caption>
<tr><th>IBM MQ 服务器 1</th><th>IBM MQ 服务器 2</th></tr>
<tr><td><pre>amqmfsck -a /shared/qmdata</pre></td><td></td></tr>
<tr><td rowspan="2"><pre>Please start this program on a second
machine
with the same parameters.

File lock acquired.

Start a second copy of this program
with the same parameters on another
server.

Writing data into test file.

To increase the effectiveness of the test,
interrupt the writing by ending the
process,
temporarily breaking the network
connection
to the networked storage,
rebooting the server or turning off the
power.</pre></td><td><pre>amqmfsck -a /shared/qmdata</pre></td></tr>
<tr><td rowspan="2"><pre>Waiting for lock...

Waiting for lock...

Waiting for lock...

Waiting for lock...

Waiting for lock...

Waiting for lock...</pre></td></tr>
<tr><td><code>Turn the power off here.</code></td></tr>
</table>

表 *10:* 同时在两台服务器上运行数据完整性检查 *(*继续*)*

IBM MQ 服务器 1	IBM MQ 服务器 2

	`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.`

测试的时间取决于文件系统的行为。 例如，在断电后，文件系统通常需要 30 到 90 秒来释放由第一个程序获取的文件锁定。 如果在第一个测试程序填充该文件之前，您没有太多时间来引入该故障，请使用 **amqmfsck** 的 `-x` 选项来删除该测试文件。 使用更大的测试文件从头开始尝试测试。

c) 验证测试文件中数据的完整性。

在 IBM MQ 服务器 2 上:

```
amqmfsck -i /shared/qmdata
```

服务器通过以下消息进行响应:

```
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. 删除测试文件。

在 IBM MQ 服务器 2 上:

```
amqmfsck -x /shared/qmdata

Test files deleted.
```

服务器通过以下消息进行响应:

```
Test files deleted.
```

结果

如果测试成功完成，那么程序返回的退出代码为零，否则返回非零。

示例

第一组三个示例显示了生成最小输出的命令。

在一台服务器上成功测试基本文件锁定

```
> amqmfsck /shared/qmdata
The tests on the directory completed successfully.
```

在一个服务器上测试基本文件锁定失败

```
> amqmfsck /shared/qmdata
AMQ6245: Error Calling 'write()[2]' on file '/shared/qmdata/amqmfsck.lck' error '2'.
```

在两台服务器上成功进行锁定测试

表 *11:* 在两台服务器上成功锁定

IBM MQ 服务器 1	**IBM MQ 服务器 2**

`> amqmfsck -w /shared/qmdata`
`Please start this program on a second`
`machine with the same parameters.`
`Lock acquired.`
`Press Return`
`or terminate the program to release the lock.`	

	`> amqmfsck -w /shared/qmdata` `Waiting for lock...`
`[ Return pressed ]` `Lock released.`	

	`Lock acquired.`
	`The tests on the directory completed`
	`successfully`

第二组三个示例显示了使用详细方式的相同命令。

在一台服务器上成功测试基本文件锁定

```
> amqmfsck -v /shared/qmdata
System call: stat("/shared/qmdata")'
System call: fd = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fchmod(fd, 0666)
System call: fstat(fd)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: write(fd)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd1 = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fcntl(fd1, F_SETLK, F_RDLCK)
System call: fd2 = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fcntl(fd2, F_SETLK, F_RDLCK)
System call: close(fd2)
System call: write(fd1)
System call: close(fd1)
The tests on the directory completed successfully.
```

在一个服务器上测试基本文件锁定失败

```
> amqmfsck -v /shared/qmdata
System call: stat("/shared/qmdata")
System call: fd = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fchmod(fd, 0666)
System call: fstat(fd)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: write(fd)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
```

```
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fcntl(fd, F_SETLK, F_RDLCK)
System call: fdSameFile = open("/shared/qmdata/amqmfsck.lck", O_RDWR, 0666)
System call: fcntl(fdSameFile, F_SETLK, F_RDLCK)
System call: close(fdSameFile)
System call: write(fd)
AMQxxxx: Error calling 'write()[2]' on file '/shared/qmdata/amqmfsck.lck', errno 2
(Permission denied).
```

在两台服务器上成功进行锁定测试

表 *12:* 在两个服务器上成功锁定-详细方式

IBM MQ 服务器 1	IBM MQ 服务器 2

`> amqmfsck -wv /shared/qmdata`
`Calling 'stat("/shared/qmdata")'`
`Calling 'fd = open("/shared/qmdata/`
`amqmfsck.lkw",`
`O_EXCL | O_CREAT | O_RDWR, 0666)'`
`Calling 'fchmod(fd, 0666)'`
`Calling 'fstat(fd)'`
`Please start this program on a second`
`machine with the same parameters.`
`Calling 'fcntl(fd, F_SETLK, F_WRLCK)'`
`Lock acquired.`
`Press Return`
`or terminate the program to release the lock.`	

	`> amqmfsck -wv /shared/qmdata`
	`Calling 'stat("/shared/qmdata")'`
	`Calling 'fd = open("/shared/qmdata/`
	`amqmfsck.lkw",`
	`O_EXCL | O_CREAT | O_RDWR,0666)'`
	`Calling 'fd = open("/shared/qmdata/amqmfsck.lkw,`
	`O_RDWR, 0666)'`
	`Calling 'fcntl(fd, F_SETLK, F_WRLCK)`
	`'Waiting for lock...`

`[ Return pressed ]`
`Calling 'close(fd)'`
`Lock released.`	

	`Calling 'fcntl(fd, F_SETLK, F_WRLCK)'`
	`Lock acquired.`
	`The tests on the directory completed`
	`successfully`

相关参考

“高可用性”样本程序

Multi 运行 *amqsfhac* 以测试消息完整性

与 **amqmfsck** 并行运行 IBM MQ MQI client 样本程序 **amqsfhac** 以演示队列管理器在故障期间维护消息完整性。

开始之前

此测试需要四个服务器。 两个服务器用于多实例队列管理器，一个用于文件系统，另一个用于将 **amqsfhac** 作为 IBM MQ MQI client 应用程序运行。

遵循 第 98 页的『在 Multiplatforms 版上验证共享文件系统行为』 中的步骤 第 99 页的『1』 ，为多实例队列管理器设置文件系统。

关于此任务

IBM MQ MQI client 样本程序 **amqsfhac** 检查使用联网存储器的队列管理器在发生故障后是否保持数据完整性。 与 **amqmfsck** 并行运行 **amqsfhac** 以演示队列管理器在故障期间维护消息完整性。

过程

1. 使用您在 过程中的步骤 第 99 页的『1』 中创建的文件系统，在另一服务器 QM1 上创建多实例队列管理器。

   请参阅 创建多实例队列管理器。
2. 在两个服务器上启动队列管理器，使其具有高可用性。

   在服务器 1 上:

   ```
   strmqm -x QM1
   ```

   在服务器 2 上:

   ```
   strmqm -x QM1
   ```
3. 设置客户机连接以运行 **amqsfhac**。
   a) 对于您的企业用于设置客户机连接的平台或平台，请使用 验证 *IBM MQ* 安装 中的过程，或者使用 可重新连接的客户机样本中的示例脚本。
   b) 修改客户机通道以具有两个 IP 地址，对应于运行 QM1 的两个服务器。

      在示例脚本中，修改:

      ```
      DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
      CONNAME('LOCALHOST(2345)') QMNAME(QM1) REPLACE
      ```

      到:

      ```
      DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
      CONNAME('server1(2345),server2(2345)') QMNAME(QM1) REPLACE
      ```

      其中 server1 和 server2 是两个服务器的主机名， 2345 是通道侦听器正在侦听的端口。 通常，此值缺省为 1414。 您可以将 1414 与缺省侦听器配置配合使用。
4. 在 QM1 上为测试创建两个本地队列。
   运行以下 MQSC 脚本:

   ```
   DEFINE QLOCAL(TARGETQ) REPLACE
   DEFINE QLOCAL(SIDEQ) REPLACE
   ```
5. 使用 **amqsfhac** 测试配置

   ```
   amqsfhac QM1 TARGETQ SIDEQ 2 2 2
   ```
6. 测试文件系统完整性时测试消息完整性。

   在 第 98 页的『在 Multiplatforms 版上验证共享文件系统行为』的步骤 第 99 页的『5』 期间运行 **amqsfhac** 。

   ```
   amqsfhac QM1 TARGETQ SIDEQ 10 20 0
   ```

   如果停止活动队列管理器实例，那么 **amqsfhac** 将在另一个队列管理器实例处于活动状态后重新连接到该实例。 再次重新启动已停止的队列管理器实例，以便您可以在下一次测试中反转故障。 您可能需要根据对环境的试验增加迭代次数，以便测试程序运行足够的时间进行故障转移。

结果

以下示例显示了在步骤 第 104 页的『6』 中运行 **amqsfhac** 的示例。 在此示例中，测试成功。

```
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
```

如果测试检测到问题，那么输出将报告故障。 在某些测试中， MQRC_CALL_INTERRUPTED 可能会报告 “Resolving to backed out”。 它与结果没有区别。 结果取决于是在发生故障之前还是之后由联网文件存储器落实对磁盘的写入。

相关参考

amqmfsck（文件系统检查）

“高可用性”样本程序

Multi 在 Multiplatforms 版上共享 IBM MQ 文件

某些 IBM MQ 文件由活动队列管理器独占访问，其他文件共享。

IBM MQ 文件拆分为程序文件和数据文件。 程序文件通常以本地方式安装在运行 IBM MQ 的每个服务器上。 队列管理器共享对缺省数据目录中的数据文件和目录的访问权。 他们需要对包含在 第 106 页的图 32 中显示的每个 qmgrs 和 log 目录中的自己的队列管理器目录树进行独占访问。

第 106 页的图 32 是 IBM MQ 目录结构的高级视图。 它显示了可以在队列管理器之间共享并使其成为远程目录的目录。 详细信息因平台而异。 虚线指示可配置路径。

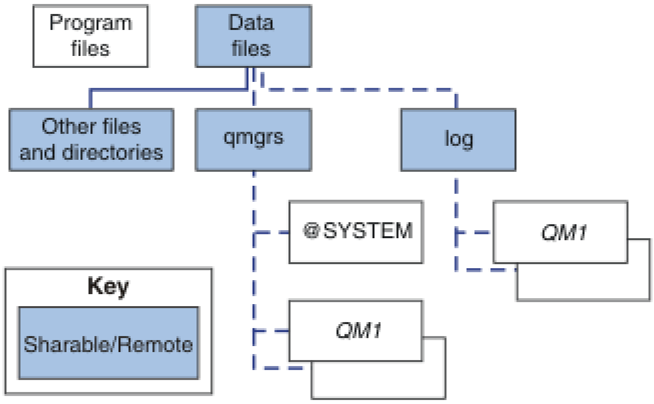


图 32: *IBM MQ* 目录结构的整体视图

程序文件

程序文件目录通常保留在缺省位置，是本地的，并且由服务器上的所有队列管理器共享。

数据文件

数据文件目录通常在缺省位置 /var/mqm (在 AIX and Linux 系统上) 本地，并可在 Windows 上安装时进行配置。 它在队列管理器之间共享。 您可以使缺省位置成为远程位置，但不要在 IBM MQ 的不同安装之间共享该位置。 IBM MQ 配置中的 DefaultPrefix 属性指向此路径。

qmgrs

有两种替代方法可指定队列管理器数据的位置。

使用 前缀

Prefix 属性指定 qmgrs 目录的位置。 IBM MQ 根据队列管理器名称构造队列管理器目录名，并将其创建为 qmgrs 目录的子目录。

Prefix 属性位于 QueueManager 节中，并从 DefaultPrefix 属性中的值继承。 缺省情况下，为了简化管理，队列管理器通常共享同一 qmgrs 目录。

QueueManager 节位于 mqs.ini 文件中。

如果更改任何队列管理器的 qmgrs 目录的位置，那么必须更改其 Prefix 属性的值。

AIX and Linux 平台的 第 106 页的图 32 中 QM1 目录的 Prefix 属性如下所示:

```
Prefix=/var/mqm
```

使用 DataPath

DataPath 属性指定队列管理器数据目录的位置。

DataPath 属性指定完整路径，包括队列管理器数据目录的名称。 DataPath 属性与 Prefix 属性不同，后者指定队列管理器数据目录的不完整路径。

DataPath 属性 (如果已指定) 位于 QueueManager 节中。 如果已指定，那么它优先于 Prefix 属性中的任何值。

QueueManager 节位于 mqs.ini 文件中。

如果更改任何队列管理器的队列管理器数据目录的位置，那么必须更改 DataPath 属性的值。

AIX 或 Linux 平台的 第 106 页的图 32 中 QM1 目录的 DataPath 属性为:

```
DataPath=/var/mqm/qmgrs/QM1
```

log

将在队列管理器配置的 Log 节中为每个队列管理器单独指定日志目录。 队列管理器配置在 qm.ini 中。

***DataPath/QmgrName/*@IPCC 子目录**

DataPath/*QmgrName*/@IPCC 子目录位于共享目录路径中。 它们用于构造 IPC 文件系统对象的目录路径。 当在系统之间共享队列管理器时，它们需要区分队列管理器的名称空间。

IPC 文件系统对象必须由系统进行区分。 对于运行队列管理器的每个系统，会将一个子目录添加到目录路径中，请参阅 第 107 页的图 33。

```
DataPath/QmgrName/@IPCC/esem/myHostName/
```

图 *33:* 示例 *IPC* 子目录

myHostName 最多包含操作系统返回的主机名的前 20 个字符。 在某些系统上，在截断之前，主机名的长度可能最多为 64 个字符。 生成的 *myHostName* 值可能由于以下两个原因导致问题:

1. 前 20 个字符不唯一。
2. 主机名由 DHCP 算法生成，该算法并不总是将相同的主机名分配给系统。

在这些情况下，使用环境变量 MQS_IPC_HOST 设置 *myHostName* ; 请参阅 第 107 页的图 34。

```
export MQS_IPC_HOST= myHostName
```

图 *34:* 示例*:* 设置 *MQS_IPC_HOST*

其他文件和目录

其他文件和目录 (例如包含跟踪文件的目录和公共错误日志) 通常共享并保留在本地文件系统上。

通过支持共享文件系统， IBM MQ 使用文件系统锁定来管理对这些文件的互斥访问。 文件系统锁定一次仅允许特定队列管理器的一个实例处于活动状态。

当您启动特定队列管理器的第一个实例时，它将获取其队列管理器目录的所有权。 如果启动第二个实例，那么仅当第一个实例已停止时，才能获取所有权。 如果第一个队列管理器仍在运行，那么第二个实例无法启动，并报告队列管理器正在其他位置运行。 如果第一个队列管理器已停止，那么第二个队列管理器将接管队列管理器文件的所有权并成为正在运行的队列管理器。

您可以自动执行第二个队列管理器从第一个队列管理器接管的过程。 使用 strmqm -x 选项启动第一个队列管理器，该选项允许另一个队列管理器从该队列管理器接管。 然后，第二个队列管理器将等待队列管理器文件解锁，然后再尝试接管队列管理器文件的所有权，然后启动。

Linux AIX AIX and Linux 系统上的目录结构

AIX and Linux 系统上的 IBM MQ 目录结构可以映射到不同的文件系统，以便于管理，提高性能和提高可靠性。

使用 IBM MQ 的灵活目录结构以利用共享文件系统来运行多实例队列管理器。

使用命令 **crtmqm** *QM1* 来创建 第 108 页的图 35 中显示的目录结构，其中 R 是产品的发行版。 它是在 IBM MQ 系统上创建的队列管理器的典型目录结构。 为了清晰起见，省略了某些目录，文件和 .ini 属性设置，另一个队列管理器名称可能会被修改。 文件系统的名称在不同的系统上会有所不同。

在典型安装中，您创建的每个队列管理器都指向本地文件系统上的公共 log 和 qmgrs 目录。 在多实例配置中， log 和 qmgrs 目录位于与另一个 IBM MQ 安装共享的网络文件系统上。

第 108 页的图 35 显示了 AIX 上 IBM MQ v7.R 的缺省配置，其中 R 是产品的发行版。 有关备用多实例配置的示例，请参阅第 112 页的『AIX and Linux 系统上的示例目录配置』。

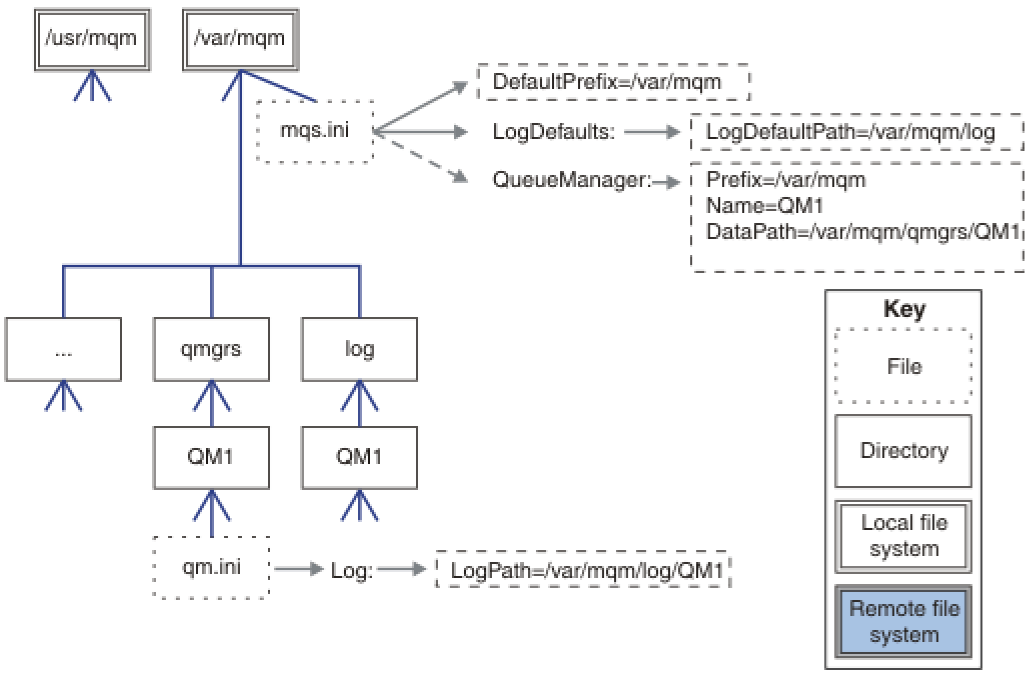


图 *35: AIX and Linux* 系统的缺省 *IBM MQ* 目录结构示例

缺省情况下，该产品安装到 AIX 上的 `/usr/mqm` 和其他系统上的 `/opt/mqm` 中。 工作目录将安装到 `/var/mqm` 目录中。

注: 如果在安装 IBM MQ 之前创建了 `/var/mqm` 文件系统，请确保 mqm 用户具有完整目录许可权，例如，文件方式 755。

注: `/var/mqm/errors` 目录应该是单独的文件系统，以防止队列管理器生成的 FFDC 填充包含 `/var/mqm` 的文件系统。

请参阅 在 AIX and Linux 系统上创建文件系统 以获取更多信息。

`log` 和 `qmgrs` 目录显示在其缺省位置中，如 `mqs.ini` 文件中的 `LogDefault` **路径** 和 `DefaultPrefix` 属性的缺省值所定义。 创建队列管理器时，缺省情况下将在 *`DefaultPrefix`*`/qmgrs` 中创建队列管理器数据目录，并在 *`LogDefaultPath`*`/log` 中创建日志文件目录。 `LogDefault` **路径** 和 `DefaultPrefix` 仅影响缺省情况下创建队列管理器和日志文件的情况。 队列管理器目录的实际位置保存在 `mqs.ini` 文件中，日志文件目录的位置保存在 `qm.ini` 文件中。

队列管理器的日志文件目录在 `LogPath` 属性的 `qm.ini` 文件中定义。 在 **crtmqm** 命令上使用 `-ld` 选项来设置队列管理器的 `LogPath` 属性; 例如， **`crtmqm`** `-ld` *`LogPath`* `QM1` 。 如果省略 `ld` 参数，那么将改为使用 `LogDefaultPath` 的值。

队列管理器数据目录在 `mqs.ini` 文件的 `QueueManager` 节中的 `DataPath` 属性中定义。 在 **crtmqm** 命令上使用 -md 选项为队列管理器设置 `DataPath`; 例如， **`crtmqm`** `- md` *`DataPath`* `QM1` 。 如果省略 md 参数，那么将改为使用 `DefaultPrefix` 或 `Prefix` 属性的值。 **前缀** 优先于 `DefaultPrefix`。

通常，在单个命令中同时指定日志和数据目录来创建 QM1 。

