9.4

IBM MQ 계획

참고

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 191 페이지의 『주의사항』에 있는 정보를 확인하십시오.

이 개정판은 새 개정판에 별도로 명시하지 않는 한, IBM® MQ 의 버전 9릴리스 4및 모든 후속 릴리스와 수정에 적용됩니다.

IBM은 귀하가 IBM으로 보낸 정보를 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 사용하거나 배포할 수 있습니다.

목차

IBM MQ 아키텍처 계획

IBM MQ 환경을 계획할 때, IBM MQ가 단일 및 다중 큐 관리자 아키텍처와, 포인트-투-포인트 및 발행/구독 메시징 스타일에 대해 제공하는 지원을 고려하십시오. 또한 자원 요구사항과, 로깅 및 백업 기능 사용을 계획하십시오.

이 태스크 정보

IBM MQ 아키텍처를 계획하기 전에 기본 IBM MQ 개념을 숙지하십시오. IBM MQ 기술 개요를 참조하십시오.

IBM MQ 아키텍처 범위는 단일 큐 관리자를 사용하는 단순 아키텍처에서 상호 연결된 큐 관리자의 보다 복잡한 네트워크에 이릅니다. 다수의 큐 관리자는 분산 큐잉 기술을 사용하여 서로 연결됩니다. 단일 큐 관리자 및 다중 큐 관리자 아키텍처에 대한 자세한 정보는 다음 주제를 참조하십시오.

- 16 페이지의 『단일 큐 관리자에 기반한 아키텍처』
- 17 페이지의 『다중 큐 관리자에 기반한 아키텍처』
  - 18 페이지의 『분산 큐와 클러스터 계획』
  - 63 페이지의 『분산 발행/구독 네트워크 계획』

z/OS IBM MQ for z/OS®에서는 공유 큐와 큐 공유 그룹을 사용하여 워크로드 밸런싱을 구현하며 확장 가능하고 고가용적인 IBM MQ 애플리케이션을 사용할 수 있습니다. 공유 큐 및 큐 공유 그룹에 대한 정보는 공유 큐 및 큐 공유 그룹을 참조하십시오.

IBM MQ에서는 두 개의 다른 릴리스 모델을 제공합니다.

- Long Term Support (LTS) 릴리스는 장기 배치 및 최대 안정성이 필요한 시스템에 가장 적합합니다.
- Continuous Delivery (CD) 릴리스는 IBM MQ의 최신 기능 개선사항을 신속하게 이용해야 하는 시스템을 위한 것입니다.

두 릴리스 유형은 동일한 방법으로 설치되지만, 사용자가 알아야 하는 지원 및 마이그레이션에 관한 고려사항이 있습니다. 자세한 정보는 IBM MQ 릴리스 유형 및 버전화를 참조하십시오.

다중 설치, 스토리지 및 성능 요구사항, 클라이언트 사용 계획에 대한 정보는 다른 하위 주제를 참조하십시오.

관련 개념

IBM MQ 릴리스 유형 및 버전화

128 페이지의 『Planning your IBM MQ environment on z/OS』
When planning your IBM MQ environment, you must consider the resource requirements for data sets, page sets, Db2, Coupling Facilities, and the need for logging, and backup facilities. Use this topic to plan the environment where IBM MQ runs.

가용성, 복구 및 재시작

관련 태스크

요구사항 검사

메시지가 유실되지 않았는지 확인(로깅)

IBM MQ 릴리스 유형: 계획 고려사항

IBM MQ 의 두 가지 기본 릴리스 유형은 Long Term Support (LTS) 및 Continuous Delivery (CD) 입니다. 지원되는 각 플랫폼에 대해 사용자가 선택하는 릴리스 유형은 주문, 설치, 유지보수 및 마이그레이션에 영향을 줍니다.

릴리스 유형에 대한 자세한 정보는 IBM MQ 릴리스 유형 및 버전화를 참조하십시오.

IBM MQ for Multiplatforms에 대한 고려사항

Multi

주문

Passport Advantage® 내에는 IBM MQ 9.4에 대한 두 개의 별도 eAssemblies 가 있습니다. 하나는 IBM MQ 9.4.0 Long Term Support 릴리스에 대한 설치 이미지를 포함하고 다른 하나는 IBM MQ 9.4.x Continuous Delivery 릴리스에 대한 설치 이미지를 포함합니다. 선택한 릴리스에 따라 eAssembly에서 설치 이미지를 다운로드합니다.

모든 IBM MQ 버전(IBM MQ 9.4의 경우 LTS 릴리스와 CD 릴리스 둘 다)이 동일한 제품 ID에 속합니다.

IBM MQ 를 사용할 수 있는 인타이틀먼트는 라이센스가 부여된 구성요소 및 가격 책정 메트릭의 제한조건에 따라 전체 제품 (PID) 에서 확장됩니다. 이는 IBM MQ 9.4에 대한 LTS 릴리스와 CD 릴리스 설치 이미지 사이에서 자유롭게 선택할 수 있음을 의미합니다.

설치

Passport Advantage에서 설치 이미지를 다운로드한 후에는 인타이틀먼트를 구매한 구성요소만 설치하도록 선택해야 합니다. 각 유료 컴포넌트에 포함된 설치 가능한 컴포넌트에 대한 추가 정보는 IBM MQ 라이센스 정보를 참조하십시오.

동일한 운영 체제 이미지에 IBM MQ 9.4.0 LTS 릴리스 및 IBM MQ 9.4.x CD 릴리스를 설치할 수 있습니다. 이를 수행하는 경우 컴포넌트는 IBM MQ 다중 버전 지원에서 지원하는 대로, 별도의 설치로 나타납니다. 각 버전에는 해당 버전과 연관된 차별되는 큐 관리자 세트가 있습니다.

각각의 새 CD 릴리스는 하나의 설치 이미지로 제공됩니다. 새 CD 릴리스를 기존 릴리스와 함께 설치하거나 설치 프로그램이 이전 CD 릴리스를 새 릴리스로 업데이트할 수 있습니다.

CD 릴리스에는 최신 결함 수정사항 및 보안 업데이트 세트와 함께 기능 개선사항이 포함되어 있습니다. 각 CD 릴리스는 누적되며 IBM MQ의 해당 버전에 대한 모든 이전 릴리스를 완전히 대체합니다. 따라서 엔터프라이즈와 관련된 기능이 포함되어 있지 않은 경우 특정 CD 릴리스를 건너뛸 수 있습니다.

유지보수

LTS 릴리스는 결함 수정사항을 제공하는 수정팩 및 보안 패치를 제공하는 누적 보안 업데이트 (CSU) 를 적용하여 서비스를 제공합니다. 수정팩 및 CSU는 주기적으로 사용 가능하며 누적됩니다.

CD의 경우 후속 버전에 있을 수 있는 최신 CD 릴리스에 대해서만 CSU가 생성됩니다.

IBM 지원 팀에서 임시 수정사항을 적용하도록 지시하는 경우가 있습니다. 임시 수정사항은 응급 또는 테스트 수정사항이라고도 하며 다음 유지보수 전달을 대기할 수 없는 긴급 업데이트를 적용하는 데 사용됩니다.

LTS 릴리스와 CD 릴리스 사이의 마이그레이션

제한조건과 제한사항이 있지만 대상 릴리스가 마이그레이션 전에 사용 중인 것보다 높은 릴리스인 경우 일반적으로 LTS 릴리스 코드 사용에서 CD 릴리스 코드 사용으로 또는 CD 릴리스 코드 사용에서 LTS 릴리스 코드 사용으로 단일 큐 관리자를 마이그레이션할 수 있습니다.

두 가지 접근 방식이 가능합니다.

- IBM MQ의 기존 설치를 업데이트하도록 코드의 새 릴리스를 대신 설치하십시오. 설치와 연관된 큐 관리자는 시작 시 코드의 새 릴리스를 사용합니다.
- 코드의 새 릴리스를 새 설치로 설치한 후, setmqm 명령을 사용하여 새 설치로 개별 큐 관리자 인스턴스를 이동합니다.

큐 관리자가 코드의 CD 릴리스 실행을 시작하면 새 릴리스 레벨을 표시하도록 큐 관리자 명령 레벨이 업데이트됩니다. 이는 릴리스에서 제공되는 새 기능이 사용 가능하며 더 이상 더 낮은 VRM 번호의 코드 릴리스를 사용하여 큐 관리자를 재시작할 수 없음을 의미합니다.

IBM MQ for z/OS에 대한 고려사항

z/OS

주문

IBM MQ for z/OS 9.4을 주문할 때 ShopZ에는 별도의 두 기능이 제공됩니다. 이 기능은 LTS 릴리스 및 CD 릴리스에 해당합니다. 두 기능은 모두 동일한 제품 ID(PID)에 적용 가능합니다. 이 ID는 라이센스가 있는 제품 ID이므로, 한 기능에 라이센스가 있으면, 필요한 경우 대체 기능을 사용할 자격(인타이틀먼트)이 있습니다. 주문할 때 LTS 릴리스 또는 CD 릴리스에 해당하는 기능을 선택합니다.

ServerPac에 포함할 제품을 선택하는 경우, 동일한 ServerPac 주문에서 LTS 릴리스와 CD 릴리스를 둘 다 선택할 수 없습니다. 동일한 대상 영역에서 SMP/E에 의해 제품을 설치할 수 없기 때문입니다.

설치

LTS 및 CD 릴리스는 별도의 FMID 세트에 제공됩니다. 이러한 FMID는 동일한 SMP/E 대상 구역에 설치할 수 없습니다. LTS 및 CD 릴리스가 모두 필요한 경우 다음을 수행하십시오.

- LTS 릴리스 및 CD 릴리스를 별도의 대상 구역에 설치하십시오.
- 두 릴리스에 대해 별도의 대상 및 분배 라이브러리를 유지보수하십시오.

큐 관리자가 큐 공유 그룹에 있는 경우 최신 CD 버전으로 업그레이드할 때 그룹의 모든 큐 관리자를 업그레이드해야 합니다.

큐 관리자의 명령 레벨은 3자리 VRM 레벨입니다. 안IBM MQ 프로그램은 호출할 수 있습니다`MQINQ` , 전달 `MQIA_COMMAND_LEVEL` 선택기, 연결된 큐 관리자의 명령 레벨을 가져옵니다.

릴리스가 다른 FMID를 사용하기 때문에 LTS 릴리스에 대한 유지보수 또는 다른 방식으로 CD 릴리스를 업데이트할 수 없습니다. 마찬가지로, 제품 코드의 버전을 LTS 릴리스에서 CD 릴리스 또는 그 반대로 전환할 수 있는 방법이 없습니다. 그러나 릴리스 모델 간에 큐 관리자를 전환할 수 있습니다. LTS 릴리스와 CD 릴리스 간의 마이그레이션을 참조하십시오.

참고:

IBM MQ 9.0.x 및 IBM MQ 9.1.x CD 릴리스에는 서로 다른 버전 및 릴리스 종속 FMID가 있습니다. 따라서 9.0.x CD 에서 9.1.x CD 로 이동하려면 최소한 하나의 완전한 SMP/E 설치가 필요합니다.

IBM MQ for z/OS 9.2.0부터 CD 릴리스는 버전 번호가 9인 모든 IBM MQ for z/OS 릴리스에 대해 동일하게 유지되는 FMID 세트를 사용합니다. IBM MQ 의 각 새 버전이 CD 및 LTS 릴리스 둘 다로 사용 가능하므로, 주 버전 경계를 넘는 경우에도 PTF를 단일 SMP/E 설치에 적용하여 CD 릴리스를 업그레이드할 수 있습니다. 예를 들어, PTF를 적용하여 IBM MQ for z/OS 9.2.0 CD에서 IBM MQ for z/OS 9.2.2 CD, IBM MQ for z/OS 9.2.4 CD, IBM MQ for z/OS 9.3.0 CD로 이동할 수 있습니다.

큐 관리자 작업 로그에서 CSQY000I 메시지를 보고 동일한 VRM 레벨의 LTS 및 CD 릴리스를 구별할 수 있습니다.

유지보수

IBM MQ for z/OS 는 유지보수를 위해 PTF를 사용합니다.

PTF는 특정 릴리스 레벨에 대응되는 특정 라이브러리 세트에 한정됩니다. UNIX System Services 기능 (즉, JMS 및 WEB UI, 커넥터 팩 및 Managed File Transfer) 의 경우 z/OS PTF는 멀티플랫폼 수정팩 및 누적 보안 업데이트 (CSU) 에 직접 맞춰집니다. 이 수정사항은 누적되며 동등한 멀티플랫폼 수정팩 또는 CSU와 동시에 사용할 수 있습니다.

CD CSU는 일반적으로 CD 릴리스 간에 사용 가능하지 않지만 다음 IBM MQ for z/OS CD 릴리스에 포함되어 있습니다. ++USERMOD를 요청하기 위해 지원 센터에 문의할 수도 있습니다.

IBM MQ for z/OS 의 기타 수정사항은 특정 파트의 고유 수정사항입니다. 이러한 수정사항은 특정 문제를 해결하고 누적되지 않으며 생성될 때 사용 가능하게 됩니다.

LTS 릴리스와 CD 릴리스 사이의 마이그레이션

제한조건 및 제한사항이 있지만 대상 릴리스가 마이그레이션 전에 사용 중인 것보다 높은 릴리스인 경우 일반적으로 LTS 릴리스 코드 사용에서 CD 릴리스 코드 사용으로 또는 CD 릴리스 코드 사용에서 LTS 릴리스 코드 사용으로 단일 큐 관리자를 마이그레이션할 수 있습니다.

IBM MQ for z/OS 9.2.0부터 역방향 마이그레이션 기능에 영향을 주지 않고 필요한 횟수만큼 동일한 VRM을 사용하여 CD 및 LTS 릴리스 간에 앞뒤로 마이그레이션할 수 있습니다. 예를 들어, IBM MQ for z/OS 9.3.0 LTS에서 큐 관리자를 시작한 후 IBM MQ for z/OS 9.3.0 CD에서 종료 및 시작하고 IBM MQ for z/OS 9.3.0 LTS에서 종료 및 시작할 수 있습니다.

IBM MQ for z/OS 에서는 일반적으로 마이그레이션 후 실행 기간이 지나면 이전 릴리스로 폴백할 수 있도록 폴백 기능 (역방향 마이그레이션) 을 제공합니다. 이 기능은 LTS 릴리스 및 수정자가 `0` 인 CD 릴리스 (예: 9.3.0 CD) 에 대해 유지되지만, 마이그레이션의 소스 또는 대상이 수정자 번호가 0이 아닌 CD 릴리스 (예: 9.2.5 또는 9.3.1) 인 경우에는 가능하지 않습니다.

다음은 올바른 마이그레이션 시나리오이며 이 원칙이 작동하는 방식을 보여줍니다.

소스 릴리스	대상 릴리스	참고
9.1.0 LTS	9.4.0 LTS 또는 9.4.0 CD	9.1.0 LTS가 표준 지원을 벗어나므로 역방향 마이그레이션이 지원되지 않습니다.
9.2.0 LTS	9.4.0 LTS 또는 9.4.0 CD	역방향 마이그레이션이 지원됩니다.
9.3.0 LTS	9.4.0 LTS 또는 9.4.0 CD	역방향 마이그레이션이 지원됩니다.
9.3.5 CD	9.4.0 LTS 또는 9.4.0 CD	소스 릴리스가 수정자가 0이 아닌 CD이므로 역방향 마이그레이션이 지원되지 않습니다.
9.4.0 LTS 또는 9.4.0 CD	9.4.1 CD	대상 릴리스가 수정자가 0이 아닌 CD이므로 역방향 마이그레이션이 지원되지 않습니다. 이주를 확인하기 위해 `Write to operator with reply` CSQY041D 가 발행됩니다.

관련 태스크

z/OS z/OS에서 유지보수 적용 및 제거

관련 정보

다운로드 중IBM MQ 9.4

GDPR 대비를 위한 IBM MQ 및 IBM MQ Appliance 온프레미스 고려사항

관련 PID:

분산

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

z/OS

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

IBM MQ Appliance

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

주의사항:

이는 귀사의 GDPR 대비를 준비하도록 도와주기 위한 것입니다. 이는 GDPR 준비로 조직을 지원하기 위해 고려해야 하는 제품 사용의 측면과 구성할 수 있는 IBM MQ의 기능에 대한 정보를 제공합니다. 고객이 기능을 선택하고 구성할 수 있는 여러 방법이 있고 본 제품을 그 자체로 그리고 제3자 애플리케이션 및 시스템과 병행하여 사용할 수 있는 다양한 방식이 있으므로 이 정보는 완전한 목록은 아닙니다.

고객은 유럽 연합의 GDPR을 포함한 다양한 법령과 규정을 준수해야 할 책임이 있습니다. 고객은 고객의 비즈니스에 영향을 줄 수 있는 관련 법령 및 규정에 대한 확인과 해석, 그러한 법령 및 규정의 준수를 위해 필요한 고객의 모든 조치와 관련하여 적절한 법률 자문을 받아야 할 단독 책임이 있습니다.

본 컨텐츠에 설명된 제품, 서비스 및 기타 기능은 일부 고객의 상황에 적합하지 않거나 사용 가능성이 제한될 수 있습니다. IBM은 법률, 회계 또는 감사 관련 자문을 제공하지 않으며, IBM의 제품이나 서비스가 클라이언트의 관련 법령 및 규정을 준수한다는 진술이나 보증을 제공하지 않습니다.

목차

GDPR

일반 개인정보 보호법률(General Data Protection Regulation, "GDPR")은 유럽 연합("EU")에 의해 채택되어 2018년 5월 25일부터 적용됩니다.

GDPR이 중요한 이유

GDPR은 개인정보 처리를 위해 보다 엄격한 정보 보호 규제 체계를 규정하고 있습니다. GDPR 규정 사항은 다음과 같습니다.

- 개인을 위한 새롭고 강화된 권리
- 개인정보의 정의 확장
- 처리자에 대한 새로운 의무
- 미준수에 대한 상당한 금전적 제재 가능성
- 데이터 유출(breach)에 대한 의무적 고지

GDPR에 대해 자세히 알아보기:

- EU GDPR 정보 포털
- ibm.com/GDPR 웹 사이트

제품 구성 - GDPR 대비를 위한 고려사항

다음 절에서는 귀사의 조직이 GDPR에 대비하는 데 필요한 IBM MQ 구성 고려사항을 제공합니다.

데이터 수명 주기

IBM MQ는 애플리케이션이 비동기식으로 애플리케이션 제공 데이터를 교환할 수 있게 하는 트랜잭션 메시지 지향 미들웨어 제품입니다. IBM MQ는 애플리케이션 연결을 위한 다양한 메시징 API, 프로토콜 및 브릿지를 지원합니다. 이와 같이 IBM MQ를 사용하여 많은 양식의 데이터를 교환할 수 있으며, 일부 데이터는 잠재적으로 GDPR의 영향을 받을 수 있습니다. IBM MQ가 데이터를 교환할 수 있는 몇 가지 써드파티 제품이 있습니다. 이 중 일부는 IBM 소유이며 나머지 많은 제품들은 다른 기술 공급자에 의해 제공됩니다. 소프트웨어 제품 호환성 보고서 웹 사이트에서는 연관된 소프트웨어의 목록을 제공합니다. 써드파티 제품의 GDPR 대비 관련 고려사항은 제

품 문서를 참조해야 합니다. IBM MQ 관리자는 큐, 토픽 및 구독에 대한 정의로 IBM MQ가 전달되는 데이터와 상호 작용하는 방식을 제어합니다.

IBM MQ를 통한 데이터 플로우의 유형은 무엇입니까?

IBM MQ가 애플리케이션 데이터에 대한 비동기 메시징 서비스를 제공할 때 유스 케이스가 애플리케이션 배치에 따라 다르므로 이 질문에 대한 하나의 명확한 답변은 없습니다. 애플리케이션 메시지 데이터는 큐 파일(페이지 세트 또는 z/OS의 커플링 기능)에 보관되며 로그와 아카이브 및 메시지 자체에 GDPR로 통제되는 데이터가 포함될 수 있습니다. 애플리케이션 제공 메시지 데이터도 오류 로그, 추적 파일 및 FFST와 같이 문제점 판별 용도로 수집된 파일에 포함될 수 있습니다. 또한 z/OS에서 애플리케이션 제공 메시지 데이터가 주소 공간 또는 커플링 기능 덤프에 포함될 수 있습니다.

다음은 IBM MQ를 사용하여 교환할 수 있는 개인 데이터의 일반적인 몇 가지 예입니다.

- 고객의 직원(예를 들어, IBM MQ를 사용하여 고객의 급여 대장 또는 HR 시스템을 연결할 수 있음)
- 고객이 소유한 클라이언트의 개인 데이터(예를 들어, 고객이 IBM MQ를 사용하여 영업 리드를 수행하고 해당 CRM 시스템에 데이터를 저장하는 등 고객의 클라이언트와 관련된 애플리케이션 간에 데이터를 교환할 수 있음)
- 고객이 소유한 클라이언트의 민감한 개인 데이터(예를 들어, IBM MQ를 클라이언트 애플리케이션 통합 시 HL7 기반 의료 서비스 레코드 등 개인 데이터 교환이 필요한 업계 컨텍스트 내에서 사용할 수 있음)

IBM MQ는 애플리케이션 제공 메시지 데이터 외에 다음 유형의 데이터를 처리합니다.

- 인증 신임 정보(예: 사용자 이름 및 비밀번호, API 키 등)
- 일반적으로 식별 가능한 개인 정보(예: 디바이스 ID, 사용 기반 ID, IP 주소 등 - 개인으로 링크될 때)

IBM과의 온라인 문의에 사용되는 개인용 데이터

IBM MQ 고객은 온라인 의견/피드백/요청을 제출하여 주로 다음과 같은 다양한 방법으로 IBM MQ 주제에 대해 IBM 에 문의할 수 있습니다.

- IBM Developer의 IBM MQ 영역에 있는 페이지의 공개 댓글 영역
- IBM Documentation의 IBM MQ 제품 정보 페이지에 있는 공개 댓글 영역
- IBM 지원 센터 포럼의 공개 댓글
- IBM Integration Ideas의 공개 댓글

일반적으로 사용자 문의사항에 대한 개별적인 답변에는 고객 이름 및 이메일 주소만 사용되며, 이러한 개인정보 사용은 IBM 온라인 개인정보처리방침을 준수합니다.

데이터 수집

IBM MQ를 사용하여 개인 데이터를 수집할 수 있습니다. IBM MQ의 사용자 사용 및 요구사항을 평가하여 GDPR의 요구를 충족시키는 경우 사용자 상황에서 IBM MQ를 통해 전달되는 개인 데이터 유형을 고려해야 합니다. 다음과 같은 측면을 고려할 수 있습니다.

- 데이터가 어떤 방식으로 큐 관리자에 도달합니까? (어떤 프로토콜을 거칩니까? 데이터가 암호화되어 있습니까? 데이터가 서명되어 있습니까?)
- 큐 관리자에서 데이터를 송신하는 방법은 무엇입니까? (어떤 프로토콜을 거칩니까? 데이터가 암호화되어 있습니까? 데이터가 서명되어 있습니까?)
- 데이터가 큐 관리자를 통해 전달될 때 어떻게 저장됩니까? (메시지가 비지속적인 경우에도 모든 메시징 애플리케이션은 Stateful 매체에 메시지 데이터를 기록할 수 있습니다. 메시징 기능이 제품을 통해 전달되는 애플리케이션 메시지 데이터의 측면을 잠재적으로 노출하는 방식을 알고 있습니까?)
- 써드파티 애플리케이션에 액세스하기 위해 IBM MQ에 필요한 경우 신임 정보를 수집하고 저장하는 방법은 무엇입니까?

IBM MQ는 LDAP 등 인증이 필요한 다른 시스템과 서비스와 통신해야 합니다. 필요에 따라 이와 같은 통신에 사용하기 위해 IBM MQ에서 인증 데이터(사용자 ID, 비밀번호)를 구성하고 저장합니다. 가능하면 IBM MQ 인증에 개인 신임 정보를 사용하지 않는 것이 좋습니다. 인증 데이터에 사용되는 스토리지 보호를 고려하십시오. (다음 데이터 스토리지를 참조하십시오.)

데이터 스토리지

메시지 데이터가 큐 관리자를 통해 이동하는 경우 IBM MQ는 Stateful 매체에 직접 이 데이터(아마도 여러 개의 사본)를 보관합니다. IBM MQ 사용자는 저장된 메시지 데이터 보안을 고려할 수 있습니다.

다음 항목은 IBM MQ가 사용자가 GDPR 준수를 보장할 때 고려할 애플리케이션 제공 데이터를 보관하는 영역을 강조표시합니다.

- 애플리케이션 메시지 큐:

  IBM MQ는 애플리케이션 간 비동기 데이터 교환을 허용하는 메시지 큐를 제공합니다. 큐에 저장된 비지속적 메시지 및 지속적 메시지는 stateful 매체에 기록됩니다.
- 파일 전송 에이전트 큐:

  IBM MQ Managed File Transfer는 메시지 큐를 활용하여 신뢰할 수 있는 파일 데이터 전송, 개인 데이터가 포함된 파일 및 이러한 큐에 저장된 전송 기록을 통합합니다.
- 전송 큐:

  큐 관리자 간에 안전하게 메시지를 전송하기 위해 메시지를 전송 큐에 임시 저장합니다.
- 데드 레터 큐:

  큐가 큐 관리자에 구성된 경우, 메시지를 목적지 큐에 넣을 수 없어 데드-레터 큐에 저장되는 일부 상황이 있습니다.
- 백아웃 큐:

  JMS 및 XMS 메시징 인터페이스는 다른 올바른 메시지를 처리할 수 있도록 많은 백아웃이 발생한 후 포이즌 메시지를 백아웃 큐로 이동할 수 있게 하는 기능을 제공합니다.
- AMS 오류 큐:

  IBM MQ Advanced Message Security 는 보안 정책을 준수하지 않는 메시지를 SYSTEM.PROTECTION.ERROR.QUEUE 오류 큐.
- 보유된 발행:

  IBM MQ는 구독 애플리케이션이 이전 발행을 재호출할 수 있도록 하는 보유된 발행 기능을 제공합니다.
- 지연된 전달:

  IBM MQ 는 메시지가 나중에 대상으로 전달될 수 있도록 하는 JMS 2.0 및 Jakarta Messaging 3.0 전달 지연 기능을 지원합니다. 아직 전달되지 않은 메시지는 SYSTEM.DDELAY.LOCAL.QUEUE 큐에 저장됩니다.

자세한 내용은 다음을 참조하십시오.

- 로깅: 메시지가 유실되지 않았는지 확인
- MFT 에이전트 큐 설정
- 데드-레터 큐 사용
- JMS용 IBM MQ 클래스에서 포이즌 메시지 처리
- AMS 오류 처리
- 보유된 발행물
- JMS 2.0 전달 지연

다음 항목은 IBM MQ가 사용자가 GDPR 준수를 보장할 때 고려할 애플리케이션 제공 데이터를 간접적으로 보관합니다.

- 추적 라우트 메시징:

  IBM MQ는 애플리케이션 사이에서 메시지가 이동하는 라우트를 기록하는 추적 라우트 기능을 제공합니다. 생성된 이벤트 메시지에는 IP 주소와 같은 기술적으로 식별 가능한 개인 정보가 포함될 수 있습니다.
- 애플리케이션 활동 추적:

  IBM MQ는 애플리케이션 및 채널의 메시징 API 활동을 기록하는 애플리케이션 활동 추적을 제공하며 애플리케이션 활동 추적은 이벤트 메시지에 애플리케이션 제공 메시지 데이터의 컨텐츠를 기록할 수 있습니다.
- 서비스 추적:

IBM MQ는 메시지 데이터가 플로우되는 내부 코드 경로를 기록하는 서비스 추적 기능을 제공합니다. 이러한 기능의 부분으로서 IBM MQ는 디스크에 저장된 추적 파일에 애플리케이션 제공 메시지 데이터의 컨텐츠를 기록할 수 있습니다.

- 큐 관리자 이벤트:

  IBM MQ는 권한, 명령 및 구성 이벤트와 같은 개인 데이터를 포함할 수 있는 이벤트 메시지를 생성할 수 있습니다.

자세한 내용은 다음을 참조하십시오.

- 라우트 추적 메시징
- 추적 사용
- 이벤트 모니터링
- 큐 관리자 이벤트

애플리케이션 제공 메시지 데이터의 사본에 대한 액세스를 보호하려면 다음 조치를 고려하십시오.

- 파일 시스템에서 IBM MQ 데이터에 대한 권한이 있는 사용자 액세스를 제한하십시오(예: UNIX and Linux® 플랫폼에서 'mpm' 그룹의 사용자 멤버십 제한).
- 전용 큐 및 액세스 제어를 통해 IBM MQ 데이터에 대한 애플리케이션 액세스를 제한하십시오. 애플리케이션 간 큐와 같은 자원의 불필요한 공유를 피하고 큐 및 토픽 자원에 대한 세부 단위의 액세스 제어를 제공하십시오.
- 고가용성(HA) 또는 재해 복구(DR) 구성에서는 IBM MQ 데이터의 복제된 사본에 대한 액세스를 제한하고, 복제에 사용되는 연결을 보호하십시오.
- IBM MQ Advanced Message Security를 사용하여 엔드-투-엔드 서명 및/또는 메시지 데이터의 암호화를 제공하십시오.
- 파일 또는 볼륨 레벨 암호화를 사용하여 IBM MQ 데이터, 추적 또는 로그를 포함할 수 있는 디렉토리 또는 파일 시스템을 보호하십시오.
- 서비스 추적을 IBM으로 업로드한 후 잠재적으로 개인 데이터가 포함된 컨텐츠에 대해 걱정이 되는 경우 서비스 추적 파일 및 FFST 데이터를 삭제할 수 있습니다.

자세한 내용은 다음을 참조하십시오.

- 권한이 있는 사용자
- 멀티플랫폼에서 파일 시스템 지원 계획
- IBM MQ Appliance에서 파일 시스템 암호화

IBM MQ 관리자는 LDAP, Salesforce와 같은 써드파티 서비스의 신임 정보(사용자 이름 및 비밀번호, API 키 등)를 LDAP과 같은 써드파티 서비스의 경우. 이 데이터는 일반적으로 파일 시스템 권한을 통해 보호되는 큐 관리자 데이터 디렉토리에 저장됩니다.

IBM MQ 큐 관리자가 작성되면 데이터 디렉토리가 그룹 기반 액세스 제어로 설정되어 IBM MQ가 구성 파일을 읽고 신임 정보를 사용하여 이러한 시스템에 연결할 수 있습니다. IBM MQ 관리자는 권한이 있는 사용자로 간주되며 이 그룹의 구성원이므로 파일에 대한 읽기 액세스 권한을 갖습니다. 일부 파일은 변조되지만 암호화되지는 않습니다. 이러한 이유로 신임 정보에 대한 액세스를 완전히 보호하려면 다음 조치를 고려해야 합니다.

- IBM MQ 데이터에 대한 권한이 있는 사용자 액세스를 제한하십시오(예: UNIX and Linux 플랫폼에서 'mpm' 그룹의 멤버십 제한).
- 파일 레벨 또는 볼륨 레벨 암호화를 사용하여 큐 관리자 데이터 디렉토리의 컨텐츠를 보호하십시오.
- 프로덕션 구성 디렉토리의 백업을 암호화하고 적절한 액세스 제어를 사용하여 이를 저장하십시오.
- 인증 실패에 대한 감사 추적, 액세스 제어 및 보안에 대한 구성 변경, 명령 및 구성 이벤트 제공을 고려하십시오.

자세한 내용은 다음을 참조하십시오.

- IBM MQ 보안

데이터 액세스

다음 제품 인터페이스를 통해 IBM MQ 큐 관리자 데이터에 액세스할 수 있습니다. 이 제품 인터페이스 중 일부는 원격 연결을 통해 액세스하도록 설계되었고 나머지는 로컬 연결을 통해 액세스하도록 설계되었습니다.

- IBM MQ Console[원격에만 해당]
- IBM MQ 관리 REST API[원격에만 해당]
- IBM MQ 메시징 REST API[원격에만 해당]
- MQI[로컬 및 원격]
- JMS[로컬 및 원격]
- XMS[로컬 및 원격]
- IBM MQ Telemetry(MQTT)[원격에만 해당]
- IBM MQ Light(AMQP)[원격에만 해당]
- IBM MQ IMS 브릿지[로컬에만 해당]
- IBM MQ CICS 브릿지[로컬에만 해당]
- IBM MQ MFT 프로토콜 브릿지[원격에만 해당]
- IBM MQ Connect:Direct 브릿지[원격에만 해당]
- IBM MQ MQAI[로컬 및 원격]
- IBM MQ PCF 명령[로컬 및 원격]
- IBM MQ MQSC 명령[로컬 및 원격]
- IBM MQ Explorer[로컬 및 원격]
- IBM MQ 사용자 엑시트[로컬에만 해당]
- IBM MQ Internet Pass-Thru[원격에만 해당]
- Red Hat® OpenShift® Monitoring(Prometheus) 메트릭(메트릭은 큐 관리자 통계에 대한 숫자 데이터임)
- IBM MQ Appliance 직렬 콘솔[로컬에만 해당]
- IBM MQ Appliance SSH[원격에만 해당]
- IBM MQ Appliance REST API[원격에만 해당]
- IBM MQ Appliance 웹 UI[원격에만 해당]
- V 9.4.0 IBM MQ Kafka 커넥터 (Kafka Connect) [로컬 및 원격]

인터페이스는 사용자가 IBM MQ 큐 관리자 및 여기에 저장된 메시지를 변경할 수 있도록 설계되었습니다. 관리 및 메시징 조작은 요청이 이루어질 때 세 단계를 거치도록 보안됩니다.

- 인증
- 역할 맵핑
- 권한 부여

인증:

메시지 또는 관리 조작이 로컬 연결에서 요청되면 이 연결의 소스가 동일한 시스템에서 프로세스를 실행합니다. 프로세스를 실행하는 사용자는 운영 체제에서 제공하는 인증 단계를 거쳐야 합니다. 연결이 이루어지는 프로세스 소유자의 사용자 이름은 ID로 확인됩니다. 예를 들어, 이는 애플리케이션이 시작된 쉘을 실행하는 사용자의 이름이 될 수 있습니다. 로컬 연결에 대해 가능한 인증 양식은 다음과 같습니다.

1. 확인된 사용자 이름(로컬 OS)
2. 선택적 사용자 이름 및 비밀번호(OS, LDAP 또는 사용자 정의 써드파티 저장소)
3. 보안 토큰 (JWT) IBM MQ 전용

관리 조치가 원격 연결에서 요청된 경우 IBM MQ와의 통신이 네트워크 인터페이스를 통해 이루어집니다. 다음 ID 양식은 네트워크 연결을 통해 인증에 제공될 수 있습니다.

1. 확인된 사용자 이름(원격 OS에서)

2. 사용자 이름 및 비밀번호(OS, LDAP 또는 사용자 정의 써드파티 저장소)
3. 소스 네트워크 주소(예: IP 주소)
4. X.509 디지털 인증서(상호 SSL/TLS 인증)
5. 보안 토큰 (예: LTPA2 토큰 또는 JWT 토큰)
6. 기타 사용자 정의 보안(써드파티 엑시트로 제공되는 기능)
7. SSH 키