```
 crtmqm
-md DataPath -ld
LogPath QM1
```

当队列管理器停止时，您可以通过编辑 `qm.ini` 文件中的 `DataPath` 和 `LogPath` 属性来修改现有队列管理器的队列管理器日志和数据目录的位置。

`errors` 目录的路径 (与 `/var/mqm` 中所有其他目录的路径一样) 不可修改。 但是，这些目录可以安装在不同的文件系统上，也可以象征性地链接到不同的目录。

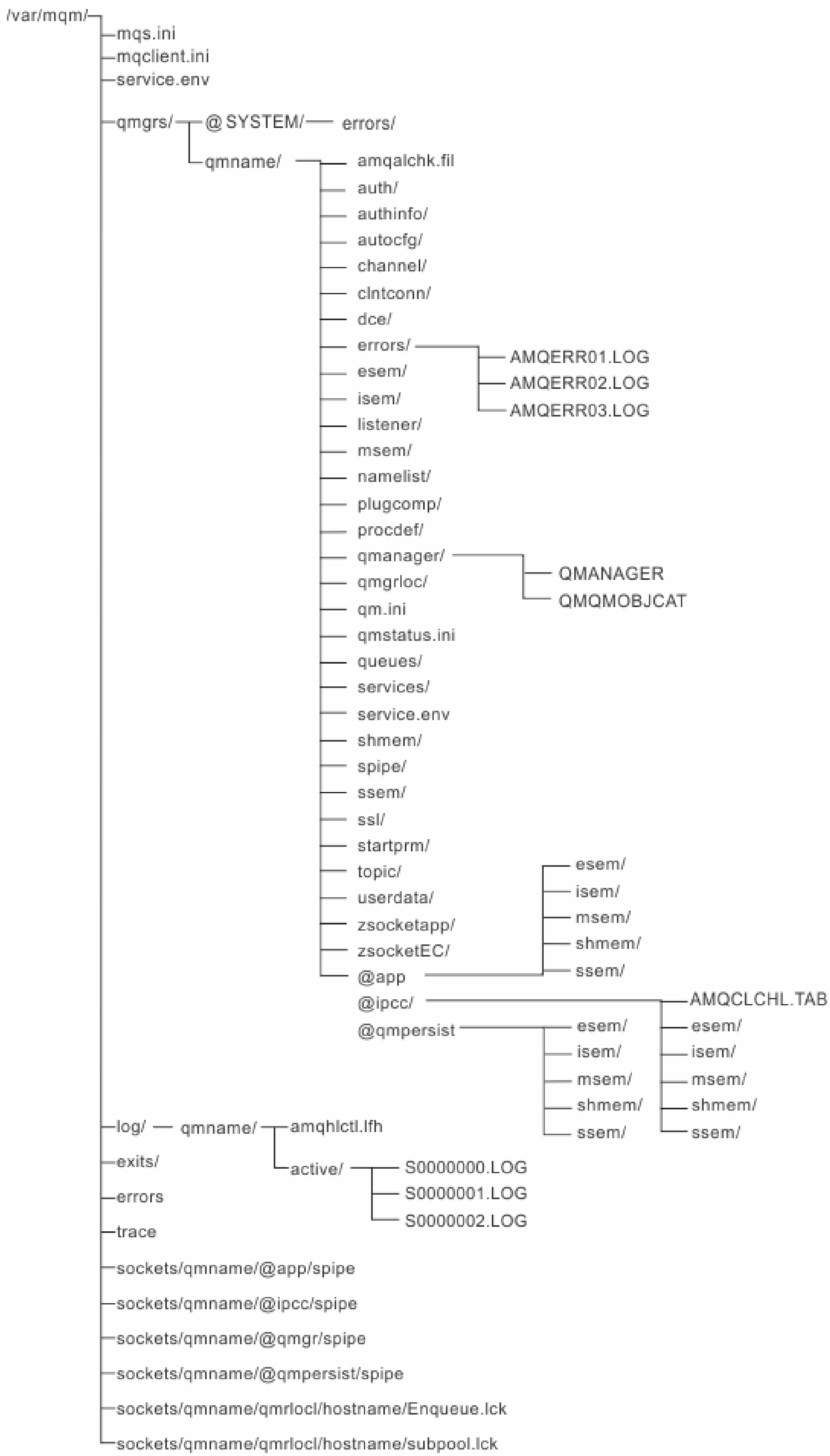
Linux AIX *AIX and Linux* 系统上的目录内容

与队列管理器关联的目录的内容。

有关产品文件位置的信息，请参阅 选择安装位置

有关备用目录配置的信息，请参阅 第 96 页的『多平台上的规划文件系统支持』。

V 9.2.0 在队列管理器已使用一段时间之后，以下目录结构代表 IBM MQ 。 您拥有的实际结构取决于在队列管理器上执行的操作。


/var/mqm/
mqs.ini
mqclient.ini
service.env
qmgrs/
@SYSTEM/
errors/
qmname/
amqalchk.fil
auth/
authinfo/
autocfg/
channel/
clntconn/
dce/
errors/
AMQERR01.LOG
AMQERR02.LOG
AMQERR03.LOG
esem/
isem/
listener/
msem/
namelist/
plugcomp/
procdef/
qmanager/
QMANAGER
QMQMOBJCAT
qmgrloc/
qm.ini
qmstatus.ini
queues/
services/
service.env
shmem/
spipe/
ssem/
ssl/
startprm/
topic/
userdata/
zsocketapp/
zsocketEC/
@app
esem/
isem/
msem/
shmem/
ssem/
@ipcc/
AMQCLCHL.TAB
esem/
isem/
msem/
shmem/
ssem/
@qmpersist
esem/
isem/
msem/
shmem/
ssem/
log/
qmname/
amqhlctl.lfh
active/
S0000000.LOG
S0000001.LOG
S0000002.LOG
exits/
errors
trace
sockets/qmname/@app/spipe
sockets/qmname/@ipcc/spipe
sockets/qmname/@qmgr/spipe
sockets/qmname/@qmpersist/spipe
sockets/qmname/qmrlocl/hostname/Enqueue.lck
sockets/qmname/qmrlocl/hostname/subpool.lck

/var/mqm/

/var/mqm 目录包含适用于整个 IBM MQ 安装而不是个别队列管理器的配置文件和输出目录。

表 *13: AIX and Linux* 上 */var/mqm* 目录的已记录内容

目录或文件名	内容
mqs.ini	IBM MQ 安装范围配置文件，在队列管理器启动时读取。 可使用 **AMQ_MQS_INI_LOCATION** 环境变量修改文件路径。 确保在运行 **strmqm** 命令的 shell 中设置并导出此命令。
mqclient.ini	IBM MQ MQI client 程序读取的缺省客户机配置文件。 可使用 **MQCLNTCF** 环境变量修改文件路径。
service.env	包含服务进程的机器作用域环境变量。 文件路径已固定。
个错误/	机器作用域错误日志和 FFST 文件。 目录路径已修正。 另请参阅 FFST: IBM MQ for UNIX 和 Linux 系统。
套接字/	包含每个队列管理器的信息，仅供系统使用。
跟踪/	跟踪文件。 目录路径已修正。
Web/	mqweb 服务器目录。
个出口/ **exits64/**	包含用户通道出口程序的缺省目录。 可在 mqs.ini 文件的 ApiExit 节中修改位置。

/var/mqm/qmgrs/*qmname*/

/var/mqm/qmgrs/*qmname*/ 包含队列管理器的目录和文件。 该目录已锁定，以供活动队列管理器实例独占访问。 可以在 mqs.ini 文件中直接修改目录路径，也可以使用 **crtmqm** 命令的 **md** 选项进行修改。

表 *14: AIX and Linux* 上 */var/mqm/qmgrs/qmname* 目录的已记录内容

目录或文件名	内容
qm.ini	队列管理器配置文件，在队列管理器启动时读取。
个错误/	队列管理器作用域错误日志。 *qmname* = @system 包含未知或不可用队列管理器的通道相关消息。
@ipcc/ AMQCLCHL.TAB	缺省客户机通道控制表，由 IBM MQ 服务器创建，并由 IBM MQ MQI client 程序读取。 可使用 **MQCHLLIB** 和 **MQCHLTAB** 环境变量修改文件路径。
QMANAGER	队列管理器对象文件: QMANAGER 队列管理器对象目录: QMQMOBJCAT

表 14: *AIX and Linux* 上 */var/mqm/qmgrs/qmname* 目录的已记录内容 *(*继续*)*	
目录或文件名	**内容**
authinfo/	队列管理器中定义的每个对象都与这些目录中的一个文件相关联。 文件名与定义名称大致匹配; 请参阅 了解 IBM MQ 文件名。
通道/	
clntconn/	
侦听器/	
名称列表/	
进程/	
队列/	
服务/	
主题/	
...	IBM MQ 使用的其他目录 (例如 @ipcc) 将仅由 IBM MQ 修改。
V 9.2.0 **用户数据/**	可用于存储应用程序的持久状态 (当将队列管理器移动到不同节点时， RDQM 可以使用此状态-请参阅 存储持久应用程序状态。)
V 9.2.0 ***DataPath*\autocfg**	用于自动配置

/var/mqm/log/*qmname*/

/var/mqm/log/*qmname*/ 包含队列管理器日志文件。 该目录已锁定，以供活动队列管理器实例独占访问。 可以在 qm.ini 文件中修改目录路径，也可以使用 **crtmqm** 命令的 **ld** 选项进行修改。

表 15: *AIX and Linux* 上 */var/mqm/log/qmname* 目录的已记录内容	
目录或文件名	**内容**
amqhlctl.lfh	日志控制文件。
活动/	此目录包含编号为 S0000000.LOG， S0000001.LOG， S0000002.LOG，依此类推。

/opt/mqm

缺省情况下， /opt/mqm 是大多数平台上的安装目录。 请参阅 第 93 页的『Multiplatforms 版上的磁盘空间需求』，以获取有关企业使用的一个或多个平台上的安装目录所需的空间量的更多信息。

Linux AIX *AIX and Linux* 系统上的示例目录配置

AIX and Linux 系统上备用文件系统配置的示例。

您可以通过各种方式定制 IBM MQ 目录结构，以实现许多不同的目标。

- 将 qmgrs 和 log 目录放在远程共享文件系统上以配置多实例队列管理器。
- 将单独的文件系统用于数据和日志目录，并将这些目录分配给不同的磁盘，以通过减少 I/O 争用来提高性能。
- 将更快的存储设备用于对性能影响更大的目录。 物理设备等待时间通常是持久消息传递性能中比本地还是远程安装设备更重要的因素。 以下列表显示哪些目录最敏感和最不敏感的性能。

  1. log
  2. qmgrs

3. 其他目录，包括 /usr/mqm

- 在分配给具有良好弹性的存储器 (例如冗余磁盘阵列) 的文件系统上创建 qmgrs 和 log 目录。
- 最好将公共错误日志存储在 var/mqm/errors 本地，而不是存储在网络文件系统上，以便可以记录与网络文件系统相关的错误。

第 113 页的图 36 是从中派生备用 IBM MQ 目录结构的模板。 在模板中，虚线表示可配置的路径。 在示例中，虚线将替换为与 AMQ_MQS_INI_LOCATION 环境变量以及 mqs.ini 和 qm.ini 文件中存储的配置信息相对应的实线。

注: 路径信息显示在 mqs.ini 或 qm.ini 文件中。 如果在 **crtmqm** 命令中提供路径参数，请省略队列管理器目录的名称: 队列管理器名称由 IBM MQ 添加到路径中。

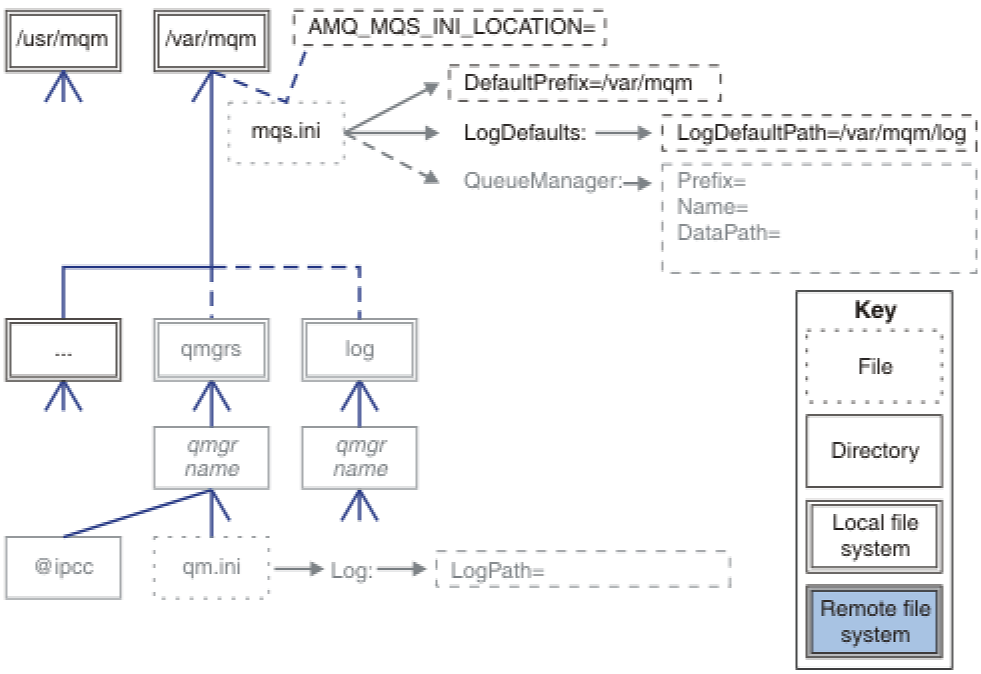


图 *36:* 目录结构模式模板

IBM MQ 的典型目录结构

第 114 页的图 37 是通过发出命令 **crtmqm** *QM1* 在 IBM MQ 中创建的缺省目录结构。

mqs.ini 文件具有 QM1 队列管理器的节，此节是通过引用 DefaultPrefix 的值创建的。 qm.ini 文件中的 Log 节具有 LogPath 的值，通过引用 mqs.ini 中的 LogDefaultPath 进行设置。

使用可选 **crtmqm** 参数可覆盖 DataPath 和 LogPath 的缺省值。

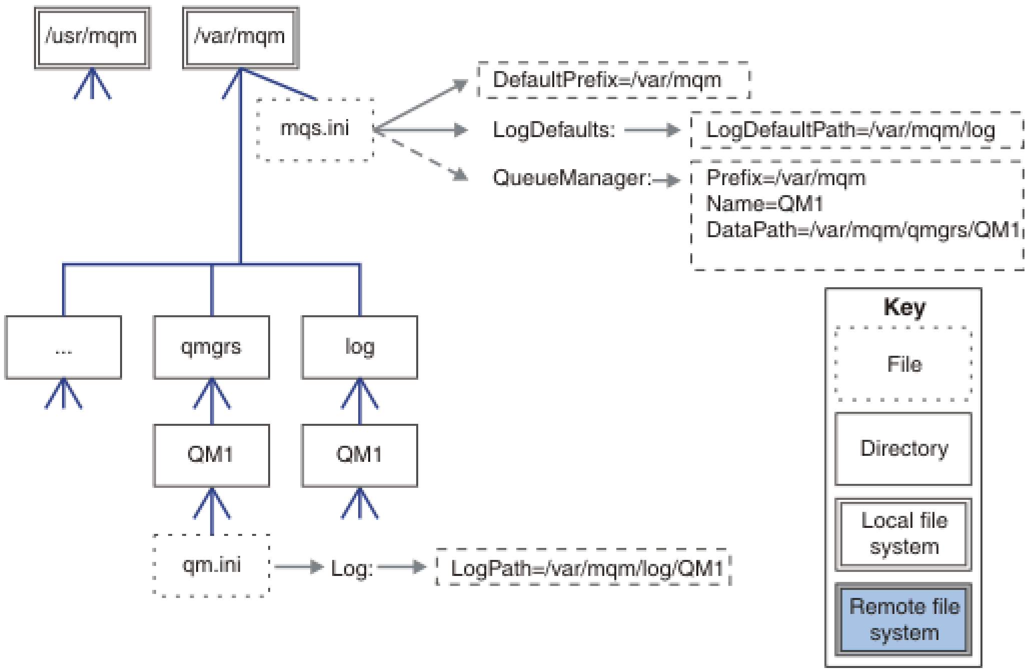


图 *37: AIX and Linux* 系统的缺省 *IBM MQ* 目录结构示例

共享缺省 qmgrs 和 log 目录)

第 115 页的『共享所有内容』的替代方法是单独共享 qmgrs 和 log 目录 (第 114 页的图 38)。 在此配置中，无需设置 AMQ_MQS_INI_LOCATION， 因为缺省 mqs.ini 存储在本地 /var/mqm 文件系统中。 文件和目录 (例如 mqclient.ini 和 mqserver.ini) 也不会共享。

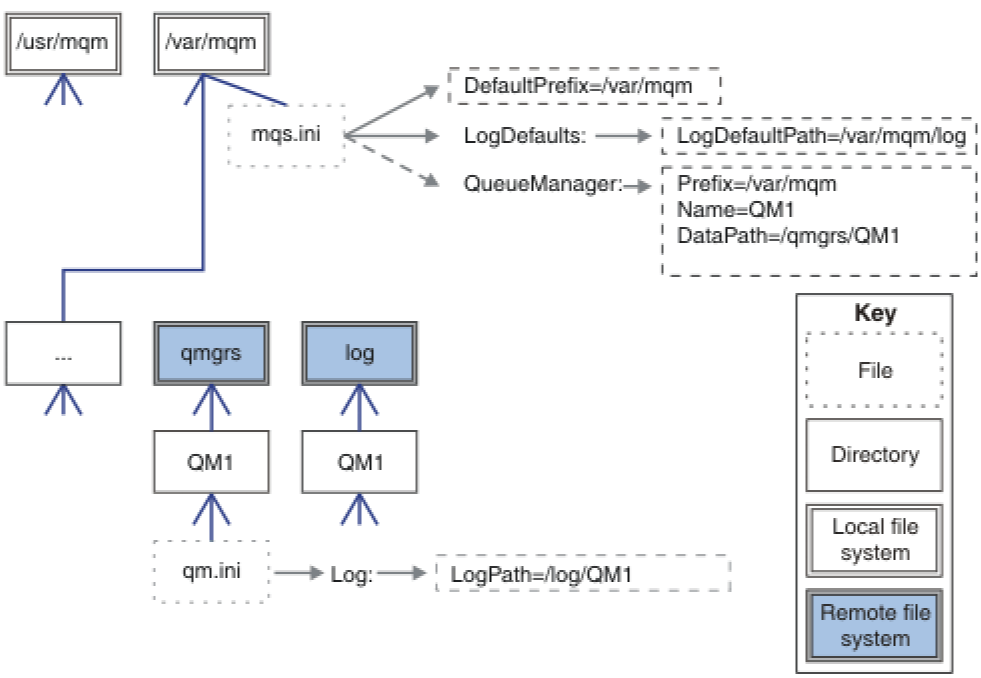


图 *38:* 共享 *qmgrs* 和 *log* 目录

共享名为 qmgrs 和 log 的目录

第 115 页的图 39 中的配置将 log 和 qmgrs 放置在名为 /ha 的公共命名远程共享文件系统中。 可以通过两种不同的方式创建相同的物理配置。

1. 设置 `LogDefaultPath=/ha`，然后运行命令 **crtmqm** - md */ha/qmgrs* QM1。 结果与 第 115 页的图 39 中的说明完全相同。
2. 保留缺省路径不变，然后运行命令 **crtmqm** - ld */ha/log* - md */ha/qmgrs* QM1。

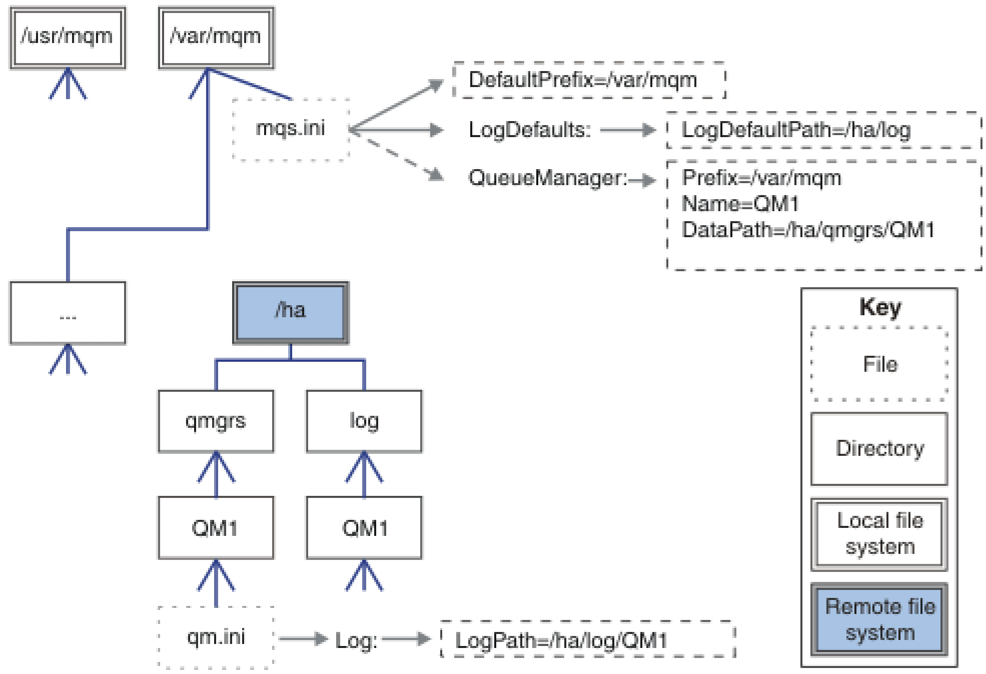


图 *39:* 共享名为 *qmgrs* 和 *log* 的目录

共享所有内容

第 116 页的图 40 是具有快速联网文件存储器的系统的简单配置。

将 `/var/mqm` 安装为远程共享文件系统。 缺省情况下，当您启动 QM1 时，它将查找 `/var/mqm`，在共享文件系统上查找并读取 `/var/mqm` 中的 `mqs.ini` 文件。 您可以将每个服务器上的 `AMQ_MQS_INI_LOCATION` 环境变量设置为指向不同的 `mqs.ini` 文件，而不是将单个 `/var/mqm/mqs.ini` 文件用于所有服务器上的队列管理器。

注: `/var/mqm/errors/` 中通用错误文件的内容在不同服务器上的队列管理器之间共享。

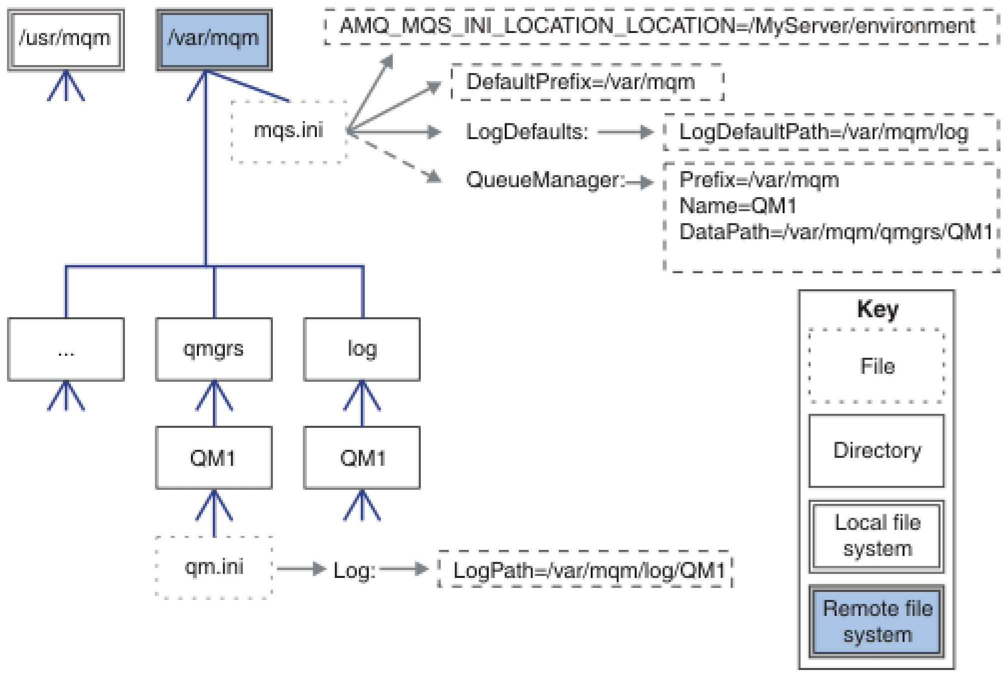


图 *40:* 共享所有内容

请注意，不能将此用于多实例队列管理器。 原因是多实例队列管理器中的每个主机都必须具有其自己的 `/var/mqm` 本地副本，以跟踪本地数据 (例如信号量和共享内存)。 无法在主机之间共享这些实体。

Windows Windows 系统上的目录结构

如何在 Windows 上查找队列管理器配置信息和目录。

IBM MQ for Windows 安装的缺省目录为:

程序目录
`C:\Program Files\IBM\MQ`

数据目录
`C:\ProgramData\IBM\MQ`

要点: Windows 对于 Windows 安装，会按照上述内容使用目录，除非存在先前的产品安装，该安装仍包含注册表项和/或队列管理器。 在此情况下，新安装将使用原有的数据目录位置。 有关更多信息，请参阅程序和数据目录位置。

如果要知道使用的是哪个安装目录和哪个数据目录，请运行 dspmqver 命令。

安装目录列示在 **InstPath** 字段中，数据目录列示在 **DataPath** 字段中。

例如，运行 **dspmqver** 命令将显示以下信息:

```
>dspmqver
Name:        IBM MQ
Version:     9.0.0.0
Level:       p900-L160512.4
BuildType:   IKAP - (Production)
Platform:    IBM MQ for Windows (x64 platform)
Mode:        64-bit
O/S:         Windows 7 Professional x64 Edition, Build 7601: SP1
InstName:    Installation1
InstDesc:
Primary:     Yes
InstPath:    C:\Program Files\IBM\MQ
DataPath:    C:\ProgramData\IBM\MQ
MaxCmdLevel: 900
LicenseType: Production
```

多实例队列管理器

要配置多实例队列管理器，必须将日志和数据目录放在联网存储器上，最好是放在与正在运行队列管理器实例的任何服务器不同的服务器上。

在 **crtmqm** 命令 **-md** 和 **-ld** 上提供了两个参数，以便更轻松地指定队列管理器数据和日志目录的位置。 指定 **-md** 参数的效果是四重:

1. `mqs.ini` 节 `QueueManager\`*QmgrName* 包含指向队列管理器数据目录的新变量 *DataPath*。 与 *Prefix* 变量不同，路径包含队列管理器目录的名称。
2. 存储在 mqs.ini 文件中的队列管理器配置信息将缩减为 *Name*， *Prefix*， *Directory* 和 *DataPath*。

Windows 目录内容

列出 IBM MQ 目录的位置和内容。

IBM MQ 配置具有三组主要文件和目录:

1. 可执行文件以及仅在应用维护时更新的其他只读文件。 例如:
   - 自述文件
   - IBM MQ Explorer 插件和帮助文件
   - 许可证文件

   第 117 页的表 16 中描述了这些文件。
2. 可能可修改的文件和目录并非特定于特定队列管理器。 第 118 页的表 17 中描述了这些文件和目录。
3. 特定于服务器上每个队列管理器的文件和目录。 第 118 页的表 18 中描述了这些文件和目录。

资源目录和文件

资源目录和文件包含用于运行队列管理器的所有可执行代码和资源。 特定于安装的 IBM MQ 配置注册表键中的变量 *FilePath* 包含资源目录的路径。

表 16: *FilePath* 目录中的目录和文件

文件路径	内容
FilePath\bin	命令和 DLL
FilePath\bin64	命令和 DLL (64 位)
FilePath\conv	数据转换表
FilePath\doc	向导帮助文件
FilePath\MQExplorer	Explorer 和 Explorer 帮助 Eclipse 插件
FilePath\gskit8	Global Security Kit
FilePath\java	Java 资源，包括 JRE
FilePath\licenses	许可证信息
FilePath\Non_IBM_License	许可证信息
FilePath\properties	内部使用
FilePath\Tivoli	内部使用
FilePath\tools	开发资源和样本
FilePath\web	在 IBM MQ Console 和 REST API 不可编辑文件的安装组件文件结构中进行了描述。
FilePath\Uninst	内部使用
FilePath\README.TXT	自述文件

不特定于队列管理器的目录

某些目录包含并非特定于特定队列管理器的文件，例如跟踪文件和错误日志。 *DefaultPrefix* 变量包含这些目录的路径。 *DefaultPrefix* 是 `AllQueueManagers` 节的一部分。

表 17: *DefaultPrefix* 目录中的目录和文件

文件路径	内容
DefaultPrefix\config	内部使用
DefaultPrefix\conv	ccsid_part2.tbl 和 ccsid.tbl data 转换控制文件，如 数据转换 中所述
DefaultPrefix\errors	非队列管理器错误日志， AMQERR *nn*.LOG
DefaultPrefix\exits	通道出口程序
DefaultPrefix\exits64	通道出口程序 (64 位)
DefaultPrefix\ipc	未使用
DefaultPrefix\qmgrs	描述位置 第 118 页的表 18
DefaultPrefix\trace	跟踪文件
DefaultPrefix\web	在用户可编辑文件的 IBM MQ Console 和 REST API 安装组件文件结构中描述
DefaultPrefix\amqmjpse.txt	内部使用

队列管理器目录

创建队列管理器时，将创建一组特定于队列管理器的新目录。

如果使用 **-md** *filepath* 参数创建队列管理器，那么路径将存储在 mqs.ini 文件的队列管理器节中的 *DataPath* 变量中。 如果在不设置 **-md** *filepath* 参数的情况下创建队列管理器，那么将在存储在 *DefaultPrefix* 中的路径中创建队列管理器目录，并将该路径复制到 mqs.ini 文件的队列管理器节中的 *Prefix* 变量中。

表 18: *DataPath* 和 *Prefix*\qmgrs\QmgrName 目录中的目录和文件

文件路径	内容
DataPath\@ipcc	AMQCLCHL.TAB(客户机连接表) 的缺省位置。
DataPath\authinfo	内部使用
DataPath\channel	
DataPath\clntconn	
DataPath\errors	错误日志， AMQERR *nn*.LOG

表 18: *DataPath* 和 *Prefix*\qmgrs*QmgrName* 目录中的目录和文件 (继续)	
文件路径	**内容**
DataPath\listener	内部使用
DataPath\namelist	
DataPath\plugcomp	
DataPath\procdef	
DataPath\qmanager	
DataPath\queues	
DataPath\services	
DataPath\ssl	
DataPath\startprm	
DataPath\topic	
DataPath\active	
DataPath\active.dat	
DataPath\amqalchk.fil	
DataPath\master	
DataPath\master.dat	
DataPath\qm.ini	队列管理器配置
DataPath\qmstatus.ini	队列管理器状态
V 9.2.0 *DataPath*\userdata	可用于存储应用程序的持久状态。
Prefix\qmgrs*QmgrName*	内部使用
Prefix\qmgrs\@SYSTEM	未使用
Prefix\qmgrs\@SYSTEM\errors	
V 9.2.0 *DataPath**autocfg*	用于自动配置

IBM i IBM i 上的目录结构

提供了 IFS 的描述，并且针对服务器，客户机和 Java 描述了 IBM MQ IFS 目录结构。

集成文件系统 (IFS) 是 IBM i 的一部分，支持类似于个人计算机 AIX and Linux 操作系统的流输入/输出和存储管理，同时针对存储在服务器中的所有信息提供集成结构。

在 IBM i 上，目录名称以字符 & (ampersand) (而不是字符 @ (at)) 开头。 例如， IBM i 上的 @system 是 &system。

IBM MQ 服务器的 IFS 根文件系统

安装 IBM MQ Server for IBM i 时，将在 IFS 根文件系统中创建以下目录。

ProdData:

概述

```
QIBM
   '-- ProdData
```

```
'-- mqm
'-- doc
'-- inc
'-- lib
'-- samp
'-- licenses
'-- LicenseDoc
'-- 5724H72_V8R0M0
```

/QIBM/ProdData/mqm
下面的子目录包含所有产品数据，例如 C++ 类，跟踪格式文件和许可证文件。 每次安装产品时，都会删除并替换此目录中的数据。

/QIBM/ProdData/mqm/doc
CL 命令的 "命令参考" 以 HTML 格式提供，并安装在此处。

/QIBM/ProdData/mqm/inc
用于编译 C 或 C++ 程序的头文件。

/QIBM/ProdData/mqm/lib
MQ 使用的辅助文件。

/QIBM/ProdData/mqm/samp
更多样本。

/QIBM/ProdData/mqm/License
许可证文件。 每种语言的两个文件命名为类似于 LA_ *xx* 和 LI_ *xx* ，其中 *xx* 是所提供的每种语言的 2 字符语言标识。

另外，以下目录存储许可协议文件:

/QIBM/ProdData/LicenseDoc/5724H72_V8R0M0
许可证文件。 这些文件的名称类似于 5724H72_V8R0M0_ *xx* ，其中 *xx* 是所提供的每种语言的 2 或 5 字符语言标识。

UserData:

概述

```
QIBM
   '-- UserData
      '-- mqm
      '-- errors
      '-- trace
      '-- qmgrs
      '-- &system
      '-- qmgrname1
      '-- qmgrname2
      '-- and so on
```

/QIBM/UserData/mqm
此目录下的子目录包含与队列管理器相关的所有用户数据。

安装产品时，将在目录 /QIBM/UserData/mqm/ 中创建 mqs.ini 文件 (除非先前安装已有此文件)。

创建队列管理器时，将在目录 /QIBM/UserData/mqm/qmgrs/ *QMGRNAME* /(其中 *QMGRNAME* 是队列管理器的名称) 中创建 qm.ini 文件。

删除产品时，将保留目录中的数据。

IBM MQ MQI client 的 IFS 根文件系统

安装 IBM MQ MQI client for IBM i 时，将在 IFS 根文件系统中创建以下目录:

ProdData:

概述

```
QIBM
   '-- ProdData

     '-- mqm
     '-- lib
```

/QIBM/ProdData/mqm
此目录下的子目录包含所有产品数据。 每次替换产品时，都会删除并替换此目录中的数据。

UserData:

概述

```
QIBM
   '-- UserData

     '-- mqm
     '-- errors
     '-- trace
```

/QIBM/UserData/mqm
此目录下的子目录包含所有用户数据。

IBM MQ Java 的 IFS 根文件系统

在 IBM i 上安装 IBM MQ Java 时，将在 IFS 根文件系统中创建以下目录:

ProdData:

概述

```
QIBM
   '-- ProdData

     '-- mqm
     '-- java
     '--samples
     '-- bin
     '-- lib
```

/QIBM/ProdData/mqm/java
下面的子目录包含所有产品数据，包括 Java 类。 每次替换产品时，都会删除并替换此目录中的数据。

/QIBM/ProdData/mqm/java/samples
下面的子目录包含所有样本 Java 类和数据。

由服务器和客户机安装创建的库

安装 IBM MQ 服务器或客户机将创建以下库:

- `QMQM`

  产品库。

- `QMQMSAMP`

  样本库 (如果选择安装样本)。

- `QMxxxx`

  仅服务器。

  每次创建队列管理器时， IBM MQ 都会自动创建一个相关联的库，其名称类似于 `QMxxxx`，其中 `xxxx` 派生自队列管理器名称。 此库包含特定于队列管理器的对象，包括日志和关联的接收方。 缺省情况下，此库

的名称派生自以 QM 字符作为前缀的队列管理器的名称。 例如，对于名为 TEST 的队列管理器，库将被称为 QMTEST。

注: 创建队列管理器时，可以根据需要指定其库的名称。 例如:

```
CRTMQM MQMNAME(TEST) MQMLIB(TESTLIB)
```

可以使用 WRKLIB 命令来列示 IBM MQ for IBM i 已创建的所有库。 针对队列管理器库，您将看到文本 `QMGR: QMGRNAME`。 命令格式为:

```
WRKLIB LIB(QM*)
```

删除产品时，将保留这些与队列管理器关联的库。

Multi 规划 MFT on Multiplatforms 的文件系统支持

IBM MQ Managed File Transfer MFT 代理程序可用于将数据传输到文件系统上的文件以及从中传输数据。 除此之外，可以将代理程序中运行的资源监视器配置为监视文件系统上的文件。

MFT 要求这些文件存储在支持锁定的文件系统上。 原因有二:

- 代理程序锁定文件以确保它在开始从中读取数据或向其写入数据后不会更改。
- 资源监视器锁定文件以检查当前是否没有其他进程在使用这些文件。

代理程序和资源监视器使用 Java 方法 **`FileChannel.tryLock()`** 来执行锁定，并且文件系统必须能够在请求时使用此调用来锁定文件。

要点: 以下文件系统不受支持，因为它们不满足 MFT 的技术要求:

- GlusterFS
- NFS V3

Multi 在 Multiplatforms 版上选择循环或线性日志记录

在 IBM MQ 中，可以选择循环或线性日志记录。 以下信息提供了这两种类型的概述。

循环日志记录的优点

循环日志记录的主要优点是:

- 易于管理。

  为工作负载正确配置循环日志记录后，无需进一步管理。 而对于线性日志记录，需要记录介质图像，并且需要归档或删除不再需要的日志扩展数据块。

- 性能更佳

  循环日志记录的性能优于线性日志记录，因为循环日志记录能够复用已格式化的日志扩展数据块。 而线性日志记录必须分配新的日志扩展数据块并对其进行格式化。

请参阅 管理日志 以获取更多信息。

线性日志记录的优点

线性日志记录的主要优点是，线性日志记录提供了防止更多故障的保护。

循环或线性日志记录都无法防止已损坏或已删除的日志，或者应用程序或管理员已删除的消息或队列。

线性日志记录 (但不是循环) 使受损对象能够恢复。 因此，线性日志记录提供了防止队列文件损坏或删除的保护，因为这些损坏的队列可以从线性日志中恢复。

循环和线性保护以防止电源丢失和通信故障，如 从电源丢失或通信故障中恢复中所述。

其他注意事项

选择线性还是循环取决于需要多少冗余。

选择更多冗余 (即线性日志记录) 的成本由性能成本和管理成本引起。

请参阅 日志记录类型 以获取更多信息。

AIX AIX 上的共享内存

如果某些应用程序类型由于 AIX 内存限制而无法连接，那么在大多数情况下，可以通过设置环境变量 EXTSHM=ON 来解决此问题。

AIX 上的某些 32 位进程可能会迂到操作系统限制，这会影响它们连接到 IBM MQ 队列管理器的能力。 到 IBM MQ 的每个标准连接都使用共享内存，但与其他 UNIX 平台不同， AIX 仅允许 32 位进程连接 11 个共享内存集。

大多数 32 位进程不会迂到此限制，但是具有高内存需求的应用程序可能无法连接到 IBM MQ ，原因码为 2102: MQRC_RESOURCE_PROBLEM。 以下应用程序类型可能会看到此错误:

- 在 32 位 Java 虚拟机中运行的程序
- 使用大型或超大型内存模型的程序
- 连接到多个队列管理器或数据库的程序
- 独立连接到共享内存集的程序

AIX 为 32 位进程提供了扩展共享内存功能，允许它们连接更多共享内存。 要使用此功能运行应用程序，请在启动队列管理器和程序之前导出环境变量 EXTSHM=ON。 EXTSHM=ON 功能部件在大多数情况下可防止此错误，但它与使用 shmctl 函数的 SHM_SIZE 选项的程序不兼容。

IBM MQ MQI client 应用程序和所有 64 位进程不受此限制影响。 无论是否已设置 EXTSHM ，它们都可以连接到 IBM MQ 队列管理器。

Linux AIX IBM MQ 和 UNIX System V IPC 资源

队列管理器使用一些 IPC 资源。 使用 **`ipcs -a`** 来查找正在使用的资源。

此信息仅适用于在 AIX and Linux 系统上运行的 IBM MQ 。

IBM MQ 使用 System V 进程间通信 (IPC) 资源 (*semaphores* 和 共享内存段) 以在系统组件之间存储和传递数据。 这些资源由连接到队列管理器的队列管理器进程和应用程序使用。 IBM MQ MQI clients 不使用 IPC 资源，但 IBM MQ 跟踪控制除外。 使用 UNIX 命令 **`ipcs -a`** 来获取有关机器上当前正在使用的 IPC 资源的数量和大小的完整信息。

Linux AIX IBM MQ 和 UNIX 进程优先级

设置进程优先级 *nice* 值时的良好实践。

此信息仅适用于在 AIX and Linux 系统上运行的 IBM MQ 。

如果在后台运行进程，那么调用 shell 可以为该进程提供更高的 *nice* 值 (从而降低优先级)。 这可能具有一般的 IBM MQ 性能影响。 在高度紧张的情况下，如果有许多就绪可运行的线程处于较高优先级，而有些线程处于较低优先级，那么操作系统调度特性会使较低优先级的线程失去处理器时间。

与队列管理器 (例如 **`runmqlsr`**) 关联的独立启动的进程具有与它们关联的队列管理器相同的 *nice* 值是好的做法。 确保 shell 未将更高的 *nice* 值分配给这些后台进程。 例如，在 ksh 中，使用设置 “`set +o bgnice`” 来阻止 ksh 提高后台进程的 *nice* 值。 您可以通过检查 “ps -efl” 列表的 *NI* 列来验证正在运行的进程的 *nice* 值。

此外，使用与队列管理器相同的 *nice* 值来启动 IBM MQ 应用程序进程。 如果它们使用不同的 *nice* 值运行，那么应用程序线程可能会阻止队列管理器线程，反之亦然，从而导致性能下降。

z/OS 在 z/OS 上规划 IBM MQ 环境

规划 IBM MQ 环境时，必须考虑数据集，页集， Db2，耦合设施的资源需求以及日志记录和备份设施的需求。 使用本主题来规划运行 IBM MQ 的环境。

在规划 IBM MQ 体系结构之前，请熟悉基本 IBM MQ for z/OS 概念，请参阅 IBM MQ for z/OS 概念中的主题。

在规划队列管理器时，您可能需要与组织中的不同人员一起工作。 早期让这些人参与通常是一个好主意，因为改变控制程序可能需要很长时间。 他们还可以告诉您需要配置哪些参数 IBM MQ for z/OS。

例如，您可能需要使用以下命令:

- 存储器管理员，用于确定队列管理器数据集的高级限定符，并为队列管理器数据集分配足够的空间。
- z/OS 系统程序员，用于将 IBM MQ 子系统定义到 z/OS 并对 IBM MQ for z/OS 库进行 APF 授权。
- 网络管理员，用于确定应该将哪些 TCP/IP 堆栈和端口用于 IBM MQ for z/OS。
- 安全性管理员，用于设置对队列管理器数据集， IBM MQ for z/OS 资源的安全概要文件和 TLS 证书的访问权。
- Db2 管理员在配置队列共享组时设置 Db2 表。

相关概念

IBM MQ 技术概述

相关任务

第 5 页的『规划 IBM MQ 体系结构』

规划 IBM MQ 环境时，请考虑 IBM MQ 为单个和多个队列管理器体系结构以及点到点和发布/预订消息传递样式提供的支持。 还要规划资源需求以及日志记录和备份工具的使用。

配置 z/OS

管理 IBM MQ for z/OS

z/OS 规划队列管理器

在设置队列管理器时，您的规划应允许队列管理器增长，以便队列管理器满足企业的需求。

配置队列管理器的最佳方法是执行以下步骤:

1. 配置基本队列管理器
2. 配置执行队列管理器到队列管理器通信以及远程客户机应用程序通信的通道启动程序
3. 如果要加密和保护消息，请配置 Advanced Message Security
4. 如果要使用基于 IBM MQ 的文件传输，请配置 z/OS 的 Managed File Transfer。
5. 如果要使用管理或消息传递 REST API 或 MQ Console 从 Web 浏览器管理 IBM MQ， 请配置 mqweb 服务器。

一些企业在其环境中有数以千计的队列管理器。 您需要立即并在五年内考虑 IBM MQ 网络。

在 z/OS 上，某些队列管理器每秒处理数千条消息，并且每秒记录超过 100 MB。 如果期望的卷非常高，那么可能需要考虑具有多个队列管理器。

在 z/OS 上， IBM MQ 可以作为队列共享组 (QSG) 的一部分运行，其中消息存储在耦合设施中，并且队列共享组中的任何队列管理器都可以访问这些消息。 如果要在队列共享组中运行，需要考虑需要多少队列管理器。 通常，每个 LPAR 都有一个队列管理器。 您还可以有一个队列管理器定期备份 CF 结构。

对配置的一些更改很容易执行，例如定义新队列。 有些更难，例如使日志和页集更大; 有些配置无法更改，例如队列管理器的名称或队列共享组名。

MP16 性能 SupportPac 中提供了性能和调整信息。

命名约定

您需要具有队列管理器数据集的命名约定。

许多企业在装入库的名称中使用发行版号，以此类推。 您可能需要考虑将别名 MQM.SCSQAUTH 指向当前正在使用的版本 (例如 MQM.V900.SCSQAUTH) ，以便在迁移到新版本的 IBM MQ 时不必更改 CICS，批处理和 IMS JCL。

您可以在 z/OS UNIX System Services 中使用符号链接来引用当前正在使用的 IBM MQ 版本的安装目录。

队列管理器所使用的数据集 (日志，页集和 JCL 库) 需要一个命名约定，以简化安全概要文件的创建以及数据集到 SMS 存储类的映射， SMS 存储类用于控制数据集在磁盘上的位置及其属性。

请注意，将 IBM MQ 版本放入页集或日志的名称中不是一个好主意。 有一天，您可能会迁移到新版本，并且数据集将具有 "错误" 名称。

应用

您需要了解业务应用程序以及配置 IBM MQ 的最佳方法。 例如，如果应用程序具有提供恢复和重复功能的逻辑，那么非持久消息可能足够好。 如果希望 IBM MQ 处理恢复，那么需要使用持久消息并在同步点中放入和获取消息。

您需要将队列与不同的业务事务隔离。 如果一个业务应用程序的队列已满，那么您不希望这会影响其他业务应用程序。 如果可能，请隔离不同页集和缓冲池或结构中的队列。

您需要了解消息的概要文件。 对于许多应用程序，队列只有几条消息。 其他应用程序可以在白天构建队列，并在夜间进行处理。 通常只有几条消息的队列，如果存在问题并且未处理消息，那么可能需要保留许多小时的消息。 您需要调整 CF 结构和页集的大小以实现预期的峰值容量。

后配置

配置队列管理器 (和组件) 后，您需要规划:

- 正在备份页集。
- 正在备份对象的定义。
- 自动备份任何 CF 结构。
- 监视 IBM MQ 消息，并在检测到问题时执行操作。
- 正在收集 IBM MQ 统计信息数据。
- 监视资源使用情况，例如虚拟存储器和每小时记录的数据量。 通过此操作，您可以查看资源使用率是否正在增加以及是否需要执行操作，例如，设置新的队列管理器

z/OS 在 z/OS 上规划存储和性能需求

您必须为 IBM MQ 系统设置切合实际且可实现的存储和性能目标。 使用本主题可帮助您了解影响存储器和性能的因素。

本主题包含有关 IBM MQ for z/OS 的存储和性能需求的信息。 它包含以下部分:

- IBM MQ 的 z/OS 性能选项
- 确定 z/OS 工作负载管理重要性和速度目标
- 第 126 页的『库存储器』
- 第 126 页的『System LX 使用情况』
- 第 127 页的『地址空间存储器』
- 第 130 页的『磁盘存储器』

请参阅 第 130 页的『在何处查找有关存储和性能需求的更多信息』 以获取更多信息。

IBM MQ 的 z/OS 性能选项

通过工作负载管理，您可以定义性能目标，并为每个目标分配业务重要性。 您以业务术语定义工作目标，系统将决定应该向工作提供多少资源 (例如处理器和存储器) 以实现其目标。 工作负载管理根据您提供的目标来控制分派优先级。 工作负载管理会根据需要提高或降低优先级以实现指定的目标。 因此，您无需对系统中每个工作的确切优先级进行微调，而是可以将重点放在业务目标上。

三种目标是:

响应时间
: 您希望处理工作的速度

执行速度
: 准备就绪时应以多快的速度运行工作，而不会因处理器，存储器， I/O 访问和队列延迟而延迟

任意
: 没有性能目标的低优先级工作的类别

响应时间目标适用于最终用户应用程序。 例如， CICS 用户可以将工作负载目标设置为响应时间目标。 对于 IBM MQ 地址空间，速率目标更合适。 队列管理器中完成的少量工作将计入此速率目标，但此工作对于性能至关重要。 队列管理器完成的大部分工作都计入最终用户应用程序的性能目标。 通道启动程序地址空间所完成的大部分工作都将计入其自身的速率目标。 通道启动程序完成的 IBM MQ 消息的接收和发送通常对于使用这些消息的业务应用程序的性能很重要。

确定 z/OS 工作负载管理重要性和速度目标

请参阅 第 126 页的『确定 z/OS 工作负载管理重要性』 以获取更多信息。

库存储器

V 9.2.0 必须为产品库分配磁盘存储器。 准确数字取决于您的配置，并且应该同时包含目标库和分发库以及 SMP/E 库。

IBM MQ for z/OS 使用的目标库使用 PDSE 格式。 确保未在综合系统外部共享任何 PDSE 目标库。 有关所需库及其大小和所需格式的更多信息，请参阅 "程序目录"。 有关程序目录的下载链接，请参阅 IBM MQ for z/OS 程序目录 PDF 文件。

System LX 使用情况

每个定义的 IBM MQ 子系统在 IPL 时保留一个系统链接索引 (LX) ，并在队列管理器启动时保留多个非系统链接索引。 当队列管理器停止并重新启动时，将复用系统链接索引。 同样，分布式排队保留一个非系统链接索引。 如果 z/OS 系统不太可能定义了足够的系统 LX ，那么您可能需要考虑这些保留的系统 LX。

如果需要，可以通过在 SYS1.PARMLIB 成员 IEASYSxx 中设置 *NSYSLX* 参数来增加系统 LX 的数量。

z/OS 确定 *z/OS* 工作负载管理重要性

有关工作负载管理和通过服务定义定义目标的完整信息，请参阅 z/OS 产品文档。

本主题建议如何设置与系统中其他重要工作相关的 z/OS 工作负载管理重要性和速度目标。 有关更多信息，请参阅 *z/OS MVS Planning: Workload Management* 。

需要定义具有高优先级的队列管理器地址空间，因为它提供子系统服务。 通道启动程序是应用程序地址空间，但通常具有高优先级，以确保发送到远程队列管理器的消息不会延迟。 Advanced Message Security (AMS) 还提供子系统服务，需要以高优先级进行定义。

使用以下服务类:

缺省 SYSSTC 服务类

- VTAM 和 TCP/IP 地址空间
- IRLM 地址空间 (IRLMPROC)

注: VTAM， TCP/IP 和 IRLM 地址空间的分派优先级必须高于所有 DBMS 地址空间，其连接的地址空间及其下级地址空间。 不允许工作负载管理将 VTAM， TCP/IP 或 IRLM 的优先级降低到 (或低于) 其他 DBMS 地址空间的优先级

对于具有您定义的名称 (例如 PRODREGN) 的服务类， 1 的高速目标和重要性如下所示:

- IBM MQ 队列管理器，通道启动程序和 AMS 地址空间
- Db2 (除 Db2 建立的存储过程地址空间以外的所有地址空间)

- CICS (所有区域类型)
- IMS (除 BMP 以外的所有区域类型)

高速目标有利于确保对所有这些地址空间尽可能快地执行启动和重新启动。

CICS 和 IMS 区域的速率目标仅在启动或重新启动期间很重要。 事务开始运行后，工作负载管理会忽略 CICS 或 IMS 速率目标，并根据区域中运行的事务的响应时间目标来分配优先级。 这些事务目标应反映他们实施的业务应用程序的相对优先级。 它们的重要性值通常为 2。 使用 IBM MQ 的任何批处理应用程序都应具有类似的速度目标和重要性，以反映它们实现的业务应用程序的相对优先级。 通常，重要性和速度目标将小于 PRODREGN 的目标。

z/OS 地址空间存储器

使用本主题来获取有关 IBM MQ 组件的地址空间需求的基本指南。

存储需求可分为以下类别:

- 公共存储器
- 队列管理器专用区域存储器使用情况
- 通道启动程序存储器使用情况

在 64 位地址空间中，有一个称为 “bar” 的虚拟线路，用于标记 2GB 地址。 条形将位于 2GB 地址 (称为 “低于条形”) 下的存储器与位于 2GB 地址 (称为 “高于条形”) 上的存储器分隔开。 界线下的存储器使用 31 位可寻址性，界线上的存储器使用 64 位可寻址性。

可以使用 JCL 上的 REGION 参数指定 31 位存储器的限制，使用 MEMLIMIT 参数指定高于 bar 存储器的限制。 这些指定的值可由 MVS 出口覆盖。

注意: 引入了对系统工作方式的更改。 现在，跨系统扩展服务 (XES) 将每个连接的高虚拟存储器中的 4GB 存储器分配到序列化列表结构。

在此更改之前，已在数据空间中分配此存储器。 应用此 APAR 后，根据 IBM MQ 计算存储器使用情况的方式，可能会发出消息 CSQY225E 和 CSQY224I，指示 **队列管理器缺少高于界线的本地存储器**。

您还会在消息 CSQY220I 中看到上述条形值的增加

有关更多信息，请参阅 IBM 支持文档 2017139。

建议的区域大小

下表显示了区域大小的建议值。

表 19: *JCL* 区域大小的建议定义

定义设置	系统
队列管理器	REGION=0M, MEMLIMIT=3G
通道启动程序	REGION=0M

公共存储器

每个 IBM MQ for z/OS 子系统都具有以下近似的存储需求:

- CSA 4KB
- ECSA 800KB，加上在 CSQ6SYSP 系统参数宏的 TRACTBL 参数中指定的跟踪表的大小。 有关更多信息，请参阅 使用 CSQ6SYSP。

此外，每个并发 IBM MQ 逻辑连接需要大约 5 KB 的 ECSA。 当任务结束时，其他 IBM MQ 任务可以复用此存储器。 在队列管理器关闭之前， IBM MQ 不会释放存储器，因此您可以通过将最大并发逻辑连接数乘以 5KB 来计算所需的最大 ECSA 量。 并发逻辑连接数是以下项数的总和:

- 与 IBM MQ 连接但未断开连接的批处理， TSO， z/OS UNIX System Services， IMS 和 Db2 存储过程地址空间 (SPAS) 区域中的任务 (TCB)。
- 已发出 IBM MQ 请求但未终止的 CICS 个事务
- 已创建的 JMS 连接，会话， TopicSessions 或 QueueSessions (用于绑定连接)，但尚未销毁或收集垃圾。
- 活动 IBM MQ 通道。

您可以使用 ACELIM 配置参数设置公共存储器的限制，由与队列管理器的逻辑连接使用。 ACELIM 控件主要与 Db2 存储过程导致对 IBM MQ 队列执行操作的站点相关。

从存储过程驱动时，每个 IBM MQ 操作都可能导致与队列管理器建立新的逻辑连接。 大型 Db2 工作单元 (例如，由于表装入) 可能会导致对公共存储器的过度需求。

ACELIM 旨在限制公共存储器使用，并通过限制系统中的连接数来保护 z/OS 系统。 仅应在已标识为使用过多 ECSA 存储器的队列管理器上进行设置。 请参阅 使用 CSQ6SYSP 中的 ACELIM 部分，以获取更多信息。

要为 ACELIM 设置值，请首先确定当前由 ACELIM 值控制的子池中的存储量。 此信息位于由统计 CLASS (3) 跟踪生成的 SMF 115 子类型 5 记录中。

IBM MQ 可以使用 SupportPac MP1B 来格式化 SMF 数据。 由 ACELIM 控制的子池中正在使用的字节数显示在 STGPOOL DD 中标题为 *ACE/PEB* 的行上。

有关 SMF 115 统计信息记录的更多信息，请参阅解释 IBM MQ 性能统计信息。

将正常值增大足够的边距，以便为增长和工作负载峰值提供空间。 将新值除以 1024， 以生成用于 ACELIM 配置的最大存储大小 (以 KB 为单位)。

通道启动程序通常需要最多使用 160KB 的 ECSA。

队列管理器专用区域存储器使用情况

IBM MQ for z/OS 可以将高于 2GB 条形的存储器用于某些内部控制块。 您可以在此存储器中具有缓冲池，这使您有可能在有足够存储空间的情况下配置更大的缓冲池。 通常，缓冲池是使用高于 2GB 条的存储器的主要内部控制块。

每个缓冲池大小在队列管理器初始化时确定，当连接使用该缓冲池的页集时，将为该缓冲池分配存储器。 新参数 LOCATION (ABOVE | BELOW) 用于指定缓冲区的分配位置。 可以使用 ALTER BUFFPOOL 命令来动态更改缓冲池的大小。

要在界线 (64 位) 以上的存储器上使用，可以在队列管理器 JCL 中的 **`EXEC PGM=CSQYASCP`** 参数上指定 MEMLIMIT 参数的值 (例如 MEMLIMIT=3G)。 您的安装可能设置了缺省值。

您应该指定 MEMLIMIT 并指定合理的存储大小，而不是 MEMLIMIT = NOLIMIT， 以防止潜在问题。 如果指定 NOLIMIT 或非常大的值，那么具有较大大小的 ALTER BUFFPOOL 命令可以使用所有可用的 z/OS 虚拟存储器，这将导致在系统中进行页面调度。 您可能需要与 z/OS 系统程序员讨论 MEMLIMIT 的值，以防可以使用的存储量存在系统范围的限制。

从 MEMLIMIT=3G 开始，并在需要增加缓冲池大小时增大此大小。

将 MEMLIMIT 的值计算为 2GB 加上条形上方缓冲池的大小，向上取整为最接近的 GB。 将 MEMLIMIT 设置为最小值 3GB，并在需要增加缓冲池大小时根据需要增大此限制。

例如，对于使用 LOCATION ABOVE 配置的 2 缓冲池，缓冲池 1 具有 10,000 个缓冲区，缓冲池 2 具有 50,000 个缓冲区。 高于界线的内存使用量等于 60,000 (缓冲区总数) * 4096 = 245,760,000 字节 = 234.375MB。 所有缓冲池 (无论位置如何) 都将使用 64 位存储器来进行控制结构。 随着缓冲池数和这些池中的缓冲区数增加，这可能会变得显着。 每个缓冲区需要额外 200 字节的 64 位存储器。 对于具有 10 个缓冲池的配置，每个缓冲池具有 20,000 个缓冲区，需要: 200 * 10 * 20,000 = 40,000,000 (相当于 40MB)。 您可以为 MEMLIMIT 大小指定 3GB， 这将允许扩展范围 (40MB + 200MB + 2GB， 向上舍入到 3GB)。

对于某些配置，使用由实存储器永久支持其缓冲区的缓冲池可能会有显着的性能优势。 您可以通过为缓冲池的 PAGECLAS 属性指定 FIXED4KB 值来实现此目的。 但是，仅当 LPAR 上有足够的可用实存储器时，您才应该执行此操作，否则其他地址空间可能会受到影响。 有关何时应该将 FIXED4KB 值用于 PAGECLAS 的信息，请参阅 IBM MQ 支持 Pac MP16: IBM MQ for z/OS -容量规划和调整

在使用高于界线的存储器之前，应与 z/OS 系统程序员进行讨论，以确保有足够的辅助存储器用于峰值时间使用，并有足够的实际存储器需求来防止页面调度。

注: 为了处理增加的虚拟存储器，可能必须增大内存转储数据集的大小。

使缓冲池如此大，以至于存在 MVS 页面调度可能会对性能产生负面影响。 您可以考虑使用没有页面的较小缓冲池，通过 IBM MQ 将消息移动到页集以及从页集移动消息。

您可以从 CSQY220I 消息监视地址空间存储器使用情况，该消息指示 2GB 条上方和下方正在使用的专用区域存储量以及剩余的存储量。

通道启动程序存储器使用情况

您必须考虑通道启动程序存储器使用情况的两个区域:

- 专用区域
- 记帐和统计信息

专用区域存储器使用情况

您应该为 CHINIT 指定 REGION=0M， 以允许它使用低于 bar 存储器的最大值。 可用于通道启动程序的存储器限制了 CHINIT 可拥有的并发连接数。

每个通道在通道启动程序地址空间中使用大约 170KB 的扩展专用区域。 如果传输大于 32KB 的消息，那么将按消息大小增加存储器。 在以下情况下，将释放增加的存储空间:

- 发送或客户机通道需要小于当前缓冲区大小一半的 10 条连续消息。
- 发送或接收脉动信号。

将释放存储器以在语言环境中复用，但是 z/OS 虚拟存储器管理器不会将其视为可用。 这意味着通道数的上限取决于消息大小和到达模式以及单个用户系统对扩展专用区域大小的限制。 在许多系统上，通道数的上限可能大约为 9000， 因为扩展区域大小不可能超过 1.6GB。 使用大于 32KB 的消息大小会减少系统中的最大通道数。 例如，如果传输长度为 100MB 的消息，并且假定扩展区域大小为 1.6GB， 那么最大通道数为 15。

通道启动程序跟踪将写入数据空间。 数据空间存储器的大小由 **TRAXTBL** 参数控制。 请参阅 ALTER QMGR。

记帐和统计信息存储使用情况

您应该允许通道启动程序访问至少 256MB 以上的虚拟存储器。 您可以通过指定 `MEMLIMIT=256M` 来执行此操作。

If you do not set the MEMLIMIT parameter in the channel initiator JCL, you can set the amount of virtual storage above the bar using the MEMLIMIT parameter in the SMFPRMxx member of SYS1.PARMLIB, or from the IEFUSI exit.

如果设置 MEMLIMIT 以将以上条形存储器限制在所需级别以下，那么通道启动程序将发出 CSQX124E 消息和类 4 记帐和统计信息跟踪不可用。

管理 MEMLIMIT 和 REGION 大小

其他机制 (例如， SYS1.PARMLIB 的 SMFPRMxx 成员中的 **MEMLIMIT** 参数或 IEFUSI 出口) 可能会在安装时用于为 z/OS 地址空间提供高于界线的缺省虚拟存储量。 请参阅 界线上的内存管理， 以获取有关限制界线上的存储器的完整详细信息。

共享消息数据集 (SMDS) 缓冲区和 MEMLIMIT

使用共享消息数据集运行消息传递工作负载时，可以通过调整 DSBUFS 和 DSBLOCK 属性来实现两个级别的优化。

SMDS 缓冲区使用的以上条形队列管理器存储量为 DSBUFS x DSBLOCK。 这意味着缺省情况下， 100 x 256KB (25MB) 用于队列管理器中的每个 CFLEVEL (5) 结构。

虽然此值并不太高，但如果您的企业或企业有许多 CFSTRUCT，那么其中一些可能会为缓冲池分配高值 MEMLIMIT，有时它们具有深度索引队列，因此总起来，它们可能会耗尽界线上的存储空间。

z/OS 磁盘存储器

在规划日志数据集， Db2 存储器，耦合设施存储器和页面数据集的磁盘存储器需求时，请使用本主题。

与存储管理员一起确定将队列管理器数据集放在何处。 例如，存储管理员可以为您提供特定 DASD 卷，或针对不同数据集类型的 SMS 存储类，数据类和管理类。

- 日志数据集必须位于 DASD 上。 这些日志具有较高的 I/O 活动，响应时间短，不需要备份。
- 归档日志可以在 DASD 或磁带上。 创建后，除非在异常情况下 (例如从备份中恢复页集)，否则可能永远不会再次读取这些页集。 它们应该具有较长的保留日期。
- 页集可能具有从低到中的活动，应该定期备份。 在高使用系统上，它们应该每天备份两次。
- BSDS 数据集应该每天备份; 它们没有很高的 I/O 活动。

所有数据集都与 Db2 所使用的数据集相似，并且类似的维护过程可用于 IBM MQ。

有关如何规划数据存储器的详细信息，请参阅以下部分:

- **日志和归档存储器**

  第 144 页的『我需要多长时间来保留归档日志』 描述了如何根据 IBM MQ 系统处理的消息量以及将活动日志卸载到归档数据集的频率来确定活动日志和归档数据集所需的存储量。
- **Db2 存储器**

  第 159 页的『Db2 存储器』 描述了如何确定 Db2 需要多少存储器用于 IBM MQ 数据。
- **耦合设施存储器**

  第 151 页的『定义耦合设施资源』 描述了如何确定使耦合设施结构具有多大的大小。
- **页集和消息存储**

  第 130 页的『规划页集和缓冲池』 描述了如何根据应用程序交换的消息大小，这些消息的数量以及创建或交换这些消息的速率来确定页面数据集所需的存储量。

z/OS 在何处查找有关存储和性能需求的更多信息

使用本主题作为参考，以查找有关存储和性能需求的更多信息。

您可以从以下来源找到更多信息:

表 *20:* 在何处查找有关存储需求的更多信息

Topic	**在哪里查找信息**
系统参数	使用 CSQ6SYSP 和 定制队列管理器
安装 IBM MQ 所需的存储器	程序目录。 有关程序目录的下载链接，请参阅 IBM MQ for z/OS 程序目录 PDF 文件。
IEALIMIT 和 IEFUSI 出口	*MVS* 安装出口，可从 zSeries Web 站点 z/OS 因特网库获取。
最新信息	IBM MQ SupportPac Web 站点 IBM MQ 和其他项目区域的 SupportPacs。
通过服务定义进行工作负载管理和定义目标	*z/OS MVS* 规划*:* 工作负载管理

z/OS 规划页集和缓冲池

用于帮助您规划页面数据集和缓冲池的初始数量和大小的信息。

本主题包含以下几部分:

规划页面集

页集使用情况

对于短命消息，通常在页集上使用很少的页，并且只有在启动时，检查点期间或关闭时，对数据集的 I/O 很少或没有。

对于长期存在的消息，通常会将包含消息的那些页面写出到磁盘。 此操作由队列管理器执行，以减少重新启动时间。

通过将短命消息放在不同的页集和不同的缓冲池中，将短命消息与长命消息分开。

页集数

使用多个大页面集可以使 IBM MQ 管理员的角色更容易，因为这意味着您需要更少的页面集，从而使队列到页面集的映射更简单。

使用多个较小的页集有许多优点。 例如，它们需要较少的时间进行备份，并且可以在备份和重新启动期间并行执行 I/O。 但是，请考虑这样做会给 IBM MQ 管理员的角色增加很大的性能成本，该管理员需要将每个队列映射到大量页集中的一个页集。

定义至少五个页面集，如下所示:

- 为对象定义保留的页集 (页集为零)
- 系统相关消息的页集
- 性能关键型长寿命消息的页集
- 性能关键型短期消息的页集
- 所有其他消息的页集

第 134 页的『定义缓冲池』 说明了以此方式在页集上分发消息的性能优势。

页集大小

在页面集中定义足够的空间，以用于预期的峰值消息容量。 请考虑任何意外的峰值容量，例如，由于队列服务器程序未在运行而导致消息生成的情况。 您可以通过分配具有辅助扩展数据块的页集，或者通过启用动态页集扩展来执行此操作。 有关更多信息，请参阅第 133 页的『启用动态页集扩展』。 很难使一个页集变小，因此通常最好分配一个更小的页集，并允许它在需要时展开。

规划页集大小时，请考虑可能生成的所有消息，包括非应用程序消息数据。 例如，触发消息，事件消息和应用程序请求的任何报告消息。

页集的大小确定从备份复原时恢复页集所花费的时间，因为大型页集需要更长时间才能复原。

注: 页集的恢复还取决于队列管理器处理自备份以来写入的日志记录所花费的时间; 此时间段由备份频率确定。 有关更多信息，请参阅第 160 页的『规划备份和恢复』。

注: 大于 4 GB 的页集需要使用 SMS 扩展可寻址性。

V 9.2.0 规划 z/OS 数据集加密

您可以将 z/OS 数据集加密功能应用于在 IBM MQ for z/OS 9.1.4 或更高版本上运行的队列管理器的页集。

必须使用 EXTENDED 属性和数据集密钥标签来分配这些页集，以确保对数据进行 AES 加密。

请参阅 使用数据集加密在 IBM MQ for z/OS 上静态数据的机密性 部分。 for more information.

计算页面集的大小

对于队列管理器对象定义 (例如，队列和进程) ，计算存储需求很简单，因为这些对象具有固定大小并且是永久对象。 但是，对于消息，由于以下原因，计算更为复杂:

- 消息大小各不相同。
- 消息是暂时的。
- 异步进程将定期回收已检索的消息所占用的空间。

如果需要，可以创建大于 4 GB 的大页集，用于在网络停止时为消息提供额外容量。 无法修改现有页集。 相反，必须创建具有扩展可寻址性和扩展格式属性的新页面集。 新页集的物理大小必须与旧页集的物理大小相同，然后必须将旧页集复制到新页集。 如果需要向后迁移，那么不得更改页集 0。 如果小于 4 GB 的页集已足够，那么无需执行任何操作。

页集为零

为对象定义保留页集 0。

对于页集零，所需的存储空间为:

```
   (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)
```

将此值除以 4096 ，以确定要在集群中为页集数据集指定的记录数。

您不需要允许存储在共享存储库中的对象，但必须允许存储或复制到页集零的对象 (处置为 GROUP 或 QMGR 的对象)。

您可以创建的对象总数受页集 0 的容量限制。 可定义的本地队列数限制为 524 287。

页集 01-99

对于页集 01-99 ，每个页集所需的存储空间由存储在该页集中的消息的数量和大小决定。 (共享队列上的消息不会存储在页集上。)

将此值除以 4096 ，以确定要在集群中为页集数据集指定的记录数。

计算消息的存储需求

本部分描述了如何将消息存储在页面上。 了解这一点可以帮助您计算必须为消息定义的页集存储量。 要计算页集上所有消息所需的大致空间，必须考虑映射到页集的所有队列的最大队列深度以及这些队列上消息的平均大小。

注: 本节中给出的结构和控制信息的大小可能会在主要发行版之间发生变化。 有关特定于 IBM MQ 发行版的详细信息，请参阅 SupportPac MP16 - WebSphere MQ， 用于 z/OS 容量规划和调整 和 IBM MQ 系列-性能报告

您必须允许消息 "获取" 可能由于 IBM MQ 无法控制的原因 (例如，由于通信协议存在问题) 而延迟。 在这种情况下，消息的 "put" 速率可能远远超过 "get" 速率。 这可能会导致页集中存储的消息数大量增加，从而导致所需存储大小增加。

页集中的每个页的长度为 4096 字节。 允许固定头信息，每个页面都有 4057 字节的空间可用于存储消息。

计算每条消息所需的空间时，首先必须考虑的是消息是否适合一个页面 (短消息) ，或者是否需要拆分为两个或多个页面 (长消息)。 以这种方式拆分消息时，必须允许在空间计算中使用其他控制信息。

对于空间计算，消息可以表示为以下内容:

消息头部分包含消息描述符和其他控制信息，其大小因消息大小而异。 消息数据部分包含所有实际消息数据以及任何其他头 (例如，传输头或 IMS 网桥头)。

页面集控制信息至少需要两个页面，这通常小于消息所需的总空间的 1%。

短消息

将短消息定义为适合一个页面的消息。

从 IBM WebSphere MQ 7.0.1 开始，小消息存储在每个页面上。

长消息

如果消息数据的大小大于 3596 字节，但不大于 4 MB ，那么会将消息归类为长消息。 当显示长消息时， IBM MQ 会将消息存储在一系列页面上，并以存储短消息的方式存储指向这些页面的控制信息。 如 第 133 页的图 41 中所示:

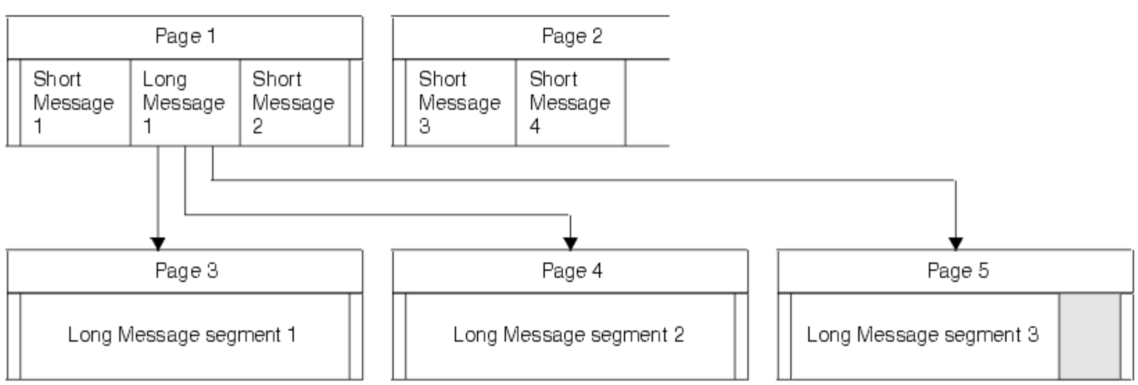


图 *41: IBM MQ* 如何在页集上存储长消息

非常长的消息

非常长的消息是大小大于 4 MB 的消息。 存储这些内容，以便每个 4 MB 使用 1037 个页面。 任何余数都以与长消息相同的方式存储，如上文所述。

z/OS 启用动态页集扩展

当队列管理器正在运行时，可以动态扩展页集。 一个页集可以有 123 个扩展数据块，并且可以分布在多个磁盘卷上。

每次页集展开时，都会使用新的数据集扩展数据块。 当需要时，队列管理器将继续扩展页集，直到达到最大扩展数据块数为止，或者直到没有更多存储器可用于在符合条件的卷上分配。

当页集扩展由于上述其中一个原因而失败时，队列管理器将该页集标记为不再进行扩展尝试。 可以通过将页集更改为 EXPAND (SYSTEM) 来重置此标记。

当分配页集中 90% 的现有空间时，将以异步方式对所有其他页集活动进行页集扩展。

页集扩展进程将对新分配的扩展数据块进行格式化，并使其可供队列管理器使用。 但是，在格式化整个扩展数据块之前，没有任何空间可供使用。 这意味着扩展很大程度上可能需要一些时间，如果应用程序在扩展完成之前填充剩余的 10% 的页面集，那么放置应用程序可能会 "阻塞"。

样本 thlqual.SCSQPROC(CSQ4PAGE) 显示如何定义辅助扩展数据块。

要控制新扩展数据块的大小，请使用 DEFINE PSID 和 ALTER PSID 命令的 EXPAND 关键字的下列其中一个选项:

- USER
- SYSTEM
- 无

USER

使用分配页集时指定的辅助扩展数据块大小。 如果未指定值，或者如果指定了值 0 ，那么不会发生动态页集扩展。

当页中的空间已使用 90% 时，将发生页集扩展，并与其他页集活动异步执行。

这可能导致一次扩展超过一个范围。

请考虑以下示例: 分配主扩展数据块为 100,000 页的页集和辅助扩展数据块为 5000 页的页集。 放置需要 9999 页的消息。 如果页集已使用 85,000 页，那么写入消息将跨越 90% 的完整边界 (90,000 页)。 此时，将另一个辅助扩展数据块分配给主扩展数据块 100,000 页，将页集大小设置为 105,000 页。 剩下的 4999 页消息继续写。 当已用页面空间达到 94,500 页 (这是 105,000 页的更新页集大小的 90%) 时，将分配另一个 5000 页扩展数据块，使页集大小达到 110,000 页。 在 MQPUT 结束时，页集已展开两次，使用了 94,500 页。 未使用第二个页集扩展中的任何页，尽管已分配这些页。

在重新启动时，如果先前使用的页集已被较小的数据集代替，那么此数据集会扩展，直到它达到先前使用的数据集大小。 只需要一个扩展数据块达到此大小。

SYSTEM

忽略定义页集时指定的辅助扩展数据块大小。 相反，队列管理器设置的值大约是当前页集大小的 10%。 该值向上取整为最接近的 DASD 柱面。

如果未指定值，或者如果指定了值 0 ，那么仍可能发生动态页集扩展。 队列管理器设置的值大约是当前页集大小的 10%。 新值将根据 DASD 的特征向上舍入。

当页集中的空间大约使用 90% 时，将发生页集扩展，并与其他页集活动一起异步执行。

在重新启动时，如果先前使用的页集已被较小的数据集代替，那么此数据集会扩展，直到它达到先前使用的数据集大小。

无

将不会执行进一步的页集扩展。

相关参考

变更 PSID
DEFINE PSID
显示使用情况

定义缓冲池

使用本主题来帮助规划您应该定义的缓冲池数及其设置。

本主题分为以下部分:

决定要定义的缓冲池数

最初应该定义四个缓冲池:

缓冲池 0

用于对象定义 (在页集零中) 和性能关键型系统相关消息队列，例如 SYSTEM.CHANNEL.SYNCQ 队列和 SYSTEM.CLUSTER.COMMAND.QUEUE 和 SYSTEM.CLUSTER.REPOSITORY.QUEUE 队列。

但是，如果要使用大量通道或集群，请务必考虑 调整缓冲池特征 中的 第 136 页的『7』 点。

将其余三个缓冲池用于用户消息。

缓冲池 1

用于重要的长期消息。

长生存期消息是那些在系统中保留超过两个检查点的消息，此时会将这些消息写出到页集。 如果您有许多长期存在的消息，那么此缓冲池应该相对较小，以便页集 I/O 均匀分布 (每当缓冲池变为 85% 已满时，就会将较旧的消息写出到 DASD 中)。

如果缓冲池太大，并且缓冲池从未达到 85% 已满，那么页集 I/O 将延迟到检查点处理。 这可能会影响整个系统的响应时间。

如果您只期望使用几条长寿消息，请定义此缓冲池，以使其足够大以容纳所有这些消息。

缓冲池 2

用于性能关键型短期消息。

通常存在高度的缓冲区复用，使用的缓冲区很少。 但是，您应该使此缓冲池大以允许意外的消息累积，例如，当服务器应用程序发生故障时。

缓冲池 3

用于所有其他 (通常是性能非关键) 消息。

诸如死信队列 SYSTEM.COMMAND.* 队列和 SYSTEM.ADMIN.* 也可以将队列映射到缓冲池 3。

如果存在虚拟存储器约束，并且缓冲池需要更小，那么缓冲池 3 是减小大小的第一个候选者。

在以下情况下，您可能需要定义其他缓冲池:

- 如果已知某个特定队列需要隔离，可能是因为它在不同时间表现出不同的行为。
  - 在不同情况下，此类队列可能需要最佳性能，或者需要隔离，以便不会对缓冲池中的其他队列产生负面影响。
  - 可以将每个此类队列隔离到其自己的缓冲池和页集中。
- 出于服务等级原因，您希望将不同的队列集相互隔离。
  - 然后，每组队列可能需要两种类型的缓冲池 1 或 2 中的一种或两种，如 缓冲池设置的建议定义中所述，需要创建特定类型的多个缓冲池。

决定每个缓冲池的设置

如果您正在使用 第 135 页的『决定要定义的缓冲池数』中描述的四个缓冲池，那么 缓冲池设置的建议定义 将提供两组缓冲池大小的值。

第一套适用于测试系统，另一套适用于生产系统或最终将成为生产系统的系统。 在所有情况下，都使用 **LOCATION**(*ABOVE*) 属性定义缓冲池

表 *21:* 缓冲池设置的建议定义

定义设置	测试系统	生产系统
BUFFPOOL 0	1 050 个缓冲区	50 000 个缓冲区
BUFFPOOL 1	1 050 个缓冲区	20 000 个缓冲区
BUFFPOOL 2	1 050 个缓冲区	50 000 个缓冲区
BUFFPOOL 3	1 050 个缓冲区	20 000 个缓冲区

如果需要超过建议的四个缓冲池，请选择最准确地描述缓冲池中队列的预期行为的缓冲池 (1 或 2) ，并使用缓冲池设置的建议定义中的信息对其进行大小调整。

确保 MEMLIMIT 设置得足够高，以便所有缓冲池都可以位于条形上方。

在预期负载下监视缓冲池的性能

您可以通过分析缓冲池性能统计信息来监视缓冲池的使用情况。 特别是，应确保缓冲池足够大，以便 QPSTSOS， QPSTSTSTLA 和 QPSTDMC 的值保持为零。

有关更多信息，请参阅 缓冲区管理器数据记录。

调整缓冲池特征

如果需要，请使用以下点从 第 135 页的『决定每个缓冲池的设置』调整缓冲池设置。

使用来自 第 136 页的『在预期负载下监视缓冲池的性能』 的性能统计信息作为指导。

1. 如果要从较低版本的 IBM MQ 进行迁移，请仅在具有更多可用实存储器时更改现有设置。
2. 通常，更大的缓冲池更有利于性能，如果缓冲池位于条形上方，那么它们可以大得多。

   但是，在任何时候都应该有足够的实存储器可用，以便缓冲池驻留在实存储器中。 与大型缓冲池相比，具有较小的不会导致页面调度的缓冲池更好。

   此外，没有任何一点具有大于使用它的页集的总大小的缓冲池，尽管如果可能发生页集扩展，那么应将其考虑在内。
3. 针对每个缓冲池设置一个页集，因为这提供了更好的应用程序隔离。
4. 如果有足够的实存储器，使您的缓冲池永远不会被操作系统调出，请考虑在缓冲池中使用页固定缓冲区。

   如果缓冲池可能经历大量 I/O ，那么这一点尤为重要，因为这将节省与在 I/O 之前对缓冲区进行页面修复以及之后对其进行页面修复相关的 CPU 成本。
5. 将缓冲池定位到条形上方有几个好处，即使这些缓冲池足够小，足以容纳在条形下方。 这些字段为：
   - 31 位虚拟存储器约束释放-例如，用于公共存储器的更多空间。
   - 如果在大量使用缓冲池时需要意外增加该缓冲池的大小，那么对队列管理器及其工作负载的影响和风险较小，方法是将更多的缓冲区添加到已高于界线的缓冲池，而不是将缓冲池移动到高于界线的位置，然后添加更多的缓冲区。
6. 调整缓冲池零和短生存期消息的缓冲池 (缓冲池 2) ，以便永不超过 15% 的可用阈值 (即 QPSTCBSL 除以 QPSTNBUF 始终大于 15%)。 如果超过 15% 的缓冲区保持可用，那么在正常操作期间可以很大程度上避免对使用这些缓冲池的页集进行 I/O 操作，尽管超过两个检查点的消息将写入页集。

   注意: 这些参数的最佳值取决于各个系统的特征。 给定的值仅用作准则，可能不适合您的系统。
7. SYSTEM.* 队列的深度，例如 SYSTEM.CHANNEL.SYNCQ，如果有足够的存储空间可用，那么可以将其放入自己的缓冲池中。

IBM MQ SupportPac MP16 - WebSphere MQ ，用于 z/OS 容量规划和调整 提供有关调整缓冲池的更多信息。

z/OS 规划日志记录环境

使用本主题来规划 IBM MQ 所使用的日志和日志归档的数量，大小和位置。

日志用于:

- 写入有关持久消息的恢复信息
- 使用持久消息记录有关工作单元的信息
- 记录有关对象更改的信息，例如定义队列
- 备份 CF 结构

以及其他内部信息。

IBM MQ 日志记录环境是使用系统参数宏来指定选项建立的，例如: 是具有单活动日志还是双活动日志，要用于归档日志卷的介质以及要具有的日志缓冲区数。

在 创建引导程序和日志数据集 和 定制系统参数模块中描述了这些宏。

注: 如果您正在使用队列共享组，请确保使用 SHAREOPTIONS (2 3) 定义引导程序和日志数据集。

本节包含有关以下主题的信息:

z/OS 日志数据集定义

使用本主题来决定日志数据集的最适当配置。

本主题包含帮助您回答以下问题的信息:

- 您的安装应该使用单日志记录还是双日志记录?
- 您需要多少活动日志数据集?
- 第 138 页的『活动日志应该有多大?』
- 活动日志放置
- V 9.2.0 第 139 页的『使用 z/OS 数据集加密的活动日志加密』

您的安装应该使用单日志记录还是双日志记录?

通常，您应该使用双重日志记录进行生产，以最大限度降低丢失数据的风险。 如果您希望测试系统反映生产情况，那么两者都应该使用双日志记录，否则测试系统可以使用单日志记录。

使用单个日志记录数据将写入一组日志数据集。 使用双日志记录数据写入两组日志数据集，因此在一个日志数据集出现问题 (例如，数据集被意外删除) 时，可以使用另一组日志中的等效数据集来恢复数据。

使用双重日志记录时，您需要的 DASD 数量是使用单一日志记录时的两倍。

如果您正在使用双重日志记录，那么还需要使用双重 BSD 和双重归档，以确保为数据恢复提供足够的资源。

双活动日志记录增加了较小的性能成本。

注意: 使用磁盘镜像技术 (例如， Metro Mirror) 不一定是双日志记录和双 BSDS 的替代方法。 如果意外删除了镜像数据集，那么两个副本都将丢失。

如果使用持久消息，单一日志记录可以将最大容量增加 10-30% ，还可以缩短响应时间。

单次日志记录使用 2-310 个活动日志数据集，而双次日志记录使用 4-620 个活动日志数据集来提供相同数量的活动日志。 因此，单次日志记录会减少记录的数据量，如果您的安装受到 I/O 限制，那么这可能很重要。

您需要多少活动日志数据集?

日志数取决于队列管理器的活动。 对于具有低吞吐量的测试系统，三个活动日志数据集可能适用。 对于高吞吐量生产系统，您可能需要最大可用日志数，因此，如果卸载日志存在问题，那么您有更多时间来解决问题。

您必须至少有三个活动日志数据集，但最好定义更多数据集。 例如，如果填充日志所花费的时间可能接近峰值装入期间归档日志所花费的时间，请定义更多日志。

注: 页集和活动日志数据集适合驻留在扩展地址卷 (EAV) 的扩展寻址空间 (EAS) 部分中，并且从 z/OS V1.12 开始，归档日志数据集也可以驻留在 EAS 中。

您还应该定义更多日志，以抵消日志归档中可能的延迟。 如果在磁带上使用归档日志，请留出安装磁带所需的时间。

请考虑有足够的活动日志空间来保留一天的数据，以防系统由于缺少 DASD 或无法写入磁带而无法归档。 如果所有活动日志都已填满，那么 IBM MQ 无法处理持久消息或事务。 拥有足够的活动日志空间非常重要。

可以动态定义新的活动日志数据集，以此最大限度降低归档延迟或问题的影响。 可以使用 **DEFINE LOG** 命令快速使新数据集联机，以避免由于活动日志中缺少空间而导致队列管理器 "失速"。

如果要定义超过 31 个活动日志数据集，那么必须配置日志记录环境以使用 V 2 格式的 BSDS。 使用版本 2 格式 BSDS 后，可以为每个日志副本环定义多达 310 个活动日志数据集。 请参阅 第 145 页的『计划增大最大可寻址日志范围』，以获取有关如何转换为版本 2 格式 BSDS 的信息。

您可以通过运行打印日志映射实用程序 (CSQJU004) 或从队列管理器初始化期间发出的 CSQJ034I 消息来确定队列管理器是使用 V 2 还是更高版本的 BSDS。 CSQJ034I 消息中日志 RBA 范围 FFFFFFFFFFFFFFFF 的末尾指示正在使用版本 2 或更高版本的格式 BSDS。 CSQJ034I 消息中的日志 RBA 范围结束 0000FFFFFFFFFFFF 指示正在使用版本 1 格式的 BSDS。

当队列管理器使用版本 2 或更高版本格式 BSDS 时，可以使用 **DEFINE LOG** 命令将超过 31 个活动日志数据集动态添加到日志副本环。

活动日志应该有多大?

从 IBM MQ 8.0 开始，归档到磁盘时支持的最大活动日志大小为 4 GB。 在产品的前发行版中，归档到磁盘时支持的最大活动日志大小为 3 GB。

归档到磁带时，最大活动日志大小为 4 GB。

您应该为生产和测试系统创建至少 1 GB 大小的活动日志。

要点: 在分配数据集时需要小心，因为 IDCAMS 会向上舍入您分配的大小。

要分配 3 GB 日志，请指定下列其中一个选项:

- 柱面 (4369)
- 兆字节 (3071)
- 磁道数 (65535)
- 记录 (786420)

其中任何一个分配 2.99995 GB。

要分配 4GB 日志，请指定下列其中一个选项:

- 柱面 (5825)
- 兆字节 (4095)
- 磁道数 (87375)
- 记录 (1048500)

其中任何一个分配 3.9997 GB。

使用条带分割数据集 (其中数据集分布在多个卷中) 时，将在用于条带分割的每个 DASD 卷上分配指定的大小值。 因此，如果要使用 4 GB 日志和四个卷进行条带分割，那么应指定:

- CYLinder (1456)
- 兆字节 (1023)

设置这些属性将分配 4 * 1456 = 5824 个柱面或 4 * 1023 = 4092 兆字节。

注: 使用扩展格式数据集时，支持条带分割。 这通常由存储管理器设置。

有关执行该过程的信息，请参阅 增大活动日志的大小 。

活动日志放置

您应该与存储管理团队一起为队列管理器设置存储池。 您需要考虑:

- 命名约定，因此队列管理器使用正确的 SMS 定义。
- 活动日志和归档日志所需的空间。 您的存储池应该有足够的空间用于整天的活动日志。
- 性能和对故障的恢复能力。

出于性能原因，您应该考虑对活动日志数据集进行条带分割。 I/O 分布在多个卷中，减少了 I/O 响应时间，从而导致更高的吞吐量。 请参阅前面的文本，以获取有关在使用条带分割时分配活动日志大小的信息。

您应该使用来自 RMF 或类似产品的报告来查看 I/O 统计信息。 每月 (或更频繁地) 对 IBM MQ 数据集执行这些统计信息的复审，以确保不会由于数据集的位置而发生延迟。

在某些情况下，可能会有大量 IBM MQ 页集 I/O ，如果它们位于同一 DASD 上，那么这可能会影响 IBM MQ 日志性能。

如果使用双重日志记录，请确保每组活动日志和归档日志保持分开。 例如，在不同的 DASD 子系统上或在不同的设备上分配它们。

这将降低在其中一个卷损坏或损坏时两者都丢失的风险。 如果日志的两个副本都丢失，那么数据丢失的概率很高。

创建新的活动日志数据时，设置应该使用 CSQJUFMT 对其进行预先格式化。 如果未预先格式化日志，那么队列管理器会在第一次使用日志时对其进行格式化，这会影响性能。

对于具有大型旋转磁盘的旧 DASD ，您必须小心使用哪些卷以获得最佳性能。

借助现代 DASD (其中数据分布在许多 PC 大小的磁盘上) ，您无需担心使用了哪些卷。

存储管理器应检查企业 DASD 以查看并解决任何性能问题。 为了实现可用性，您可能希望在一个 DASD 子系统上使用一组日志，而在另一个 DASD 子系统上使用双日志。

使用 z/OS 数据集加密的活动日志加密

V 9.2.0

您可以将 z/OS 数据集加密功能应用于在 IBM MQ for z/OS 9.1.4 或更高版本上运行的队列管理器的活动日志数据集。

必须使用 EXTENDED 属性和数据集密钥标签来分配这些活动日志数据集，以确保数据经过 AES 加密。

请参阅 使用数据集加密在 IBM MQ for z/OS 上静态数据的机密性 部分。 for more information.

z/OS V 9.2.0 将 *MetroMirror* 与 *IBM MQ* 配合使用

IBM Metro Mirror(先前称为同步对等远程复制 (PPRC)) 是两个存储子系统之间的同步复制解决方案，其中在认为写操作已完成之前，将在主卷和辅助卷上完成写操作。 Metro Mirror 可用于在发生存储子系统故障时不需要数据丢失的环境中。

受支持的数据集类型

可以使用 Metro Mirror 复制以下所有 IBM MQ 数据集类型。 但是，复制的确切内容取决于企业的可用性需求:

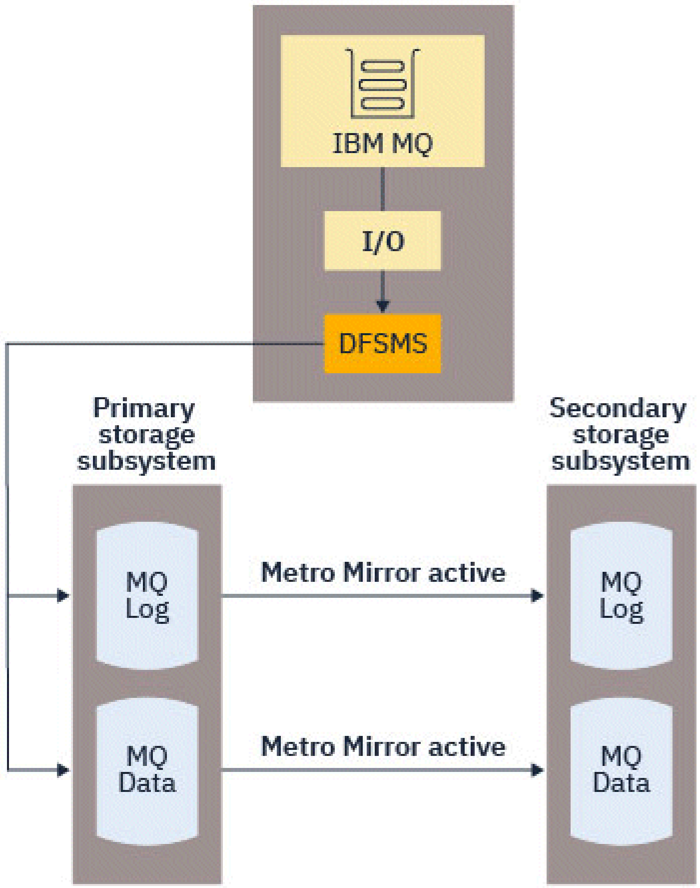
- 活动记录
- 归档日志
- 引导数据集 (BSDS)
- 页面集合

- 共享消息数据集 (SMDS)
- 用于配置的数据集，例如，在 MSTR JCL 上的 CSQINP* DD 卡中

将 zHyperWrite 与 IBM MQ 活动日志配合使用

对使用 Metro Mirror 复制的数据集进行写操作时，将首先对主卷进行写操作，然后将其复制到辅助卷。 此复制由存储子系统完成，对于发出写操作的应用程序 (例如 IBM MQ) 是透明的。

下图说明了此过程。

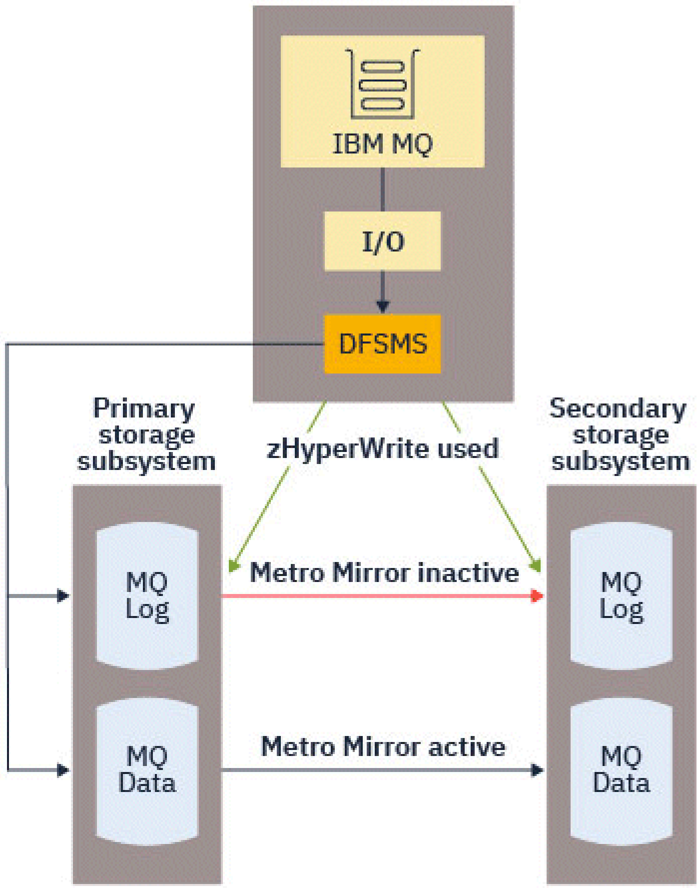


由于在写入返回到 IBM MQ 之前需要完成对主存储子系统和辅助存储子系统的写入，因此使用 Metro Mirror 可能会影响性能。 您需要将此性能影响与使用 Metro Mirror 的可用性优势进行平衡。

IBM MQ 活动日志对使用 Metro Mirror 的性能影响最为敏感。 IBM MQ 允许将 zHyperWrite 与活动日志配合使用，以帮助降低此性能影响。

zHyperWrite 是一种存储子系统技术，可与 z/OS 配合使用，以降低对使用 Metro Mirror 复制的数据集进行的写操作对性能的影响。 使用 zHyperWrite 时，将在 Data Facility Storage Management Subsystem (DFSMS) 级别并行发出对主卷和辅助卷的写操作，而不是在存储子系统级别按顺序发出，从而降低对性能的影响。

下图说明了用于活动日志的 zHyper 写入，以及用于其他 IBM MQ 数据集类型的 Metro Mirror 。 请注意，如果 zHyper 写操作失败，那么 DFSMS 将使用 Metro Mirror 透明地重新发出写操作。



zHyperWrite on IBM MQ，仅在活动日志数据集中受支持。

要将 zHyperWrite 与活动日志配合使用，您需要:

- 配置 IBM MQ 以使用 zHyperWrite 和
- 活动日志需要位于支持 zHyper 写的卷上

如果同时满足这两个条件，那么将针对 zHyperWrite 启用对活动日志的写入。

您可以使用下列其中一种方法将 IBM MQ 配置为使用 zHyperWrite:

- 在系统参数模块中指定 ZHYWRITE(YES)。
- 发出命令 SET LOG ZHYWRITE(YES)。

将活动日志数据集的以下条件设置为在支持 zHyper 写的卷上:

- 为 Metro Mirror 启用卷，并且卷支持 zHyper 写
- 确保卷已启用 HyperSwap
- 在 IECIOSxx 参数中指定 HYPERWRITE=YES

如果满足所有上述条件，那么将针对 zHyperWrite 启用对活动日志的写入。

如果不满足其中一个或多个条件，那么 IBM MQ 会正常写入活动日志，而 Metro Mirror 会复制写入 (如果已配置)。

注意:

- IBM MQ 不要求所有活动日志数据集都位于支持写的 zHyper 卷上。

  如果 IBM MQ 检测到某些活动日志数据集位于支持写的 zHyper 卷上，而其他数据集不在其中，那么它会发出消息 CSQJ166E 并继续处理。
- IBM MQ 检查活动日志数据集在首次打开数据集时是否支持 zHyper 写入。

日志数据集在队列管理器启动时打开，或者在使用 DEFINE LOG 命令动态添加时打开。 如果在队列管理器打开日志数据集时使日志数据集具有 zHyper 写功能，那么队列管理器直到重新启动后才会检测到此数据集。

您可以使用 DISPLAY LOG 命令的输出来指示当前活动日志数据集是否支持 zHyper 写。 以下示例显示这两个数据集都支持 zHyperWrite。 如果已使用 ZHYWRITE (YES) 配置队列管理器，那么将针对 zHyperWrite 启用对这些日志的写入:

```
Copy %Full zHyperWrite DSName
 1      4  CAPABLE     MQTST.SUBSYS.MQDL.LOGCOPY1.DS001
 2      4  CAPABLE     MQTST.SUBSYS.MQDL.LOGCOPY2.DS001
```

z/OS 规划日志归档存储器

使用本主题来了解维护归档日志数据集的不同方法。

您可以将归档日志数据集放置在标准标签磁带或 DASD 上，并且可以通过数据设施分层存储管理器 (DFHSM) 对其进行管理。 归档日志数据集中的每个 z/OS 逻辑记录都是来自活动日志数据集的 VSAM 控制时间间隔。 块大小是 4 KB 的倍数。

将动态分配归档日志数据集，名称由 IBM MQ 选择。 在系统参数模块中指定了此类分配所需的数据集名称前缀，块大小，单元名称和 DASD 大小。 您还可以在安装时选择让 IBM MQ 将日期和时间添加到归档日志数据集名称。

无法与 IBM MQ 一起指定新归档日志的特定卷，但您可以使用存储管理例程来管理此卷。 如果发生分配错误，那么会将卸载推迟到下次触发卸载时。

如果在安装时指定双归档日志，那么从活动日志检索的每个日志控制时间间隔将写入两个归档日志数据集。 包含在归档日志数据集对中的日志记录完全相同，但卷结束点对于多卷数据集不同步。

归档日志是否应位于磁带或 DASD 上?

在决定是将磁带还是 DASD 用于归档日志时，您应该考虑以下因素:

- 在决定磁带或磁盘之前，请查看操作过程。 例如，如果选择归档到磁带，那么当需要磁带机时，必须有足够的磁带机。 发生灾难后，所有子系统都可能需要磁带机，并且您可能没有期望的可用磁带机数量。
- 在恢复期间，一旦安装了磁带，磁带上的归档日志就可用。 如果已使用 DASD 归档，并且数据集已使用分层存储管理器 (HSM) 迁移到磁带，那么在 HSM 将每个数据集重新调用到磁盘时存在延迟。 您可以在使用归档日志之前重新调用数据集。 但是，并不总是能够预测需要它们的正确顺序。
- 在 DASD 上使用归档日志时，如果需要许多日志 (从备份恢复后恢复页集时可能是这种情况) ，那么可能需要大量 DASD 来保存所有归档日志。
- 在低使用率系统或测试系统中，在 DASD 上使用归档日志可能更方便，以消除对磁带安装的需求。
- 发出 RECOVER CFSTRUCT 命令和回退持久工作单元都会导致日志向后读取。 具有硬件压缩功能的磁带机在执行向后读取的操作时性能很差。 在 DASD 上规划足够的日志数据，以避免从磁带向后读取。

归档到 DASD 可提供更快的可恢复性，但比归档到磁带更昂贵。 如果使用双重日志记录，那么可以指定归档日志的主副本转至 DASD ，辅助副本转至磁带。 这将提高恢复速度，而无需使用任意数量的 DASD ，并且您可以将磁带用作备份。

有关如何将日志从磁带归档到 DASD 以及如何执行逆向过程的详细信息，请参阅 第 143 页的『更改归档日志的存储介质』 。

归档到磁带

如果选择归档到磁带设备，那么 IBM MQ 最多可扩展至 20 个卷。

如果您正在考虑更改活动日志数据集的大小以使该集适合一个磁带卷，请注意 BSDS 的副本与活动日志数据集的副本放在同一磁带卷上。 向下调整活动日志数据集的大小，以抵消磁带卷上 BSDS 所需的空间。

如果在磁带上使用双重归档日志，那么通常一个副本保存在本地，另一个副本保存在非现场以用于灾难恢复。

归档到 DASD 卷

IBM MQ 要求您对非磁带设备 (DASD) 上分配的所有归档日志数据集进行编目。 如果选择归档到 DASD ，那么 CSQ6ARVP 宏的 CATALOG 参数必须为 YES。 如果此参数为 NO ，并且您决定将归档日志数据集放在 DASD 上，那么每次分配归档日志数据集时，都会收到消息 CSQJ072E ，尽管 IBM MQ 仍对数据集进行编目。

如果归档日志数据集保留在 DASD 上，那么归档日志数据集可以扩展到另一个卷; 支持多卷。

如果选择使用 DASD ，请确保主空间分配 (数量和块大小) 足够大，以包含来自活动日志数据集的数据或来自相应 BSDS 的数据 (以两者中的较大者为准)。

这将最大程度地减少在卸载过程中出现不需要的 z/OS X'B37' 或 X'E37' 异常终止代码的可能性。 主空间分配是使用 CSQ6ARVP 宏的 PRIQTY (主数量) 参数设置的。

从 IBM MQ for z/OS 8.0 开始，归档日志数据集可以存在于大型或扩展格式的连续数据集上。 SMS ACS 例程现在可以使用 DSNTYPE (LARGE) 或 DSNTYPE (EXT)。 这些在 IBM MQ for z/OS 8.0 之前不受支持。

IBM MQ 支持将归档日志作为扩展格式数据集进行分配。 使用扩展格式时，最大归档日志大小将从 65535 个磁道增加到最大活动日志大小 4GB。 归档日志适合在扩展地址卷 (EAV) 的扩展寻址空间 (EAS) 中分配。

如果提供了必需的硬件和软件级别，那么使用 zEDC 将归档日志分配给使用 COMPACTION 定义的数据类可能会减少保存归档日志所需的磁盘存储空间。 有关更多信息，请参阅 IBM MQ for z/OS: 通过 IBM zEnterprise Data Compression (zEDC)减少存储占用率。

请参阅 使用 zEnterprise Data Compression (zEDC) 增强功能 ，以获取有关硬件和软件级别的详细信息以及 RACF 概要文件更改示例。

可以将 z/OS 数据集加密功能应用于在 IBM MQ 8.0 或更高版本上运行的队列管理器的归档日志。 必须通过 "自动类选择" (ACS) 例程将这些归档日志分配给使用 EXTENDED 属性定义的数据类以及确保数据已加密的数据集密钥标签。

将 SMS 与归档日志数据集配合使用

如果已安装 MVS/DFP 存储管理子系统 (DFSMS) ，那么可以为归档日志数据集编写 "自动类选择" (ACS) 用户出口过滤器，这可帮助您为 SMS 环境转换这些用户出口过滤器。

例如，这样的过滤器可以将输出路由到 DFSMS 可以管理的 DASD 数据集。 如果以此方式使用 ACS 过滤器，那么必须谨慎操作。 由于 SMS 要求对 DASD 数据集进行编目，因此必须确保 CSQ6ARVP 宏的 CATALOG DATA 字段包含 YES。 否则，将返回消息 CSQJ072E ; 但是，数据集仍由 IBM MQ 编目。

有关 ACS 过滤器的更多信息，请参阅 DFSMShsm 动态分配的数据集 。

z/OS 更改归档日志的存储介质
更改归档日志所使用的存储介质的过程。

关于此任务

此任务描述如何更改用于归档日志的存储介质，例如，从归档到磁带再到归档到 DASD。

您可以选择如何进行更改:

1. 仅使用 CSQ6ARVP 宏进行更改，以便从队列管理器下次重新启动时应用这些更改。
2. 使用 CSQ6ARVP 宏进行更改，并使用 SET ARCHIVE 命令动态进行更改。 这意味着更改将从队列管理器下次归档日志文件时开始应用，并在队列管理器重新启动后持久存储。

过程

1. 更改以将归档日志存储在 DASD 上而不是磁带上:
   a) 请阅读 第 143 页的『归档到 DASD 卷』 部分，并查看 CSQ6ARVP 参数。
   b) 在 CSQ6ARVP 中对以下参数进行更改
      - 更新 UNIT 以及 (如果需要) UNIT2 参数。
      - 更新 BLKSIZE 参数，因为 DASD 的最佳设置与磁带不同。
      - 将 PRIQTY 和 SECQTY 参数设置为足以容纳最大的活动日志或 BSDS。
      - 将 CATALOG 参数设置为 YES。
      - 确认 ALCUNIT 设置是您想要的。 您应该使用 BLK，因为它独立于设备类型。
      - 如果 ARCWTOR 参数尚未设置为 NO，请将其设置为 NO。
2. 更改以将归档日志存储在磁带而不是 DASD 上:
   a) 阅读 第 142 页的『归档到磁带』部分，并查看 CSQ6ARVP 参数。
   b) 在 CSQ6ARVP: 中对以下参数进行更改:
      - 更新 UNIT 以及 (如果需要) UNIT2 参数。
      - 更新 BLKSIZE 参数，因为磁带的最佳设置与 DASD 不同。
      - 确认 ALCUNIT 设置是您想要的。 您应该使用 BLK，因为它独立于设备类型。
      - 查看 ARCWTOR 参数的设置。

z/OS 我需要多长时间来保留归档日志

使用此部分中的信息可帮助您规划备份策略。

使用 USING CSQ6ARVP 中的 ARCRETN 参数或 SET SYSTEM 命令来指定归档日志的保留时间 (以天计)。 在此时间段之后， z/OS 可以删除数据集。

您可以在不再需要归档日志数据集时手动将其删除。

- 队列管理器可能需要归档日志以进行恢复。

  队列管理器只能在 BSDS 中保留最新的 1000 个归档，当归档日志不在 BSDS 中时，它们不能用于恢复，并且仅用于审计，分析或重放类型目的。
- 您可能希望保留归档日志，以便可以从日志中抽取信息。 例如，从日志中抽取消息，并查看放入或获取消息的用户标识。

BSDS 包含有关日志和其他恢复信息的信息。 此数据集是固定大小。 当归档日志数达到 CSQ6LOGP 中的 MAXARCH 值时，或者当 BSDS 填满时，将覆盖最旧的归档日志信息。

有一些实用程序用于从 BSDS 中除去归档日志条目，但通常， BSDS 会合并并覆盖最旧的归档日志记录。

何时需要归档日志

您需要定期备份页面集。 备份频率确定在丢失页集时需要哪些归档日志。

您需要定期备份 CF 结构。 备份频率确定在 CF 结构中丢失数据时需要哪些归档日志。

恢复可能需要归档日志。 以下信息说明何时可能需要归档日志，其中存在不同 IBM MQ 资源的问题。

丢失页集

必须从备份中恢复系统，然后重新启动队列管理器。

您需要从执行备份时开始的日志，以及在执行备份之前最多三个日志数据集。

所有 LPAR 都失去与 CF 结构的连接，或者该结构不可用

使用 RECOVER CFSTRUCT 命令来恢复结构。

结构恢复需要自上次备份以来访问结构的所有队列管理器的日志 (返回到执行备份的时间) 以及执行备份的队列管理器的日志中的结构备份本身。

如果您一直在执行 CF 结构的频繁备份，那么数据应该位于活动日志中，并且您不应该需要归档日志。

如果没有 CF 结构的最新备份，那么您可能需要归档日志。

注: 将丢失所有非持久消息; 将通过执行以下任务来重新创建所有持久消息:

1. 从日志中读取最后一个 CF 结构备份
2. 从已使用该结构的所有队列管理器中读取日志
3. 自备份以来合并更新

管理结构重建

如果需要重建管理结构，那么将从 QSG 中每个队列管理器的日志的最后一个检查点读取信息。

如果队列管理器未处于活动状态，那么 QSG 中的另一个队列管理器将读取日志。

您不应该需要归档日志。

丢失 SMDS 数据集

如果丢失 SMDS 数据集，或者该数据集已损坏，那么该数据集将变为不可用，并且其状态将设置为 FAILED。 CF 结构保持不变。

为了复原 SMDS 数据集，您需要:

1. 重新定义 SMDS 数据集，以及
2. 通过发出 RECOVER CFSTRUCT 命令来恢复 CF 结构。

   注: CF 结构上的所有非持久消息都将丢失; 将复原所有持久消息。

队列管理器日志的需求与从不可用的结构进行恢复的需求相同。

z/OS 计划增大最大可寻址日志范围

您可以通过将队列管理器配置为使用较大的日志相对字节地址 (RBA) 来增大最大可寻址日志范围。

日志 RBA 大小已从 IBM MQ for z/OS 8.0 增加。 有关此更改的概述，请参阅 较大日志相对字节地址。

V 9.2.0 如果队列管理器不在队列共享组中，那么可以随时将其转换为使用 8 字节日志 RBA 值。 如果您随后迁移回 IBM MQ for z/OS 9.0.0，请确保使用的是 **OPMODE**= (*NEWFUNC,900*)，否则队列管理器将不会启动。

V 9.2.5 如果在 IBM MQ 9.2.5 或更高版本上创建了队列管理器，那么缺省情况下已启用 8 字节日志 RBA，因此不需要转换。

在将队列共享组中的任何队列管理器转换为使用 8 字节日志 RBA 之前，队列共享组中的所有队列管理器必须处于下列其中一个级别:

- IBM MQ for z/OS 9.0.n CD， IBM MQ for z/OS 9.1.0 LTS 或更高版本
- 位于 IBM MQ for z/OS 9.0.0，并且已使用 **OPMODE**= (*NEWFUNC,800*) 或 **OPMODE**= (*NEWFUNC,900*) 启动

然后，可以依次更改每个队列管理器以使用 8 字节日志 RBA 值。 同时更改所有队列管理器并不是必需的。

当队列共享组中的队列管理器已转换为使用 8 字节日志 RBA 值时，队列共享组中的其他队列管理器可以使用转换后的队列管理器的日志，即使尚未转换为使用 8 字节日志 RBA 值也是如此。 例如，这对于对等恢复很有用。

正在撤销更改

无法回退更改。

需要多长时间?

更改需要重新启动队列管理器。 停止队列管理器，对引导数据集 (BSDS) 或数据集运行 CSQJUCNV 实用程序，以创建新的数据集，重命名这些引导数据集，然后重新启动队列管理器。 CSQJUCNV 实用程序通常需要几秒钟才能运行。

这有什么影响?

- 在使用 8 字节日志 RBA 的情况下，每次将数据写入日志数据集都有额外的字节。 因此，对于由持久消息组成的工作负载，写入日志的数据量会略有增加。
- 写入页集或耦合设施 (CF) 结构的数据不受影响。

相关任务

实现较大的日志相对字节地址

z/OS 规划通道启动程序

通道启动程序提供队列管理器之间的通信，并在其自己的地址空间中运行。

有两种类型的连接:

1. 应用程序通过网络连接到队列管理器。 这些称为客户机通道。
2. 队列管理器到队列管理器连接。 这些称为 MCA 通道。

侦听器

通道侦听器程序侦听入局网络请求，并在需要该通道时启动相应的通道。 要处理入站连接，通道启动程序至少需要配置一个 IBM MQ 侦听器任务。 侦听器可以是 TCP 侦听器，也可以是 LU 6.2 侦听器。

每个侦听器都需要一个 TCP 端口或 LU 名。 IBM MQ for Multiplatforms 通常使用 TCP/IP 端口 1414 (缺省值)。

请注意，对于每个通道启动程序，可以有多个侦听器。

TCP/IP

通道启动程序可以使用同一 z/OS 映像上的多个 TCP 堆栈进行操作。 例如， 一个 TCP 堆栈可以用于内部连接，另一个 TCP 堆栈可以用于外部连接。

定义输出通道时:

1. 设置连接的目标主机和端口。 这可以是:
   - IP 地址，例如 `10.20.4.6`
   - 主机名，例如 `mvs-prod.myorg.com`

   如果使用主机名来指定目标，那么 IBM MQ 将使用域名系统 (DNS) 来解析目标的 IP 地址。
2. 如果您正在使用多个 TCP 堆栈，那么可以在通道定义上指定 **LOCLADDR** 参数，该参数指定要使用的 IP 堆栈地址。

您应该计划具有高可用性 DNS 服务器。 如果 DNS 不可用，那么出站通道可能无法启动，并且无法处理使用主机名映射入局连接的通道认证规则。

APPC 和 LU 6.2

如果您正在使用 APPC , 那么通道启动程序需要 LU 名以及 APPC 中的配置。

队列共享组

要提供单个系统映像，并允许入局 IBM MQ 连接请求转至队列共享组中的任何队列管理器，您需要执行一些配置。 例如:

1. 硬件网络路由器。 此路由器具有企业看到的一个 IP 地址，并且可以将初始请求路由到连接到此硬件的任何队列管理器。
2. 虚拟 IP 地址 (VIPA)。 指定了企业范围的 IP 地址，该地址可以路由到综合系统中的任何一个 TCP 堆栈。 然后， TCP 堆栈可以将其路由到综合系统中的任何侦听队列管理器。

保护 IBM MQ 流量

您可以将 IBM MQ 配置为使用 TLS (或 SSL) 连接来保护连线上的数据。 要使用 TLS，您需要使用数字证书和密钥环。

您还需要与通道远程端的人员一起工作，以确保您具有兼容的 IBM MQ 定义和兼容的证书。

您可以根据以下内容来控制哪些连接可以连接到 IBM MQ 和用户标识:

- IP 地址
- 客户机用户标识
- 远程队列管理器，或
- 数字证书 (请参阅 通道认证记录)

还可以通过确保客户机应用程序提供有效的用户标识和密码来限制客户机应用程序 (请参阅 连接认证)。

您可以使通道启动程序工作，然后将每个通道配置为使用 TLS (一次一个)。

监视通道启动程序

有 MQSC 命令提供有关通道启动程序和通道的信息:

- DISPLAY CHINIT 命令提供有关通道启动程序和活动侦听器的信息。
- DISPLAY CHSTATUS 命令显示通道的活动和状态。

通道启动程序还可以生成 SMF 记录以及有关通道启动程序任务和通道活动的信息。 请参阅 第 148 页的『规划通道启动程序 SMF 数据』 以获取更多信息。

通道启动程序在通道启动和停止时向作业记录发出消息。 企业中的自动化可以使用这些消息来捕获状态。 由于某些通道仅在几秒钟内处于活动状态，因此可能会生成许多消息。 您可以通过使用 z/OS 消息处理设施或通过使用 SET SYSTEM 命令设置 **EXCLMSG** 来禁止这些消息。

配置 IBM MQ 通道定义

当您将许多队列管理器连接在一起时，可能很难管理所有对象定义。 使用 IBM MQ 集群可以简化此过程。

指定两个队列管理器作为完整存储库。 其他队列管理器需要一个与其中一个存储库的连接和一个与其中一个存储库的连接。 当需要连接到其他队列管理器时，队列管理器会自动创建并启动通道。

如果计划在集群中具有大量队列管理器，那么应该计划使队列管理器充当专用存储库，并且没有应用程序流量。

请参阅 第 17 页的『规划分布式队列和集群』 以获取更多信息。

配置通道启动程序之前的操作

1. 决定您是使用 TCP/IP 还是 APPC。
2. 如果您正在使用 TCP，请为 IBM MQ 分配至少一个端口。
3. 如果需要 DNS 服务器，请将该服务器配置为高可用性 (如果需要)。
4. 如果您正在使用 APPC，请分配 LU 名并配置 APPC。

在配置通道启动程序之后，进入生产环境之前的操作

1. 规划您将拥有的连接:
   a. 来自远程应用程序的客户机连接。
   b. 与其他队列管理器之间的 MCA 通道。 通常，您具有与每个远程队列管理器之间的通道。
2. 设置集群，或加入现有集群环境。
3. 请考虑是否需要在通道启动程序前使用多个 TCP 堆栈， VIPA 或外部路由器以实现可用性。
4. 如果计划使用 TLS:

    a. 设置密钥环
    b. 设置证书
5. 如果计划使用通道认证:
    a. 决定将入站会话映射到 MCA 用户标识的条件
    b. 通过设置队列管理器参数 **REVDNS** 来启用逆向 DNS 查找
    c. 查看安全性。 例如，删除缺省通道，并在通道的 **MCAUSER** 属性中指定仅具有必需权限的用户标识。
6. 捕获通道启动程序生成的记帐和统计信息 SMF 记录并对其进行后处理。
7. 自动监视作业日志消息。
8. 如有必要，请调整网络环境以提高吞吐量。 通过 TCP，大型发送和接收缓冲区可提高吞吐量。 您可以使用以下命令强制 MQ 使用特定的 TCP 缓冲区大小:

    ```
    RECOVER QMGR(TUNE CHINTCPRBDYNSZ nnnnn)
    RECOVER QMGR(TUNE CHINTCPSBDYNSZ nnnnn)
    ```

    将通道的 SO_RCVBUF 和 SO_SNDBUF 设置为 nnnnn 中指定的大小 (以字节为单位)。

相关概念

第 124 页的『规划队列管理器』
在设置队列管理器时，您的规划应允许队列管理器增长，以便队列管理器满足企业的需求。

z/OS 规划通道启动程序 SMF 数据

您需要规划为通道启动程序收集 SMF 数据的实现。

通道启动程序生成两种类型的记录:

- 包含有关通道启动程序及其内的任务的信息的统计信息数据。
- 具有类似于 DISPLAY CHSTATUS 命令的信息的通道记帐数据。

使用以下命令开始收集统计信息数据:

```
START TRACE(STAT) CLASS(4)
```

并使用以下命令将其停止:

```
STOP TRACE(STAT) CLASS(4)
```

使用以下命令开始收集记帐数据:

```
START TRACE(ACCTG) CLASS(4)
```

并使用以下命令将其停止:

```
STOP TRACE(ACCTG) CLASS(4)
```

您可以控制为使用通道定义或队列管理器上的 **STATCHL** 属性收集了哪些通道的记帐数据。

- 对于客户机通道，必须在队列管理器级别设置 **STATCHL** 。
- 对于自动定义的集群发送方通道，您可以使用 **STATACLS** 队列管理器属性来控制记帐数据的收集。

队列管理器的 **STATCHL** 的缺省值为 OFF。 为了收集通道记帐数据，除了启动类 4 记帐跟踪外，还必须将 **STATCHL** 的值从队列管理器或通道定义上的缺省值更改为其他值。

在以下情况下生成 SMF 记录:

- LTS 已经过由 CSQ6SYSP **STATIME** 参数指示的 从 IBM MQ for z/OS 9.2.0 到 9.2.3， 时间间隔，或者在 SMF 数据收集广播上 **STATIME** 为零。 收集通道启动程序和队列管理器的 SMF 数据的请求已同步。
- V 9.2.4 从 IBM MQ for z/OS 9.2.4 开始，已经过 CSQ6SYSP **STATIME** 或 **ACCTIME** 参数指示的时间间隔; 或者，如果 SMF 数据收集广播上的 **STATIME** 或 **ACCTIME** 为零，那么 收集通道启动程序和队列管理器的 SMF 数据的请求已同步。
- 发出了 `STOP TRACE(ACCTG) CLASS(4)` 或 `STOP TRACE(STAT) CLASS(4)` 命令，或者
- 关闭通道启动程序。 此时，将写出任何 SMF 数据。

如果通道在 SMF 时间间隔内停止，那么下次运行 SMF 处理时，会将记帐数据写入 SMF。 如果客户机连接，执行某些工作和断开连接，然后重新连接和断开连接，那么将生成两组通道记帐数据。

统计信息数据通常适用于一条 SMF 记录，但是，如果正在使用大量任务，那么可能会创建多条 SMF 记录。

将针对为其启用记帐数据的每个通道收集记帐数据，并且通常会将这些数据放入一个 SMF 记录中。 但是，如果有大量通道处于活动状态，那么可能会创建多条 SMF 记录。

收集通道启动程序 SMF 数据的成本很小。 通常， CPU 使用率的增长低于几% ，并且通常在测量误差范围内。

在使用此功能之前，您需要与 z/OS 系统程序员一起工作，以确保 SMF 具有用于其他记录的容量，并确保它们更改其用于抽取 SMF 记录以包含新的 SMF 数据的过程。

对于通道启动程序统计数据， SMF 记录类型为 115 和子类型 231。

对于通道启动程序记帐数据， SMF 记录类型为 116 ，子类型为 10。

您可以编写自己的程序来处理此数据，或者使用包含程序 MQSMF 的 SupportPac MP1B 来打印数据，并创建适合导入到传播表中的逗号分隔值 (CSV) 格式的数据。

如果在捕获通道启动程序 SMF 数据方面迂到问题，请参阅 捕获通道启动程序 (CHINIT) 的 SMF 数据时处理问题 以获取更多信息。

相关任务

解释 IBM MQ 性能统计信息

通道记帐数据故障诊断

z/OS 规划 z/OS TCP/IP 环境

要通过网络获取最佳吞吐量，必须使用大小为 64 KB 或更大的 TCP/IP 发送和接收缓冲区。 通过此大小，系统将优化其缓冲区大小。

请参阅 Dynamic Right Sizing for High Latency Networks 是什么? for more information.

可以使用以下 Netstat 命令来检查系统缓冲区大小，例如:

```
TSO NETSTAT ALL (CLIENT csq1CHIN
```

结果显示大量信息，包括以下两个值:

```
ReceiveBufferSize: 0000065536
SendBufferSize: 0000065536
```

65536 是 64 KB。 如果缓冲区大小小于 65536 ，那么必须与网络团队一起在 TCPIP 过程中增加 PROFILE DDName 中的 **TCPSENDBFRSIZE** 和 **TCPRCVBUFRSIZE** 值。 例如，可以使用以下命令:

```
TCPCONFIG TCPSENDBFRSZE 65536 TCPRCVBUFRSIZE 65536
```

如果无法更改系统范围的 **TCPSENDBFRSIZE** 或 **TCPRCVBUFRSIZE** 设置，请与 IBM 软件支持中心联系。

z/OS 规划队列共享组 (QSG)

实现共享排队环境的最简单方法是配置队列管理器，将该队列管理器添加到 QSG，然后将其他队列管理器添加到 QSG。

队列共享组使用 Db2 表来存储配置信息。 共享同一 Db2 数据共享组的所有 QSG 都使用了一组表。

共享队列消息存储在耦合设施 (CF) 中的结构中。 每个 QSG 都有自己的 CF 结构集。 您需要配置结构以满足您的需求。

消息也可以存储在共享消息数据集 (SMDS) 中。 大小超过 63KB 的消息不能存储在 CF 中。 您需要使用 SMDS 将这些消息存储在共享队列上。

消息概要文件和容量规划

您应该了解共享队列消息的消息概要文件。 以下是需要考虑的因素示例:

- 平均和最大消息大小
- 典型队列深度和异常队列深度。 例如，您可能需要有足够的容量来保存一整天的消息，而典型的队列深度在 100 条消息以下。

如果消息概要文件发生更改，那么您可以稍后增大结构的大小或实现 SMDS。

如果您希望能够处理大量峰值消息，那么可以配置 IBM MQ 以在结构的使用达到用户指定的阈值时将消息卸载到 SMDS。

您需要决定是否要对 CF 结构进行双工。 这由 CFRM 策略中的 CF 结构定义控制:

1. 一个双工结构使用两个耦合设施。 如果一个 CF 存在问题，那么不会中断服务，并且可以在第三个 CF 上重建结构 (如果一个 CF 可用)。 双工结构会显着影响共享队列上的操作性能。
2. 如果结构不是双工的，那么 CF 的问题意味着该 CF 中的结构上的共享队列将变为不可用，直到可以在另一个 CF 中重建该结构为止。

   在此情况下，可以将 IBM MQ 配置为在另一个 CF 中自动重建结构。 将从队列管理器的日志中恢复持久消息。

请注意，更改 CF 定义很容易。

您可以定义结构，使其仅可容纳非持久消息，或使其可容纳持久和非持久消息。

需要定期备份可保存持久消息的结构。 至少每小时备份一次 CF 结构，以最小化在发生故障时恢复结构所需的时间。 备份存储在执行备份的队列管理器的日志数据集中。

如果您期望共享队列上具有高吞吐量的消息，那么最好使用专用队列管理器来备份 CF 结构。 这将减少恢复结构所需的时间，因为需要从队列管理器日志中读取较少的数据。

通道

要为连接到 IBM MQ QSG 的应用程序提供单个系统映像，您可以定义共享输入通道。 如果设置了这些设置，那么进入队列共享组环境的连接可以转至 QSG 中的任何队列管理器。

您可能需要为这些通道设置网络路由器或虚拟 IP 地址 (VIPA)。

您可以定义共享输出通道。 可以从 QSG 中的任何队列管理器启动共享输出通道实例。

请参阅 共享通道 以获取更多信息。

安全性

您可以使用外部安全管理器来保护 IBM MQ 资源。 如果您正在使用 RACF，那么 RACF 概要文件将以队列管理器名称作为前缀。 例如，将使用名为 `qmgrName.APPLICATION.INPUT.` 的 MQQUEUE 类中的概要文件来保护名为 APPLICATION.INPUT 的队列

使用队列共享组时，可以继续使用以队列管理器名称为前缀的概要文件来保护资源，也可以使用队列共享组名称作为概要文件的前缀。 例如，`qsgName.APPLICATION.INPUT`。

您应该将概要文件前缀与队列共享组名配合使用，因为这意味着所有队列管理器都有一个定义，可以保存您的工作，并防止队列管理器之间的定义不匹配。

相关概念

第 124 页的『规划队列管理器』
在设置队列管理器时，您的规划应允许队列管理器增长，以便队列管理器满足企业的需求。

z/OS 规划耦合设施和卸载存储环境

在规划耦合设施 (CF) 结构以及共享消息数据集 (SMDS) 环境或 Db2 环境的初始大小和格式时，请使用本主题。

本节包含有关以下主题的信息：

定义耦合设施资源

如果您打算使用共享队列，那么必须定义 IBM MQ 将在 CFRM 策略中使用的耦合设施结构。 为此，必须首先使用有关结构的信息更新 CFRM 策略，然后激活该策略。

您的安装可能具有描述可用耦合设施的现有 CFRM 策略。 管理数据实用程序 用于根据您提供的文本语句修改策略内容。 必须向策略添加语句以定义新结构的名称，在其中定义它们的耦合设施以及这些结构的大小。

CFRM 策略还确定是否对 IBM MQ 结构进行双工以及如何在故障场景中重新分配这些结构。 共享队列恢复 包含针对影响耦合设施的故障的弹性配置 CFRM 的建议。

决定卸载存储环境

可以从耦合设施中卸载共享队列的消息数据，并将其存储在 Db2 表或称为 共享消息数据集 (SMDS) 的 IBM MQ 受管数据集中。 必须始终卸载太大而无法存储在耦合设施中的消息 (即，大于 63 KB 的消息) ，并且可以选择卸载较小的消息以减少耦合设施空间使用情况。

有关更多信息，请参阅 指定共享消息的卸载选项。

规划结构

队列共享组 (QSG) 至少需要定义两个结构。 第一个结构 (称为管理结构) 用于协调队列共享组中的 IBM MQ 内部活动。 此结构中不包含任何用户数据。 它具有固定名称 *qsg-name*CSQ_ADMIN (其中 *qsg-name* 是队列共享组的名称)。 后续结构称为应用程序结构，用于保存 IBM MQ 共享队列上的消息。 每个结构最多可容纳 512 个共享队列。

名为 *qsg-name*CSQSYSAPPL 的应用程序结构用于系统队列。 定义此结构是可选的，但需要此结构才能使用某些功能部件。 缺省情况下， SYSTEM.QSG.CHANNEL.SYNCQ 和 SYSTEM.QSG.UR.RESOLUTION.QUEUE 队列是在 *qsg-name*CSQSYAPPL 结构上定义的。

使用多个结构

队列共享组最多可以连接到 64 个耦合设施结构。 其中一个结构必须是管理结构。 如果定义了此结构，那么其中另一个结构可能是 *qsg-name*CSQSYSAPPL 结构。 最多可以将 63 (如果定义了 *qsg-name*CSQSYSAPPL) 结构用于消息数据。 出于以下任何原因，您可以选择使用多个应用程序结构:

- 您有一些队列可能包含大量消息，因此需要整个耦合设施的所有资源。
- 您需要大量共享队列，因此必须在多个结构之间进行拆分，因为每个结构只能包含 512 个队列。
- RMF 报告了结构的使用特征，建议您将其包含的队列分布在多个耦合设施中。
- 出于数据隔离原因，您希望某些队列数据保存在物理上与其他队列数据不同的耦合设施中。
- 使用结构级别属性和命令 (例如 BACKUP CFSTRUCT) 来执行持久共享消息的恢复。 为了简化备份和恢复，可以将保存非持久消息的队列分配给与保存持久消息的那些结构不同的结构。

选择要在其中分配结构的耦合设施时，请考虑以下几点:

- 您的数据隔离需求。
- 耦合设施的波动性 (即，其通过断电来保存数据的能力)。
- 访问系统与耦合设施之间或耦合设施之间的故障独立性。
- 安装在耦合设施上的耦合设施控制代码 (CFCC) 的级别 (IBM MQ 需要级别 9 或更高级别)。

规划结构大小

行政结构

管理结构 (*qsg-name*CSQ_ADMIN) 必须足以包含队列共享组中每个队列管理器的 1000 个列表条目。 当队列管理器启动时，将检查该结构，以确定它是否足以满足当前 已定义 到队列共享组的队列管理器数。 如果队列管理器已由 CSQ5PQSG 实用程序添加，那么它们将被视为定义到队列共享组。 您可以使用 MQSC DISPLAY GROUP 命令检查哪些队列管理器已定义到组。

注: 计算结构的大小时，除了队列共享组中的队列管理器数量外，还应允许大型工作单元的大小。

第 152 页的表 22 显示队列共享组中定义的各种数量的队列管理器的管理结构所需的最小大小。 这些尺寸是针对 CFCC 级别 14 耦合设施结构建立的; 对于更高级别的 CFCC，它们可能需要更大。

表 *22:* 最小管理结构大小

队列共享组中定义的队列管理器数	需要的存储
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

表 *22:* 最小管理结构大小 *(*继续*)*	
队列共享组中定义的队列管理器数	**需要的存储**
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
24	25856 KB
25	27136 KB
26	27904 KB
27	28672 KB
28	29696 KB
29 日	30464 KB
30	31232 KB
31 日	32256 KB

将队列管理器添加到现有队列共享组时，存储需求可能已增加到超出 第 152 页的表 22 中建议的大小。如果是这样，请使用以下过程来估算 *qsg-name*CSQ_ADMIN 结构所需的存储空间:

1. 对队列共享组的现有成员发出 MQSC 命令 **DISPLAY CFSTATUS(CSQ_ADMIN)** 。
2. 抽取 CSQ_ADMIN 结构的 ENTSMAX 信息。
3. 如果此数目小于您要在队列共享组中定义的队列管理器总数的 1000 倍，请增大结构大小。

应用程序结构

保存 IBM MQ 消息所需的应用程序结构的大小取决于要同时保存在结构上的消息的可能数目和大小。

第 154 页的图 42 中的图形显示了您应该使 CF 结构具有多大的大小来保存共享队列上的消息。 要计算分配大小，您需要以下信息:

- 队列上消息的平均大小。
- 可能存储在结构中的消息总数。

查找沿水平轴的消息数。 选择与消息大小对应的曲线，并从垂直轴确定所需值。 例如，对于长度为 1 KB 的 20 万条消息，提供范围在 256 到 512 MB 之间的值。

第 154 页的表 23 以表格形式提供了相同的信息。

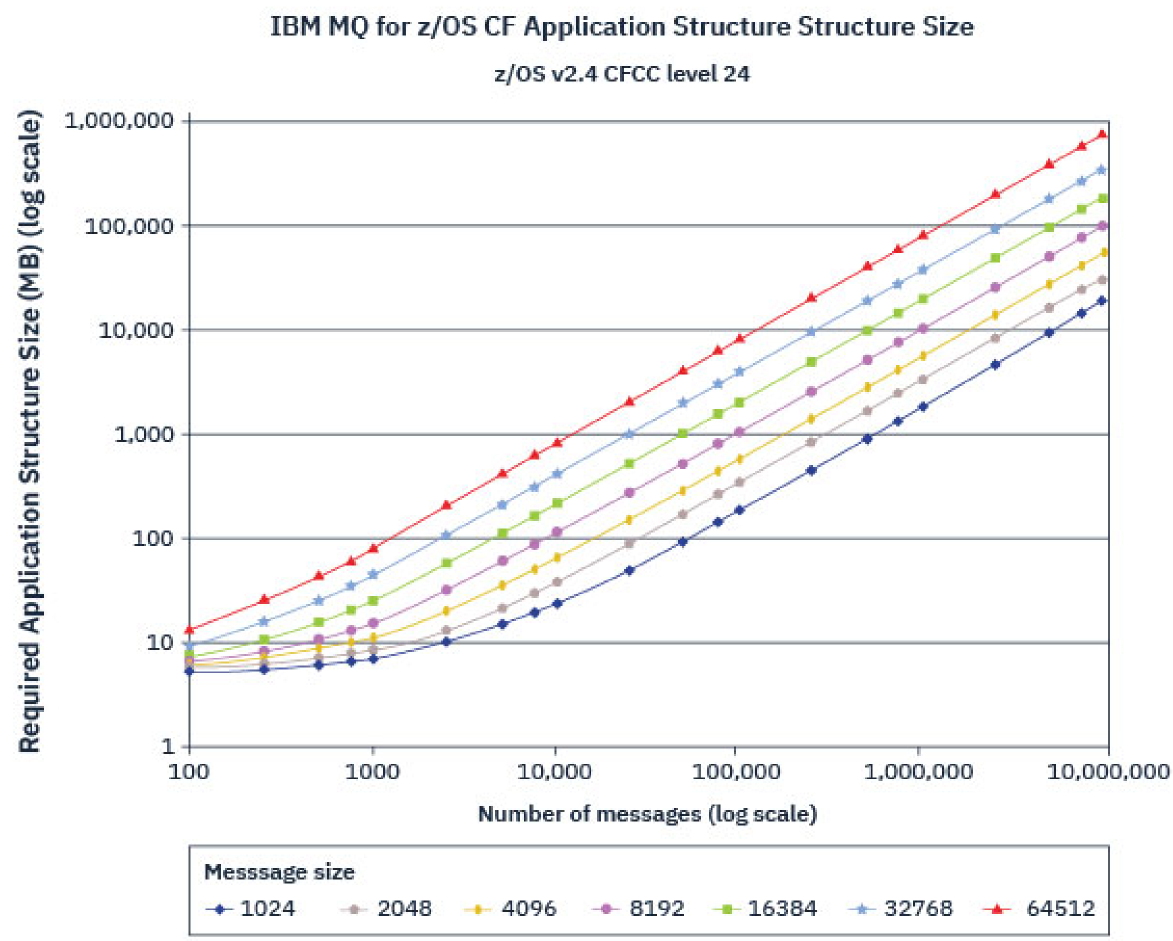


图 *42:* 计算耦合设施结构的大小

使用此表来帮助计算使耦合设施结构具有多大的大小:

表 *23:* 计算耦合设施结构的大小

消息数	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

CFRM 策略应包含以下语句:

- INITSIZE 是第一个队列管理器连接到该结构时所分配的大小 (以 KB 为单位)。
- SIZE 是结构可以达到的最大大小。
- FULLTHRESHOLD 设置 z/OS 发出消息 IXC585E 以指示结构变满的阈值的百分比值。

最佳实践是确保 INITSIZE 和 SIZE 在因子 2 内。 例如，通过先前确定的数字，您可以包含以下语句:

```
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)
```

如果结构使用达到发出警告消息的阈值，那么需要干预。 您可以使用 IBM MQ 来禁止对结构中的某些队列执行 MQPUT 操作，以防止应用程序写入更多消息，启动更多应用程序以从队列获取消息，或者停顿将消息放入队列的某些应用程序。

或者，您可以使用 z/OS 工具来改变现有结构大小。 以下 z/OS 命令:

```
SETXCF START,ALTER,STRNAME=structure-name,SIZE=newsize
```

将结构大小更改为 *newsize*，其中 *newsize* 是小于在结构的 CFRM 策略上指定的 SIZE 值但大于当前耦合设施大小的值。

您可以使用 MQSC DISPLAY CFSTATUS 命令来监视耦合设施结构的使用。

如果未执行任何操作，并且队列结构已满，那么将向应用程序返回 MQRC_STORAGE_MEDIUM_FULL 返回码。 如果管理结构变满，那么确切的症状取决于哪些进程迂到错误，但它们可能包括以下问题:

- 没有对命令的响应。
- 由于落实处理期间发生问题，导致队列管理器失败。

CSQSYSAPPL 结构

*qsg-name*CSQSYSAPPL 结构是系统队列的应用程序结构。 表 3 演示了如何估算在 *qsg-name*CSQSYSAPPL 结构上定义的缺省队列的消息数据大小的示例。

表 *24:* 显示针对大小调整的 *CSQSYSAPPL* 使用情况的表。

***qsg-name*CSQSYSAPPL 用法**	**确定规模**
SYSTEM.QSG.CHANNEL.SYNCQ	共享通道的每个活动实例 500 字节的 2 消息
SYSTEM.QSG.UR.RESOLUTION.QUEUE	1000 条 2 KB 的消息

建议的初始结构定义值如下所示:

```
STRUCTURE NAME(qsg-nameCSQSYSAPPL)
INITSIZE(20480)        /* 20 MB */
SIZE(30720)            /* 30 MB */
FULLTHRESHOLD(85)
```

可以根据共享通道和组恢复单元的使用情况来调整这些值。

将共享队列映射到结构

要将应用程序结构定义到 IBM MQ，请使用 DEFINE CFSTRUCT 命令。 向 IBM MQ 定义结构时，请勿在结构名称中包含 QSG 名称前缀。 例如，要向 IBM MQ 定义在 CFRM 策略中具有名称 *qsg-name*APPLICATION1 的应用程序结构，请发出以下命令:

```
DEFINE CFSTRUCT(APPLICATION1)
```

队列定义的 CFSTRUCT 属性用于将队列映射到结构。 在此属性中指定不带 QSG 名称前缀的 CF 结构的名称。 例如，以下命令在 APPLICATION1 结构上定义共享队列:

```
DEFINE QLOCAL(myqueue) QSGDISP(SHARED) CFSTRUCT(APPLICATION1)
```

z/OS 规划共享消息数据集 *(SMDS)* 环境

如果要将队列共享组与 SMDS 卸载配合使用，那么 IBM MQ 需要连接到一组共享消息数据集。 使用本主题来帮助了解存储 IBM MQ 消息数据所需的数据集需求和配置。

共享消息数据集 (由关键字 SMDS 描述) 是队列管理器用于存储耦合设施结构中存储的共享消息的卸载消息数据的数据集。

注: 为结构定义 SMDS 数据集时，每个队列管理器必须有一个数据集。

启用此形式的数据卸载时， **CFSTRUCT** 需要一组相关联的共享消息数据集，队列共享组中的每个队列管理器都有一个数据集。 共享消息数据集组是使用 **CFSTRUCT** 定义上的 **DSGROUP** 参数向 IBM MQ 定义的。 其他参数可用于提供更多可选信息，例如要使用的缓冲区数以及数据集的扩展属性。

每个队列管理器都可以写入它拥有的数据集，以存储通过该队列管理器写入的消息的共享消息数据，并且可以读取组中的所有数据集。

描述与结构关联的每个数据集的状态和属性的列表作为 **CFSTRUCT** 定义的一部分在内部进行维护，因此每个队列管理器都可以检查定义以找出当前可用的数据集。

可以使用 **DISPLAY CFSTATUS TYPE(SMDS)** 命令显示当前状态和可用性，并使用 **DISPLAY SMDS** 命令显示与指定 **CFSTRUCT** 关联的数据集的参数设置来显示此数据集信息。

各个共享消息数据集由拥有队列管理器名称 (通常使用 **SMDS** 关键字指定) 和 **CFSTRUCT** 结构名称的组合有效标识。

本部分描述了以下主题:

- DSGROUP 参数
- DSBLOCK 参数
- 共享消息数据集特征
- 共享消息数据集空间管理
- 访问共享消息数据集
- 创建共享消息数据集
- 共享消息数据集性能和容量注意事项
- 激活共享消息数据集

请参阅 DEFINE CFSTRUCT 以获取这些参数的详细信息。

有关管理共享消息数据集的信息，请参阅 管理共享消息数据集 以获取更多详细信息。

DSGROUP 参数

CFSTRUCT 定义上的 **DSGROUP** 参数标识要在其中存储该结构的大型消息的数据集组。 可以使用其他参数来指定要用于空间分配目的的逻辑块大小以及缓冲池大小和自动数据集扩展选项的值。

必须先设置 **DSGROUP** 参数，然后才能启用对数据集的卸载。

- 如果正在 **CFLEVEL(5)** 处定义新的 **CFSTRUCT** ，并且指定或假定了选项 **OFFLOAD(SMDS)** ，那么必须在同一命令上指定 **DSGROUP** 参数。
- 如果要变更现有 **CFSTRUCT** 以将 **CFLEVEL** 增大到 **CFLEVEL(5)** ，并且指定或假定了选项 **OFFLOAD(SMDS)** ，那么必须在同一命令上指定 **DSGROUP** 参数 (如果尚未设置)。

DBLOCK 参数

将每个数据集内的空间作为使用 **CFSTRUCT** 定义上的 **DSBLOCK** 参数指定的固定大小 (通常为 256 KB) 的逻辑块分配给队列，然后作为每个逻辑块内 4 KB 的页面范围 (对应于物理块大小和控制区间大小) 分配给各个消息。 逻辑块大小还确定可以在单个 I/O 操作中读取或写入的最大消息数据量，这与 SMDS 缓冲池的缓冲区大小相同。

DSBLOCK 参数的较大值可以通过减少单独 I/O 操作数来提高非常大的消息的性能。 但是，较小的值会减少每个活动请求所需的缓冲区存储量。 **DSBLOCK** 参数的缺省值为 256 KB ，这将在这些需求之间提供合理的平衡，因此通常不需要指定此参数。

共享消息数据集特征

共享消息数据集定义为 VSAM 线性数据集 (LDS)。 每个卸载的消息都存储在数据集中的一个或多个块中。 存储的数据由耦合设施条目中的信息直接寻址，例如扩展形式的虚拟存储器。 没有单独的索引或类似的控制信息存储在数据集本身中。

直接寻址方案意味着对于适合一个块的消息，只需要单个 I/O 操作即可读取或写入该块。 当消息跨越多个块时，可以完全重叠每个块的 I/O 操作以最小化耗用时间，前提是有足够的可用缓冲区。

共享消息数据集还包含少量常规控制信息，包括第一页中的头，该头包括恢复和重新启动状态信息，以及用于在队列管理器正常终止时保存可用块空间映射的空间映射检查点区域。

共享消息数据集空间管理

作为容量，性能和操作注意事项的背景信息，了解队列管理器如何管理共享消息数据集中的空间的概念可能很有用。

每个共享消息数据集中的可用空间由其拥有的队列管理器使用空间映射进行跟踪，该映射指示每个逻辑块中正在使用的页数。 当数据集处于正常关闭状态时，将在主存储器中维护空间映射，并将其保存在数据集中。 (在恢复情况下，将通过扫描耦合设施结构中的消息来自动重建空间映射，以找出当前正在使用的数据集页面)。

当写入具有已卸载消息数据的共享消息时，队列管理器将为每个消息块分配一系列页面。 如果存在指定队列的部分使用的当前逻辑块，那么队列管理器将从该块中的下一个可用页开始分配空间，否则将分配新的逻辑块。 如果整个消息不适合当前逻辑块，那么队列管理器将在逻辑块末尾分割消息数据，并为下一个消息块分配新的逻辑块。 重复此操作，直到为整个消息分配了空间为止。 最后一个逻辑块中任何未使用的空间都将另存为队列的新当前逻辑块。 当数据集正常关闭时，当前逻辑块中的任何未使用的页面都会在保存之前返回到空间映射。

当已读取具有已卸载消息数据的共享消息并准备好将其删除时，队列管理器通过将消息的耦合设施条目传输到所属队列管理器 (可能是同一队列管理器) 监视的清除列表来处理删除请求。 当条目到达此列表时，拥有队列管理器将读取和删除条目，并将释放的页面范围返回到空间映射。 当逻辑块中所有已使用的页面都已释放时，该块可供复用。

访问共享消息数据集

每个共享消息数据集都必须位于可供队列共享组中的所有队列管理器访问的共享直接访问存储器上。

在正常运行期间，每个队列管理器打开自己的共享消息数据集以进行读/写访问，并打开其他队列管理器的任何活动共享消息数据集以进行只读访问，因此可以读取这些队列管理器存储的消息。 这意味着每个队列管理器用户标识至少需要对其自己的共享消息数据集的 UPDATE 访问权以及对结构的所有其他共享消息数据集的 READ 访问权。

如果需要使用 **RECOVER CFSTRUCT** 恢复共享消息数据集，那么可以从队列共享组中的任何队列管理器执行恢复过程。 可用于执行恢复处理的队列管理器需要对它可能需要恢复的所有数据集进行 UPDATE 访问

创建共享消息数据集

每个共享消息数据集通常应该在创建或更改相应的 **CFSTRUCT** 定义之前创建，以允许使用这种形式的消息卸载，因为 **CFSTRUCT** 定义更改通常会立即生效，并且只要队列管理器尝试访问已分配给该结构的共享队列，就需要该数据集。 在 SCSQPROC (CSQ4SMDS) 中提供了用于分配和预先格式化共享消息数据集的样本作业。 必须定制并运行该作业，以便为使用 CFSTRUCT 和 OFFLOAD (SMDS) 的每个队列管理器分配共享消息数据集。

如果队列管理器发现已启用卸载支持，并尝试打开其共享消息数据集但尚未创建该数据集，那么共享消息数据集将标记为不可用。 然后，在创建数据集并通知队列管理器重试 (例如使用 **START SMDSCONN** 命令) 之前，队列管理器将无法存储任何大型消息。

共享消息数据集是使用 Access Method Services **DEFINE CLUSTER** 命令作为 VSAM 线性数据集创建的。 定义必须指定 **SHAREOPTIONS(2 3)** 以允许一个队列管理器打开该队列管理器以进行写访问，并允许任意数量的队列管理器同时读取该队列管理器。 必须使用缺省控制区间大小 4 KB。 如果数据集可能需要扩展至超过 4 GB，那么必须使用具有 VSAM 扩展可寻址性属性的 SMS 数据类来定义该数据集。 共享消息数据集有资格驻留在扩展地址卷 (EAV) 的扩展寻址空间 (EAS) 部分中。

在初始使用之前，每个共享消息数据集都可以为空或预先格式化为二进制零 (使用 **CSQJUFMT** 或类似实用程序，例如样本作业 SCSQPROC (CSQ4SMDS))。 如果它为空或仅在打开时部分格式化，那么队列管理器会自动将剩余空间格式化为二进制零。

共享消息数据集性能和容量注意事项

每个共享消息数据集用于存储由拥有队列管理器从同一系统中的区域写入关联 **CFSTRUCT** 的共享消息的卸载数据。 每个消息的存储数据包括描述符 (当前约 350 字节)，消息头和消息体。 每个已卸载的消息都存储在数据集中的一个或多个页面 (大小为 4 KB 的物理块) 中。

因此，可以通过将整体消息大小 (包括描述符) 舍入到 4 KB 的下一个倍数，然后乘以消息数来估算给定数量的已卸载消息所需的数据集空间。

对于页集，当共享消息数据集几乎已满时，可以选择自动展开。 可以使用 **CFSTRUCT** 定义上的 **DSEXPAND** 参数来设置此自动扩展的缺省行为。 可以使用 **ALTER SMDS** 命令上的 **DSEXPAND** 参数来覆盖每个队列管理器的此设置。 当数据集达到 90% 已满并且需要更多空间时，将触发自动扩展。 如果允许扩展，但 VSAM 拒绝了扩展尝试，因为在定义数据集时未指定辅助空间分配，那么将使用数据集当前大小的 20% 的辅助分配重试扩展。

如果共享消息数据集是使用扩展可寻址性属性定义的，那么最大大小仅由 VSAM 注意事项限制为最大 16 TB 或 59 个卷。 这明显大于本地页集的 64 GB 最大大小。

激活共享消息数据集

当队列管理器成功连接到应用程序耦合设施结构时，它将使用关联的 **DSGROUP** 参数来检查该结构定义是否指定卸载。 如果是这样，那么队列管理器将分配并打开自己的共享消息数据集以进行写访问，然后打开其他队列管理器拥有的任何现有共享消息数据集以进行读访问。

首次打开共享消息数据集时 (在将其记录为在队列共享组中处于活动状态之前)，第一页将不包含有效的头。 队列管理器填充头信息以标识队列共享组，结构名称和拥有队列管理器。

完成头后，队列管理器将新的共享消息数据集注册为活动数据集，并广播一个事件以通知任何其他活动队列管理器有关新数据集的信息。

每次队列管理器打开共享消息数据集时，它都会验证头信息，以确保仍在使用正确的数据集，并且该数据集尚未损坏。

z/OS 规划*“Db2”*环境

如果您正在使用队列共享组，那么 IBM MQ 需要连接到作为数据共享组成员的 Db2 子系统。 使用本主题可帮助了解用于保存 IBM MQ 数据的 Db2 需求。

IBM MQ 需要知道要连接到的数据共享组的名称，以及要连接到此数据共享组的 Db2 子系统 (或 Db2 组) 的名称。 这些名称在 CSQ6SYSP 系统参数宏的 QSGDATA 参数中指定 (在 使用 CSQ6SYSP 中描述)。

在数据共享组中，共享 Db2 表用于保存:

- 队列共享组的配置信息。
- IBM MQ 共享对象和组对象的属性。
- (可选) 与卸载的 IBM MQ 消息相关的数据。

V 9.2.0 IBM MQ 提供了一组样本作业，用于定义必需的 Db2 表空间，表和索引。 这些作业使用通用表空间 (UTS)。 产品的早期版本有两组作业，一组用于 UTS，另一组用于较旧类型的表空间，最新版本的 Db2 已不推荐使用这些作业。

V 9.2.0 IBM MQ 仍可与较旧类型的表空间配合使用，如果您已有现有队列共享组，那么这可能适用。 但是，如果要创建新的队列共享组，那么它应该使用 UTS。

V 9.2.0 Db2 V12 功能级别 508 现在提供了一个非中断性迁移过程，用于将多表表空间迁移到通用表空间。 您可以使用此方法将现有队列共享组所使用的多表表空间迁移到通用表空间，而不必中断整个队列共享组。

缺省情况下， Db2 使用运行作业的人员的用户标识作为 Db2 资源的所有者。 如果删除此用户标识，那么将删除与其关联的资源，因此将删除该表。 请考虑使用组标识来拥有表，而不是单个用户标识。 您可以通过将 `GROUP=groupname` 添加到 JOB 卡，并在任何 SQL 语句之前指定 `SET CURRENT SQLID='groupname'` 来执行此操作。

IBM MQ 使用 Db2 的 RRS 连接设施。 这意味着您可以指定要连接到的 Db2 组的名称。 连接到 Db2 组连接名称 (而不是特定 Db2 子系统) 的优点是 IBM MQ 可以连接 (或重新连接) 到属于该组的 z/OS 映像上的任何可用 Db2 子系统。 在要运行队列共享 IBM MQ 子系统的每个 z/OS 映像上，必须有一个作为数据共享组成员的 Db2 子系统，并且 RRS 必须处于活动状态。

Db2 存储器

对于大多数安装，需要的 Db2 存储量约为 3390 设备上的 20 或 30 个柱面。 但是，如果要计算存储需求，下表提供了一些信息以帮助您确定 Db2 IBM MQ 数据所需的存储量。 该表描述了每个 Db2 行的长度，以及在相关 Db2 表中添加或删除每行时的长度。 请将此信息与 *Db2 for z/OS Installation Guide* 中有关计算 Db2 表及其索引的空间需求的信息一起使用。

表 *25:* 规划 *Db2* 存储需求

Db2 表名称	**行的长度**	**在下列情况下，将添加行:**	**在以下情况下，将删除行:**
CSQ.ADMIN_B_QSG	252 字节	使用 CSQ5PQSG 实用程序的 ADD QSG 函数将队列共享组添加到表中。	使用 CSQ5PQSG 实用程序的 REMOVE QSG 函数从表中除去队列共享组。 (在删除队列共享组记录时，将自动从所有其他 Db2 表中删除与此队列共享组相关的所有行。)
CSQ.ADMIN_B_QMGR	最多 3828 个字节	使用 CSQ5PQSG 实用程序的 ADD QMGR 函数将队列管理器添加到表中。	将使用 CSQ5PQSG 实用程序的 REMOVE QMGR 函数从表中除去队列管理器。
CSQ.ADMIN_B_STRUCTURE	1454 字节	定义了指定 QSGDISP (SHARED) 属性的第一个本地队列定义，该定义用于指定队列共享组中先前未知的结构。	将删除指定 QSGDISP (SHARED) 属性的最后一个本地队列定义，该定义用于指定队列共享组中的结构。
CSQ.ADMIN_B_SCST	342 字节	已启动共享通道。	共享通道变为不活动状态。
CSQ.ADMIN_B_SSKT	254 字节	将启动具有 NPMSPEED (NORMAL) 属性的共享通道。	具有 NPMSPEED (NORMAL) 属性的共享通道将变为不活动状态。
CSQ.ADMIN_B_STRBACKUP	514 字节	将向 CSQ.ADMIN_B_STRUCTURE 表。 在运行 BACKUP CFSTRUCT 命令之前，每个条目都是一个哑元条目，这将覆盖哑元条目。	将从 CSQ.ADMIN_B_STRUCTURE 表。
CSQ.OBJ_B_AUTHINFO	3400 字节	定义了具有 QSGDISP (GROUP) 的认证信息对象。	将删除具有 QSGDISP (GROUP) 的认证信息对象。

表 25: 规划 Db2 存储需求 (继续)			
Db2 表名称	**行的长度**	**在下列情况下，将添加行:**	**在以下情况下，将删除行:**
CSQ.OBJ_B_QUEUE	最多 3707 字节	• 定义了具有 QSGDISP (GROUP) 属性的队列。 • 定义了具有 QSGDISP (SHARED) 属性的队列。 • 将打开具有 DEFTYPE (SHAREDYN) 属性的模型队列。	• 将删除具有 QSGDISP (GROUP) 属性的队列。 • 将删除具有 QSGDISP (SHARED) 属性的队列。 • 使用 DELETE 选项关闭具有 DEFTYPE (SHAREDYN) 属性的动态队列。
CSQ.OBJ_B_NAMELIST	最多 15127 字节	定义了具有 QSGDISP (GROUP) 属性的名称列表。	将删除具有 QSGDISP (GROUP) 属性的名称列表。
CSQ.OBJ_B_CHANNEL	最多 14127 个字节	定义了具有 QSGDISP (GROUP) 属性的通道。	将删除具有 QSGDISP (GROUP) 属性的通道。
CSQ.OBJ_B_STGCLASS	最多 2865 个字节	定义了具有 QSGDISP (GROUP) 属性的存储类。	将删除具有 QSGDISP (GROUP) 属性类的存储类。
CSQ.OBJ_B_PROCESS	最多 3347 个字节	定义了具有 QSGDISP (GROUP) 属性的进程。	将删除具有 QSGDISP (GROUP) 属性的进程。
CSQ.OBJ_B_TOPIC	最多 14520 字节	定义了具有 QSGDISP (GROUP) 属性的主题对象。	将删除具有 QSGDISP (GROUP) 属性的主题对象。
CSQ.EXTEND_B_QMGR	小于 430 字节	使用 CSQ5PQSG 实用程序的 ADD QMGR 函数将队列管理器添加到表中。	将使用 CSQ5PQSG 实用程序的 REMOVE QMGR 函数从表中除去队列管理器。
CSQ.ADMIN_B_MESSAGES	87 字节	对于大型消息 PUT (每个 BLOB 1)。	对于大型消息 GET (每个 BLOB 1)。
CSQ.ADMIN_MSGS_BAUX1 CSQ.ADMIN_MSGS_BAUX2 CSQ.ADMIN_MSGS_BAUX3 CSQ.ADMIN_MSGS_BAUX4		这些 4 表包含为消息的每个 BLOB 添加到其中一个 4 表中的大型消息的消息有效内容。 BLOBS 最大长度为 511 KB， 因此如果消息大小大于 711 KB， 那么此消息将有多个 BLOB。	

使用大量大小大于 63 KB 的共享队列消息可能会对 IBM MQ 系统产生重大性能影响。 有关更多信息，请参阅 SupportPac MP16， Capacity Planning and Tuning for IBM MQ for z/OS，网址为: SupportPacs for IBM MQ 和其他项目区域。

z/OS 规划备份和恢复

在您的站点上开发备份和恢复过程对于避免成本高昂且耗时的数据丢失至关重要。 IBM MQ 提供了在系统故障后将队列和消息恢复到其当前状态的方法。

本主题包含以下几部分:

• 第 161 页的『恢复过程』
• 第 161 页的『备份和恢复提示』

- 第 163 页的『恢复页集』
- 第 164 页的『恢复 CF 结构』
- 第 164 页的『实现具体恢复目标』
- 第 165 页的『其他产品的备份注意事项』
- 第 166 页的『恢复和 CICS』
- 第 166 页的『恢复和 IMS』
- 第 166 页的『准备在备用站点上进行恢复』
- 第 166 页的『队列管理器备份活动的示例』

恢复过程

为 IBM MQ 开发以下过程:

- 创建恢复点。
- 正在备份页集。
- 正在备份 CF 结构。
- 正在恢复页集。
- 从空间不足情况 (IBM MQ 个日志和页集) 中恢复。
- 正在恢复 CF 结构。

有关这些信息，请参阅 管理 IBM MQ for z/OS 。

熟悉站点上用于以下内容的过程:

- 从硬件或电源故障中恢复。
- 从 z/OS 组件故障中恢复。
- 使用非现场恢复从站点中断中恢复。

z/OS 备份和恢复提示

使用本主题来了解一些备份和恢复任务。

队列管理器重新启动进程通过将日志信息应用于页集来将数据恢复到一致状态。 如果页集已损坏或不可用，那么可以使用页集的备份副本 (如果所有日志都可用) 来解决问题。 如果日志数据集已损坏或不可用，那么可能无法完全恢复。

请考虑下列几点:

- 定期生成备份副本
- 请勿废弃您可能需要的归档日志
- 请勿将 DDname 更改为页集关联

定期获取备份副本

恢复点 是用于描述 IBM MQ 页集的一组备份副本以及恢复这些页集所需的相应日志数据集的术语。 这些备份副本提供了潜在的重新启动点，以防发生页集丢失事件（例如，页集 I/O 错误）。 如果使用这些备份副本重新启动队列管理器，那么 IBM MQ 中的数据将保持一致，直到生成这些副本为止。 如果所有日志都从此点可用，那么可以将 IBM MQ 恢复到故障点。

最近的备份副本，更快的 IBM MQ 可以恢复页集中的数据。 页集的恢复取决于可用的所有必需日志数据集。

在规划恢复时，您需要确定获取备份副本的频率以及要保留的完整备份周期数。 这些值告诉您必须保留日志数据集和页集备份副本以进行 IBM MQ 恢复的时间长度。

在决定获取备份副本的频率时，请考虑恢复页集所需的时间。 所需时间由以下各项确定:

- 要遍历的日志量。
- 操作员安装和除去归档磁带卷所需的时间。
- 读取恢复所需的日志部分所需的时间。
- 重新处理已更改的页面所需的时间。
- 用于备份副本的存储介质。
- 用于创建和复原备份副本的方法。

通常，生成备份副本的频率越低，恢复所花费的时间就越少，但生成副本所花费的时间就越多。

对于每个队列管理器，您应该获取以下内容的备份副本:

- 归档日志数据集
- 归档时创建的 BSDS 副本
- 页集
- 您的对象定义
- 您的 CF 结构

要降低备份副本丢失或损坏的风险，请考虑:

- 将备份副本存储在不同存储卷上的原始副本。
- 将备份副本存储在与原始副本不同的站点上。
- 至少生成页集的每个备份的两个副本，如果您使用的是单个日志记录或单个 BSDS ，那么将生成归档日志和 BSDS 的两个副本。 如果您正在使用双重日志记录或 BSDS ，请生成归档日志或 BSDS 的单一副本。

在将 IBM MQ 移至生产环境之前，请完全测试并记录备份过程。

备份页集

您需要定期备份页集。 一些企业每天两次备份页面集。

您需要自备份以来的活动日志和归档日志才能使用备份进行恢复。 如果在队列管理器运行时进行了备份，那么您需要足够的日志数据来返回四个检查点。

您可以使用 ADRDSSU FastReplication 来备份页集，并且可以在队列管理器处于活动状态时执行此操作。 请注意，您需要确保存储池中有足够的空间。

备份对象定义

创建对象定义的备份副本。 为此，请使用实用程序的 COMMAND 函数的 MAKEDEF 功能 (如 使用 CSQUTIL 的 COMMAND 函数中所述)。

每当生成队列管理器数据集的备份副本并保留最新版本时，都应该执行此操作。

备份耦合设施结构

如果已设置任何队列共享组，即使未使用这些组，也必须对 CF 结构进行定期备份。 为此，请使用 IBM MQ BACKUP CFSTRUCT 命令。 只能在使用 RECOVER (YES) 属性定义的 CF 结构上使用此命令。 如果持久共享消息的任何 CF 条目引用存储在共享消息数据集 (SMDS) 或 Db2 中的已卸载消息数据，那么将检索已卸载数据并将其与 CF 条目一起备份。 不应单独备份共享消息数据集。

建议您每小时备份一次所有 CF 结构，以最大程度地缩短复原 CF 结构所需的时间。

您可以在单个队列管理器上执行所有 CF 结构备份，这具有将日志使用增加限制为单个队列管理器的优点。 或者，您可以在队列共享组中的所有队列管理器上执行备份，这具有在队列共享组中分散工作负载的优点。 无论您使用哪种策略， IBM MQ 都可以找到备份并从队列共享组中的任何队列管理器执行 RECOVER CFSTRUCT。 需要访问队列共享组中所有队列管理器的日志以恢复 CF 结构。

备份消息安全策略

如果要使用 Advanced Message Security 来创建消息安全策略的备份，请使用 消息安全策略实用程序 (CSQ0UTIL) 创建备份以运行带有 -export 参数的 **dspmqspl** ，然后保存输出到 EXPORT DD 的策略定义。

每当您生成队列管理器数据集的备份副本并保留最新版本时，都应该创建消息安全策略的备份。

请勿废弃您可能需要的归档日志

IBM MQ 可能需要在重新启动期间使用归档日志。 您必须保留足够的归档日志，以便可以完全复原系统。 IBM MQ 可以使用归档日志从复原的备份副本中恢复页集。 如果已废弃该归档日志，那么 IBM MQ 无法将页集复原到其当前状态。 废弃归档日志数据集中描述了废弃归档日志的时间和方式。

您可以使用 `/cpf DIS USAGE TYPE(ALL)` 命令来显示恢复队列管理器的页集和队列共享组的结构所需的日志 RBA 和日志范围序号 (LRSN)。 然后，应该使用 打印日志映射实用程序 (CSQJU004) 来打印引导数据集 (BSDS) 信息，以便队列管理器找到包含日志 RBA 的日志。

对于 CF 结构，需要在队列共享组中的每个队列管理器上运行 CSQJU004 实用程序以查找包含 LRSN 的日志。 您需要这些日志和任何后续日志才能恢复页集和结构。

请勿将 DDname 更改为页集关联

IBM MQ 将页集编号 00 与 DDname CSQP0000 关联，将页集编号 01 与 DDname CSQP0001 关联，以此类推，直至 CSQP0099。 IBM MQ 根据与页集相关联的 DDname 来写入页集的恢复日志记录。 因此，不得移动已与 PSID DDname 关联的页集。

z/OS 恢复页集

使用本主题来了解恢复页面集时涉及的因素，以及如何最大程度地缩短重新启动时间。

恢复策略中的一个关键因素是您可以容忍队列管理器中断的时间。 总中断时间可能包括从备份恢复页集或在异常终止后重新启动队列管理器所花费的时间。 影响重新启动时间的因素包括您备份页集的频率以及在检查点之间写入日志的数据量。

要将异常终止后的重新启动时间最小化，请使工作单元保持较短，以便在系统重新启动时最多使用两个活动日志。 例如，如果要设计 IBM MQ 应用程序，请避免在第一个同步点 MQI 调用与落实点之间放置具有较长等待时间间隔的 `MQGET` 调用，因为这可能会导致工作单元具有较长的持续时间。 导致工作单元长的另一个常见原因是通道启动程序的批处理时间间隔超过 5 分钟。

可以使用 DISPLAY THREAD 命令来显示工作单元的 RBA 并帮助解决旧的工作单元。

您必须多久备份一次页集?

如果需要合理短的恢复时间，那么频繁的页集备份至关重要。 即使当页集非常小或者该页集中的队列上存在少量活动时，也适用。

如果在页集中使用持久消息，那么备份频率应该以小时为单位，而不是以天为单位。 对于页集零也是如此。

要计算近似备份频率，请从确定目标总恢复时间开始。 这包括以下内容:

1. 对问题作出反应所花费的时间。
2. 复原页集备份副本所花费的时间。

   如果使用 SnapShot 备份/复原，那么执行此任务所需的时间为几秒。 有关 SnapShot 的信息，请参阅 *DFSMSdss Storage Administration Guide*。
3. 队列管理器重新启动所需的时间，包括恢复页集所需的额外时间。

   这主要取决于自上次备份该页集以来必须从活动日志和归档日志中读取的日志数据量。 除了与受损页集直接关联的日志数据之外，还必须读取所有此类日志数据。

   注: 使用 模糊备份 (在工作单元处于活动状态时生成日志和页集的快照) 时，可能需要读取最多三个额外的检查点，这可能导致需要读取一个或多个额外的日志。

在决定允许恢复页集的时间长度时，需要考虑的因素包括:

- 在正常处理期间将数据写入活动日志的速率取决于消息到达系统中的方式以及消息速率。

  通过通道接收或发送的消息导致比在本地生成和检索的消息更多的数据日志记录。

- 可以从归档日志和活动日志中读取数据的速率。

  读取日志时，可实现的数据速率取决于所使用的设备以及特定 DASD 子系统上的总负载。

  对于大多数磁带机，对于具有较大块大小的归档日志，可以实现更高的数据速率。 但是，如果需要归档日志进行恢复，那么还必须读取活动日志上的所有数据。

z/OS 恢复 CF 结构

使用本主题来了解 CF 结构的恢复过程。

队列共享组中必须至少有一个队列管理器处于活动状态才能处理 RECOVER CFSTRUCT 命令。 CF 结构恢复不会影响队列管理器重新启动时间，因为恢复由已处于活动状态的队列管理器执行。

恢复过程由两个由 RECOVER CFSTRUCT 命令管理的逻辑步骤组成:

1. 正在查找和复原备份。
2. 从已使用 CF 结构的队列共享组中所有队列管理器的日志合并对 CF 结构上保留的持久消息的所有已记录更新，并将更改应用于备份。

第二步可能需要更长时间，因为可能需要读取大量日志数据。 如果频繁备份和/或同时恢复多个 CF 结构，那么可以减少所花费的时间。

执行恢复的队列管理器会使用 Db2 中的数据和引导数据集在所有其他队列管理器的日志上查找相关的备份。 此队列管理器以正确的时间顺序跨队列共享组回放这些备份，并正好从上次备份之前一直回放到故障点。

恢复 CF 结构所需的时间取决于必须重放的恢复日志数据量，而这又取决于备份的频率。 在最坏的情况下，读取队列管理器日志的时间与写入日志的时间一样长。 因此，例如，如果您有一个包含 6 个队列管理器的队列共享组，那么一个小时的日志活动可能需要 6 个小时才能重放。 通常需要的时间少于此，因为可以批量进行读取，并且可以并行读取不同的队列管理器的日志。 作为起点，我们建议您每小时备份一次 CF 结构。

当存在失败的 CF 结构时，所有队列管理器都可以继续使用其他 CF 结构中的非共享队列和队列。 如果管理结构也失败，那么必须先启动队列共享组中的至少一个队列管理器，然后才能发出 RECOVER CFSTRUCT 命令。

备份 CF 结构可能需要相当大的日志写入容量，因此可能会对执行备份的队列管理器施加很大的负载。 选择负载很轻的队列管理器来执行备份; 对于繁忙的系统，请向队列共享组添加额外的队列管理器，并将其专用于执行备份。

z/OS 实现具体恢复目标

使用本主题以获取有关如何通过调整备份频率来实现特定恢复目标时间的指导。

如果要实现特定恢复目标 (例如，在 *xx* 秒内的正常启动时间之外完成队列管理器恢复和重新启动处理) ，那么可以使用以下计算来估算备份频率 (以小时计):

```
Formula (A)
                    Required restart time * System recovery log read rate
                          (in secs)                  (in MB/sec)
Backup frequency  = -----------------------------------------------------
   (in hours)                Application log write rate (in MB/hour)
```

注: 下面提供的示例旨在突出显示需要频繁备份页集。 计算假定大多数日志活动都派生自大量持久消息。 但是，存在日志活动量不易计算的情况。 例如，在队列共享组环境中，除了其他资源之外，在其中更新共享队列的工作单元可能会导致将 UOW 记录写入 IBM MQ 日志。 因此，公式 (A) 中的应用程序日志写入速率只能从观察到的 IBM MQ 日志填充速率中准确得出。

例如，请考虑 IBM MQ MQI clients 在其中每秒生成 100 条持久消息的总负载的系统。 在这种情况下，将在本地生成所有消息。

如果每条消息的用户长度为 1 KB，那么每小时记录的数据量大约为:

```
100 * (1 + 1.3) KB * 3600 = approximately 800 MB

where
      100            = the message rate a second
      (1 + 1.3) KB   = the amount of data logged for
                       each 1 KB of persistent messages
```

考虑总体目标恢复时间为 75 分钟。 如果您已允许 15 分钟来对问题作出反应并复原页集备份副本，那么队列管理器恢复和重新启动必须在 60 分钟 (3600 秒) 内完成，并应用公式 (A)。 假定所有必需的日志数据都在 RVA2-T82 DASD 上，而该 DASD 的恢复速率大约为每秒 2.7 MB，那么这需要至少每隔一个页集备份频率:

```
3600 seconds * 2.7 MB a second / 800 MB an hour = 12.15 hours
```

如果 IBM MQ 应用程序日大约持续 12 小时，那么每天一个备份是合适的。 但是，如果应用程序日持续 24 小时，那么每天两次备份更合适。

另一个示例可能是生产系统，其中所有消息都用于请求/应答应用程序 (即，在接收方通道上接收持久消息，生成持久应答消息并将其发送到发送方通道下)。

在此示例中，实现的批处理大小为 1，因此每个消息都有一个批处理。 如果每秒有 50 个请求应答，那么总负载为每秒 100 个持久消息。 如果每条消息的长度为 1 KB，那么每小时记录的数据量大约为:

```
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
```

要实现队列管理器恢复并在 30 分钟 (1800 秒) 内重新启动，再次假定所有必需的日志数据都在 RVA2-T82 DASD 上，这要求至少每执行一次页集备份:

```
1800 seconds * 2.7 MB a second / 1500 MB an hour = 3.24 hours
```

定期审查备份频率

监视 IBM MQ 日志使用情况 (以每小时 MB 为单位)。 如果需要，请定期执行此检查并修改页集备份频率。

z/OS 其他产品的备份注意事项

如果将 IBM MQ 与 CICS 或 IMS 配合使用，那么还必须考虑这些产品对备份策略的影响。 数据设施分层存储管理器 (DFHSM) 管理数据存储器，并且可以与 IBM MQ 所使用的存储器进行交互。

使用 DFHSM 进行备份和恢复

数据设施分层存储管理器 (DFHSM) 在系统中的存储设备之间执行自动空间可用性和数据可用性管理。 如果使用它，那么您需要知道它会自动将数据移动到 IBM MQ 存储器以及从该存储器移动数据。

DFHSM 通过将最近未使用的数据集移动到备用存储器来高效管理 DASD 空间。 它还通过自动将新的或已更改的数据集复制到磁带或 DASD 备份卷，使您的数据可用于恢复。 它可以删除数据集，或者将它们移动到另一个设备。 其操作每天在指定时间发生，并允许在删除或移动数据集之前将其保留预定时间段。

您还可以手动执行所有 DFHSM 操作。 *Data Facility* 分层存储管理器用户指南 说明如何使用 DFHSM 命令。如果将 DFHSM 与 IBM MQ 配合使用，请注意 DFHSM 执行以下操作:

- 使用已编目的数据集。
- 对页集和日志进行操作。
- 支持 VSAM 数据集。

恢复和 CICS

CICS 资源的恢复不受 IBM MQ 的存在影响。 CICS 将 IBM MQ 识别为非 CICS 资源 (或外部资源管理器)，并包含 IBM MQ 作为使用 CICS 资源管理器接口 (RMI) 的任何同步点协调请求的参与者。 有关 CICS 恢复的更多信息，请参阅 *CICS Recovery and Restart Guide*。 有关 CICS 资源管理器接口的信息，请参阅 *CICS* 定制指南。

恢复和 IMS

IMS 将 IBM MQ 识别为外部子系统，并将其识别为同步点协调的参与者。 在 *IMS* 定制指南中描述了外部子系统资源的 IMS 恢复。

z/OS 准备在备用站点上进行恢复

如果 IBM MQ 计算中心完全丢失，那么可以在恢复站点上的另一个 IBM MQ 系统上进行恢复。

要在恢复站点上恢复 IBM MQ 系统，必须定期备份页集和日志。 与所有数据恢复操作一样，灾难恢复的目标是尽可能少地丢失数据，工作负载处理 (更新) 和时间。

在恢复站点:

- 恢复 IBM MQ 队列管理器 **必须** 与丢失的队列管理器同名。
- 确保恢复队列管理器上使用的系统参数模块包含与丢失的队列管理器相同的参数。

管理 IBM MQ for z/OS 中描述了灾难恢复过程。

z/OS 队列管理器备份活动的示例

本主题显示为队列管理器备份活动的示例。

规划队列管理器备份策略时，关键考虑因素是保留正确数量的日志数据。 管理日志 描述了如何通过引用队列管理器的系统恢复 RBA 来确定需要哪些日志数据集。 IBM MQ 使用以下信息确定系统恢复 RBA:

- 当前处于活动状态的工作单元。
- 尚未从缓冲池清空到磁盘的页集更新。
- CF 结构备份，以及此队列管理器的日志是否包含使用它们的任何恢复操作中所需的信息。

必须保留足够的日志数据才能执行介质恢复。 当系统恢复 RBA 随时间推移而增加时，仅当执行后续备份时，必须保留的日志数据量才会减少。 CF 结构备份由 IBM MQ 管理，因此在报告系统恢复 RBA 时将予以考虑。 这意味着在实践中，仅当执行页集备份时，必须保留的日志数据量才会减少。

第 167 页的图 43 显示了作为队列共享组成员的队列管理器上的备份活动示例，恢复 RBA 如何随每个备份而变化以及如何影响必须保留的日志数据量。 在此示例中，队列管理器使用本地资源和共享资源: 页集和两个 CF 结构 STRUCTURE1 和 STRUCTURE2。

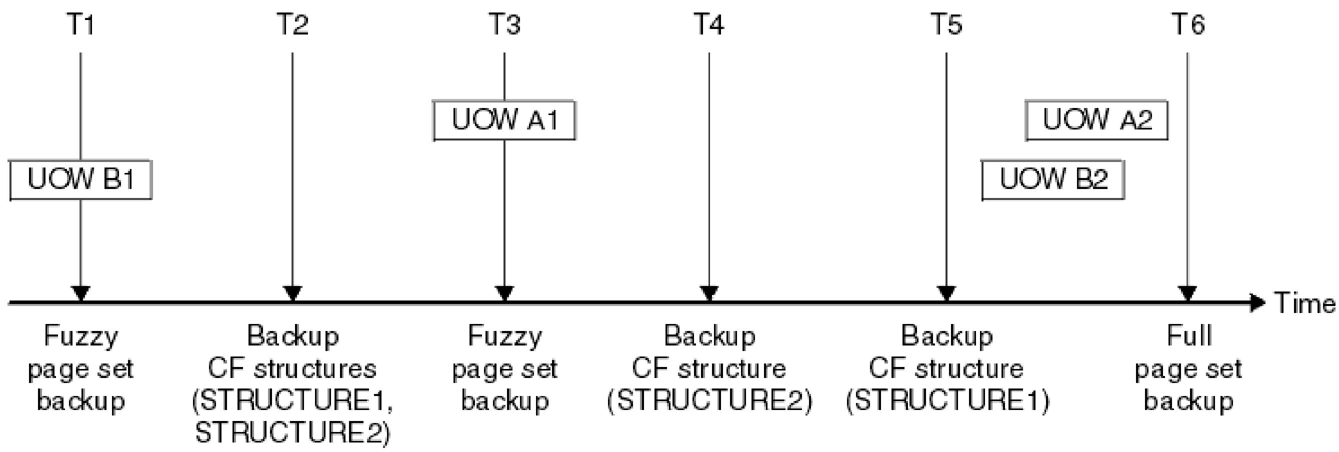


图 *43:* 队列管理器备份活动的示例

这是在每个时间点发生的情况:

时间点 T1

将创建页集的模糊备份，如 如何备份和恢复页集中所述。

队列管理器的系统恢复 RBA 是以下最低值:

- 此时正在备份的页集的恢复 RBA。
- 恢复 CF 应用程序结构所需的最低恢复 RBA。 这与恢复先前创建的 STRUCTURE1 的备份相关。
- 队列管理器 (UOWB1) 中最旧的当前活动工作单元的恢复 RBA。

此时间点的系统恢复 RBA 由 DISPLAY USAGE 命令发出的消息提供，该命令是模糊备份过程的一部分。

时间点 T2

将创建 CF 结构的备份。 首先备份 CF 结构 STRUCTURE1，然后备份 STRUCTURE2。

必须保留的日志数据量保持不变，因为仍需要在 T1 处从系统恢复 RBA 确定相同的数据才能使用在 T1 处进行的页集备份进行恢复。

时间点 T3

将创建另一个模糊备份。

队列管理器的系统恢复 RBA 是以下最低值:

- 此时正在备份的页集的恢复 RBA。
- 恢复 CF 结构 STRUCTURE1 所需的最低恢复 RBA，因为 STRUCTURE1 是在 STRUCTURE2 之前备份的。
- 队列管理器 (UOWA1) 中最早的当前活动工作单元的恢复 RBA。

此时间点的系统恢复 RBA 由 DISPLAY USAGE 命令发出的消息提供，该命令是模糊备份过程的一部分。

现在，您可以减少保留的日志数据，如此新系统恢复 RBA 所确定。

时间点 T4

备份 CF 结构 STRUCTURE2。 用于恢复最旧的必需 CF 结构备份的恢复 RBA 与在时间 T2 备份的 CF 结构 STRUCTURE1 的备份相关。

创建此 CF 结构备份不会影响必须保留的日志数据量。

时间点 T5

对 CF 结构 STRUCTURE1 进行备份。 用于恢复最旧的所需 CF 结构备份的恢复 RBA 现在与恢复在 T4 时备份的 CF 结构 STRUCTURE2 相关。

创建此 CF 结构备份不会影响必须保留的日志数据量。

时间点 T6

将对页集进行完全备份，如 如何备份和恢复页集中所述。

队列管理器的系统恢复 RBA 是以下最低值:

- 此时正在备份的页集的恢复 RBA。
- 恢复 CF 结构所需的最低恢复 RBA。 这与 CF 结构 STRUCTURE2 的恢复相关。
- 队列管理器中最早的当前活动工作单元的恢复 RBA。 在这种情况下，没有当前的工作单元。

此时间点的系统恢复 RBA 由 DISPLAY USAGE 命令发出的消息提供，该命令是完全备份过程的一部分。

同样，可以减少保留的日志数据，因为与完全备份关联的系统恢复 RBA 是最新的。

z/OS 规划“z/OS UNIX”环境

IBM MQ 队列管理器，通道启动程序和 mqweb 服务器中的某些进程使用 z/OS UNIX System Services (z/OS UNIX) 进行正常处理。

队列管理器和通道启动程序启动式任务用户标识需要定义了 UID 的 OMVS 段才能访问 z/OS UNIX。 用户标识在 z/OS UNIX 中不需要特殊许可权。

注: 虽然队列管理器和通道启动程序使用 z/OS UNIX 工具 (例如，与 TCP/IP 服务进行接口) ，但它们不需要访问 z/OS UNIX 文件系统中 IBM MQ 安装目录的任何内容。 因此，队列管理器和通道启动程序不需要任何配置来指定 z/OS UNIX 文件系统的路径。

托管 IBM MQ Console 和 REST API 的 mqweb 服务器使用 z/OS UNIX 文件系统中 IBM MQ 安装目录中的文件。 它还需要访问另一个用于存储数据 (例如，配置和日志文件) 的文件系统。 需要定制 mqweb 启动式任务 JCL 以引用这些 z/OS UNIX 文件系统。

z/OS UNIX 文件系统中 IBM MQ 目录的内容也由连接到 IBM MQ 的应用程序使用。 例如，使用 IBM MQ classes for Java 或 IBM MQ classes for JMS 接口的应用程序。

请参阅以下主题以获取相关配置指示信息:

- 与 IBM MQ classes for Java 相关的环境变量
- IBM MQ classes for Java 库
- 设置环境变量
- 配置 Java 本机接口 (JNI) 库

z/OS 规划 Advanced Message Security

TLS (或 SSL) 可用于加密和保护网络上流动的消息，但当消息位于队列上 ("静态") 时，这不会保护消息。 Advanced Message Security (AMS) 保护消息，使其从第一次放入队列的时间开始，直到得到消息为止，以便只有消息的预期收件人才能读取该消息。 在放入处理期间对消息进行加密和签名，并在获取处理期间不受保护。

可以配置 AMS 以通过不同方式保护消息:

1. 可以对消息进行签名。 消息是明文的，但有一个校验和，是签名的。 这允许检测消息内容中的任何更改。 从已签名的内容中，您可以识别对数据进行签名的人员。
2. 可以对消息进行加密。 没有解密密钥的任何人都看不到这些内容。 将为每个收件人加密解密密钥。
3. 可以对消息进行加密和签名。 他们的解密密钥将针对每个收件人进行加密，并且从签名中您可以识别发送消息的人员。

加密和签名使用数字证书和密钥环。

您可以将客户机设置为使用 AMS，以便在将数据放入客户机通道之前对数据进行保护。 可以将受保护的消息发送到远程队列管理器，您需要配置远程队列管理器以处理这些消息。

设置 AMS

AMS 地址空间用于执行 AMS 工作。 这将设置额外的安全性，以允许访问和保护密钥环和证书的使用。

通过使用实用程序 (CSQ0UTIL) 来定义队列的安全策略，可以配置要保护的队列。

设置 AMS 后

您需要为放置消息的人员以及获取消息的人员设置数字证书和密钥环。

如果 z/OS 上的用户 Alice 需要向 Bob 发送消息，那么 AMS 需要 Bob 的公用证书副本。

如果 Bob 想要处理来自 Alice 的消息，那么 AMS 需要 Alice 的公用证书或 Alice 使用的相同认证中心证书。

注意: 您需要:

- 仔细规划谁可以放入或从队列中获取
- 标识人员及其证书名称。

容易犯错，问题就很难解决。

相关概念

第 124 页的『规划队列管理器』
在设置队列管理器时，您的规划应允许队列管理器增长，以便队列管理器满足企业的需求。

z/OS 规划 Managed File Transfer

使用本部分来指导您需要如何设置系统以在 z/OS 上运行 Managed File Transfer (MFT)。

z/OS 规划 Managed File Transfer -硬件和软件需求

使用本主题作为有关需要如何在系统上设置硬件和软件需求以在 z/OS 上运行 Managed File Transfer (MFT) 的指导。

软件需求

Managed File Transfer 是使用 Java 编写的，其中包含一些用于配置和操作程序的 shell 脚本和 JCL。

要点: 您必须熟悉 z/OS UNIX System Services (z/OS UNIX) 才能配置 Managed File Transfer。 例如:

- 文件目录结构，名称如 `/u/userID/myfile.txt`
- z/OS UNIX 命令，例如:
  - `cd` (更改目录)
  - `ls`（列示）
  - `chmod` (更改文件许可权)
  - `chown` (更改文件所有权或可访问文件或目录的组)

您需要 z/OS UNIX 中的以下产品才能配置和运行 MFT:

1. Java，例如，在目录 `/java/java80_bit64_GA/J8.0_64/` 中
2. IBM MQ 9.2.0，例如，在目录 `/mqm/V9R2M0` 中
3. 如果要将 Db2 用于状态和历史记录，那么需要安装 Db2 JDBC 库，例如，安装在 `/db2/db2v10/jdbc/libs` 目录中。

产品注册

在启动时， Managed File Transfer 会检查 `sys1.parmlib`(IFAPRDxx) 并置中的注册。 以下代码是如何注册 MFT 的示例:

```
PRODUCT OWNER('IBM CORP')
NAME('WS MQ FILE TRANS')
ID(5655-MFT)
VERSION(*) RELEASE(*) MOD(*)
```

```
FEATURENAME('WS MQ FILE TRANS')
STATE(ENABLED)
```

磁盘空间

V 9.2.0 IBM MQ for z/OS 程序目录说明了 Managed File Transfer 的 DASD 和 zFS 存储需求。 有关 IBM MQ for z/OS 的程序目录的下载链接，请参阅 IBM MQ 9.2 产品文档和程序目录的 PDF 文件。

z/OS 规划 Managed File Transfer -拓扑

使用本主题作为在系统上运行 Managed File Transfer (MFT) on z/OS 所需的拓扑的指导。

Managed File Transfer 队列管理器

IBM MQ Managed File Transfer 拓扑由以下部分组成:

代理及其关联的队列管理器
代理使用其代理队列管理器上托管的系统队列来维护状态信息和接收工作请求。

一个命令队列管理器
这充当 MFT 拓扑中的网关。 它通过发送方和接收方通道或集群连接到代理队列管理器。 运行某些命令时，它们会直接连接到命令队列管理器，并向指定的代理发送消息。 此消息将通过 IBM MQ 网络路由到代理队列管理器，在该队列管理器中，代理将获取此消息并进行处理。

一个协调队列管理器
这是一个了解整个拓扑的中心集线器。 协调队列管理器通过发送方和接收方通道或使用集群连接到拓扑中的所有代理队列管理器。 代理定期向协调队列管理器发布状态信息，并将其传输模板存储在其中。

单个队列管理器可以在拓扑中执行多个角色。 例如，可以将同一队列管理器配置为拓扑的协调队列管理器和命令队列管理器。

如果您正在使用多个队列管理器，那么需要在队列管理器之间设置通道。 您可以通过使用集群或使用点到点连接来执行此操作。

使用 IBM MQ Managed File Transfer for z/OS 时，在确定要用于拓扑中不同角色的队列管理器时，需要考虑许多事项。

代理队列管理器

IBM MQ Managed File Transfer for z/OS 代理的代理队列管理器必须在 z/OS 上运行。

如果:

- 代理程序正在 IBM MQ 9.1 或更高版本上运行 Managed File Transfer for z/OS
- 并且，代理队列管理器已获得 IBM MQ Advanced for z/OS Value Unit Edition (Advanced VUE) 的许可

代理可以使用 CLIENT 传输连接到队列管理器。

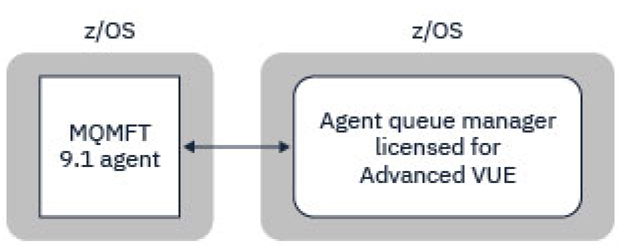


图 *44: z/OS* 上的 *MFT 9.1* 代理可以使用 *CLIENT* 传输连接到队列管理器，假定队列管理器已获得 *Advanced VUE* 的许可。

如果:

- 代理程序正在 IBM MQ 9.0 或更低版本上运行 Managed File Transfer for z/OS

- 或者，代理队列管理器正在 IBM MQ 9.0 或更高版本上运行 Managed File Transfer for z/OS， 并且代理队列管理器已获得 MFT， IBM MQ Advanced for z/OS 或 Advanced VUE 的许可

代理必须使用 BINDINGS 传输连接到队列管理器。

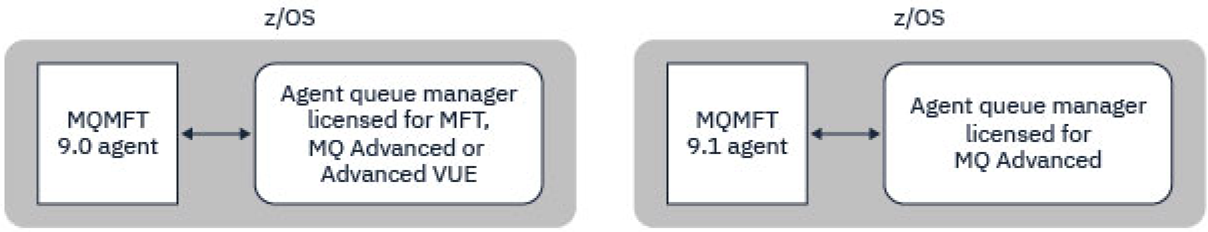


图 *45:* 具有针对 *MFT* 或 *IBM MQ Advanced* 许可的代理队列管理器的 *z/OS* 和 *9.1* 代理上的 *MFT 9.0* 代理必须使用 *BINDINGS* 传输进行连接。

命令队列管理器

哪些 MFT 命令和进程连接到哪个队列管理器 主题显示了连接到 Managed File Transfer 拓扑的命令队列管理器的所有命令。

注: 在 z/OS 上运行这些命令时， 命令队列管理器也必须在 z/OS 上。

如果命令队列管理器已获得 Advanced VUE 的许可，那么这些命令可以使用 CLIENT 传输连接到队列管理器。 否则，这些命令必须使用 BINDINGS 传输连接到命令队列管理器。

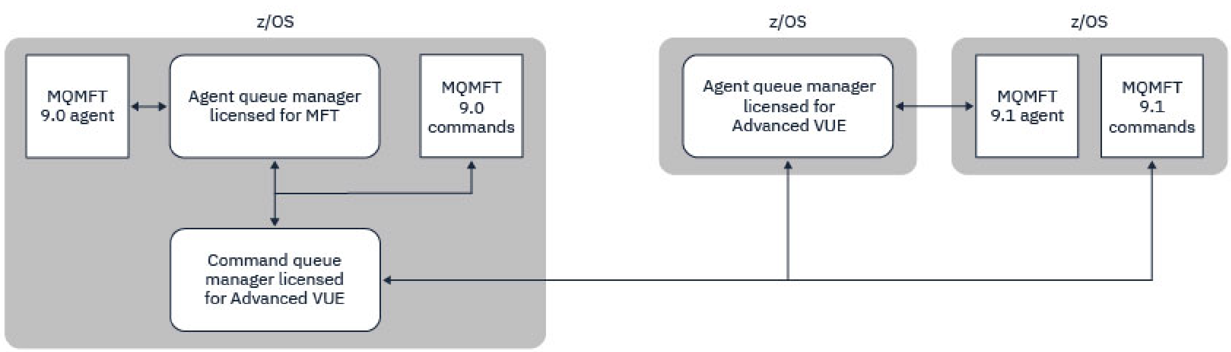


图 *46:* 命令连接到 *MFT* 拓扑的命令队列管理器。 在 *z/OS* 上运行这些命令时，命令队列管理器也必须在 *z/OS* 上

协调队列管理器

IBM MQ Managed File Transfer for z/OS 代理程序可以是其中协调队列管理器在 z/OS 上运行或在多平台上运行的拓扑的一部分。

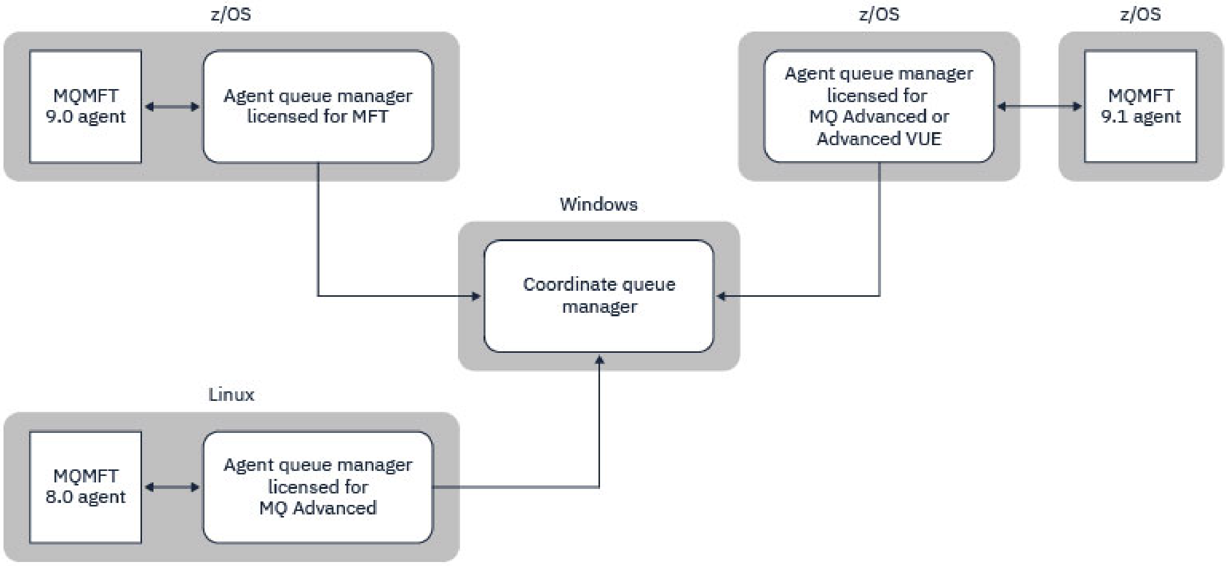


图 *47: 在 z/OS 上运行的 MFT 代理可以是其中协调队列管理器在 IBM MQ 多平台上运行的 MFT 拓扑的一部分。*

MFT 命令和进程连接到哪个队列管理器 主题显示用于连接到 Managed File Transfer 拓扑的协调队列管理器的命令。 可以在 z/OS 上运行这些命令，然后连接到在其他平台上运行的协调队列管理器。

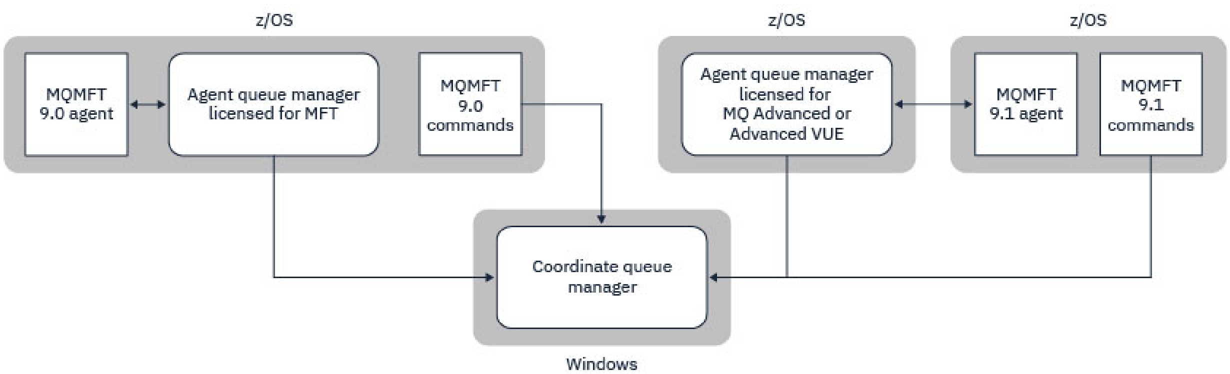


图 *48: 某些命令 (例如* ***fteListAgents****) 直接连接到 MFT 拓扑的协调队列管理器。*

我需要多少个代理?

代理执行传输数据的工作，当您请求传输数据时，指定代理的名称。

缺省情况下，代理程序可以同时处理 25 个发送请求和 25 个接收请求。 您可以配置这些进程。 请参阅 z/OS 上的 Managed File Transfer 配置选项 以获取更多信息。

如果代理程序繁忙，那么工作将排队。 处理请求所花费的时间取决于多个因素，例如，要发送的数据量，网络带宽以及网络上的延迟。

您可能希望有多个代理程序并行处理工作。

您还可以控制代理程序可以访问的资源，因此您可能希望某些代理程序使用有限的数据子集。

如果要处理具有不同优先级的请求，可以使用多个代理程序并使用工作负载管理器来设置作业的优先级。

运行代理程序

通常，代理程序是长时间运行的进程。 可以将这些进程作为以批处理方式运行的作业或作为已启动的任务提交。

z/OS 规划 Managed File Transfer -安全注意事项

使用本主题作为系统上运行 Managed File Transfer (MFT) on z/OS 所需的安全注意事项的指导。

安全性

您需要确定哪些用户标识将用于 MFT 配置和 MFT 操作。

您需要识别您传输的文件或队列，以及将向 MFT 提交传输请求的用户标识。

定制代理和记录器时，可指定允许运行 MFT 服务或执行 MFT 管理的用户组。

您应该先设置此组，然后再开始定制 MFT。 由于 MFT 使用 IBM MQ 队列，因此如果在队列管理器中启用了安全性，那么 MFT 需要访问以下资源:

表 *26: MQADMIN* 资源类

名称	需要访问权
QUEUE.SYSTEM.FTE.EVENT.agent_name	更新
QUEUE.SYSTEM.FTE.COMMAND.agent_name	更新
CONTEXT.SYSTEM.FTE.COMMAND.agent_name	更新
QUEUE.SYSTEM.FTE.STATE.agent_name	更新
QUEUE.SYSTEM.FTE.DATA.agent_name	更新
QUEUE.SYSTEM.FTE.REPLY.agent_name	更新
QUEUE.SYSTEM.FTE.AUTHAGT1.agent_name	更新
QUEUE.SYSTEM.FTE.AUTHTRN1.agent_name	更新
QUEUE.SYSTEM.FTE.AUTHOPS1.agent_name	更新
QUEUE.SYSTEM.FTE.AUTHSCH1.agent_name	更新
QUEUE.SYSTEM.FTE.AUTHMON1.agent_name	更新
QUEUE.SYSTEM.FTE.AUTHADM1.agent_name	更新

表 *27: MQQUEUE* 资源类

名称	需要访问权
SYSTEM.FTE.AUTHAGT1.agent_name	更新
SYSTEM.FTE.AUTHTRN1.agent_name	更新
SYSTEM.FTE.AUTHOPS1.agent_name	更新
SYSTEM.FTE.AUTHSCH1.agent_name	更新
SYSTEM.FTE.AUTHMON1.agent_name	更新

您可以使用用户沙箱来确定请求传输的用户可以访问文件系统的哪些部分。

要启用用户沙箱，请将 `userSandboxes=true` 语句添加到要限制的代理的 *agent.properties* 文件，并将相应的值添加到 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name/UserSandboxes.xml` 文件。

有关更多信息，请参阅 使用用户沙箱 。

此用户标识在 `UserSandboxes.xml` 文件中配置。

此 XML 文件具有诸如用户标识或用户标识 * 之类的信息以及可以使用 (包括) 或不能使用 (排除) 的资源的列表。 您需要定义可访问哪些资源的特定用户标识: 例如:

表 28: 示例用户标识以及对特定资源的访问权			
用户标识	**访问**	**包含或排除**	**资源**
管理 *	读	包含	/home/user/**
管理 *	读	排除	/home/user/private/**
系统生产	读	包含	/home/user/**
管理 *	读	包含	Application.reply.queue

注意:

1. 如果指定了 type=queue ， 那么资源为队列名称或 queue@qmgr。
2. 如果资源以 // 开头，那么该资源是数据集; 否则， 该资源是 z/OS UNIX 中的文件。
3. 用户标识是 MQMD 结构中的用户标识，因此这可能不会反映实际放置消息的用户标识。
4. 对于本地队列管理器上的请求，可以使用 MQADMIN CONTEXT.* 以限制哪些用户可以设置此值。
5. 对于通过远程队列管理器传入的请求，您必须假定分布式队列管理器已启用安全性，以防止在 MQMD 结构中对用户标识进行未经授权的设置。
6. Linux 机器上的 SYSPROG1 用户标识与 z/OS 上的安全性检查的用户标识 SYSPROG1 相同。

z/OS 计划在 z/OS 上使用 IBM MQ Console 和 REST API

IBM MQ Console 和 REST API 是在称为 mqweb 的 WebSphere Liberty (Liberty) 服务器中运行的应用程序。 mqweb 服务器作为启动式任务运行。 MQ Console 允许使用 Web 浏览器来管理队列管理器。 REST API 提供了一个简单的程序化接口，供应用程序执行队列管理器管理和执行消息传递。

安装和配置文件

您需要安装 IBM MQ for z/OS UNIX System Services Web Components 功能部件，该功能部件将安装在 z/OS UNIX System Services (z/OS UNIX) 中运行 mqweb 服务器所需的文件。 您需要熟悉 z/OS UNIX 才能配置和管理 mqweb 服务器。

z/OS UNIX 中的 IBM MQ 文件与正确操作 mqweb 服务器所需的各种属性集一起安装。 如果需要复制 IBM MQ z/OS UNIX 安装文件 (例如，如果已在一个系统上安装 IBM MQ ， 并在另一个系统上运行 IBM MQ)， 那么应复制在安装期间创建的 IBM MQ ZFS ， 并将其只读安装到目标。 以其他方式复制文件可能会导致某些文件属性丢失。

您需要在创建 mqweb 服务器时决定 Liberty 用户目录的位置并创建该用户目录。 此目录包含配置和日志文件，并且位置可以类似于 /var/mqm/mqweb。

将 MQ Console 和 REST API 与不同级别的队列管理器配合使用

MQ Console 和 REST API 只能直接与在同一版本，发行版和修订版 (VRM) 上运行的队列管理器进行交互。 例如， IBM MQ 9.1.0 随附的 MQ Console 和 REST API 只能与位于 IBM MQ 9.1.0 的本地队列管理器进行交互，而 IBM MQ 9.0.5 随附的 MQ Console 和 REST API 只能与位于 IBM MQ 9.0.5 的本地队列管理器进行交互。

对于 REST API，您可以通过配置网关队列管理器来管理与 mqweb 服务器版本不同的队列管理器。 但是，您需要至少一个与 mqweb 服务器版本相同的队列管理器来充当网关队列管理器。 有关更多信息，请参阅使用 REST API 进行远程管理。

迁移

如果只有一个队列管理器，那么可以将 mqweb 服务器作为单个启动式任务运行，并在迁移队列管理器时更改其使用的库。

如果您有多个队列管理器，那么在迁移期间，可以使用具有不同名称的启动式任务在不同版本上启动 mqweb 服务器。 这些名称可以是您想要的任何名称。 例如，可以使用名为 MQWB0910 的启动式任务启动 IBM MQ 9.1.0 mqweb 服务器，并使用名为 MQWB0905 的启动式任务启动 IBM MQ 9.0.5 mqweb 服务器。

然后，当您将队列管理器从一个版本迁移到更高版本时，队列管理器在 mqweb 服务器中对于更高版本可用，而在 mqweb 服务器中对于更低版本不再可用。

将所有队列管理器迁移到更高版本后，可以删除较低版本的 mqweb 服务器。

HTTP 端口

mqweb 服务器最多将两个端口用于 HTTP:

- 一个用于 HTTPS，缺省值为 9443。
- 一个用于 HTTP。 缺省情况下未启用 HTTP，但如果已启用 HTTP，那么缺省值为 9080。

如果正在使用缺省端口值，那么必须分配其他端口。 如果针对 IBM MQ 的多个版本同时运行了多个 mqweb 服务器，那么必须为每个版本分配单独的端口。 有关设置 mqweb 服务器使用的端口的更多信息，请参阅 配置 HTTP 和 HTTPS 端口。

可以使用以下 TSO 命令来显示有关端口的信息:

```
NETSTAT TCP tcpip (PORT portNumber)
```

其中 *tcpip* 是 TCP/IP 地址空间的名称， *portNumber* 指定要显示其相关信息的端口号。

安全性-启动 mqweb 服务器

mqweb 服务器用户标识需要特定权限。 有关更多信息，请参阅 mqweb 服务器启动式任务用户标识所需的权限。

安全性-使用 MQ Console 和 REST API

使用 MQ Console 和 REST API 时，必须以已配置注册表中包含的用户身份进行认证。 将为这些用户分配特定角色，以确定用户可以执行的操作。 例如，要使用 messaging REST API，必须为用户分配 `MQWebUser` 角色。 有关 MQ Console 和 REST API 的可用角色以及这些角色授予的访问权的更多信息，请参阅 MQ Console 和 REST API 上的角色。

有关为 MQ Console 和 REST API 配置安全性的更多信息，请参阅 MQ Console 和 REST API 安全性。

声明

本信息是为在美国提供的产品和服务编写的。

IBM 可能在其他国家或地区不提供本文档中讨论的产品、服务或功能。 有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。 任何对 IBM 产品、程序或服务的引用并非意在明示或默示只能使用 IBM 的产品、程序或服务。 只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务都可以代替 IBM 产品、程序或服务。 但是， 评估和验证任何非 IBM 产品、程序或服务的操作，由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。 提供本文档并未授予用户使用这些专利的任何许可。 您可以以书面形式将许可查询寄往：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

有关双字节（DBCS）信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

知识产权许可
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 063-8506 Japan

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区：International Business Machines Corporation "按现状" 提供本出版物，不附有任何种类的 (无论是明示的还是暗含的) 保证，包括但不限于暗含的有关非侵权，适销和适用于某种特定用途的保证。 某些国家或地区在某些交易中不允许免除明示或暗含的保证。 因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。 此处的信息将定期更改；这些更改将编入本资料的新版本中。 IBM 可以随时对本出版物中描述的产品和/或程序进行改进和/或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式 充当对那些 Web 站点的保证。 那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：(i) 允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及 (ii) 允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM Corporation
软件互操作性协调员，部门 49XA
北纬 3605 号公路
罗切斯特，明尼苏达州 55901
U.S.A.

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。 因此，在其他操作环境中获得的数据可能会有明显的不同。 有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。 此外，有些测量是通过推算而估计的， 实际结果可能会有差异。 本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中 获取。 IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。 有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本信息包含日常商业运作所使用的数据和报表的示例。 为了尽可能全面地说明这些数据和报表，这些示例包括个人、公司、品牌和产品的名称。 所有这些名称都是虚构的，如与实际商业企业所使用的名称和地址有任何雷同，纯属巧合。

版权许可：

本信息包含源语言形式的样本应用程序，用以阐明在不同操作平台上的编程技术。 如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。 这些示例并未在所有条件下作全面测试。 因此，IBM 不能担保或默示这些程序的可靠性、可维护性或功能。

如果您正在查看本信息的软拷贝，图片和彩色图例可能无法显示。

编程接口信息

编程接口信息 (如果提供) 旨在帮助您创建用于此程序的应用软件。

本书包含有关允许客户编写程序以获取 WebSphere MQ 服务的预期编程接口的信息。

但是，该信息还可能包含诊断、修改和调优信息。 提供诊断、修改和调优信息是为了帮助您调试您的应用程序软件。

要点: 请勿将此诊断，修改和调整信息用作编程接口，因为它可能会发生更改。

商标

IBM IBM 徽标 ibm.com 是 IBM Corporation 在全球许多管辖区域的商标。 当前的 IBM 商标列表可从 Web 上的 “Copyright and trademark information”www.ibm.com/legal/copytrade.shtml 获取。 其他产品和服务名称可能是 IBM 或其他公司的商标。

Microsoft 和 Windows 是 Microsoft Corporation 在美国和/或其他国家或地区的商标。

UNIX 是 Open Group 在美国和其他国家或地区的注册商标。

Linux 是 Linus Torvalds 在美国和/或其他国家或地区的商标。

此产品包含由 Eclipse 项目 (https://www.eclipse.org/) 开发的软件。

Java 和所有基于 Java 的商标和徽标是 Oracle 和/或其附属公司的商标或注册商标。

部件号:

(1P) P/N: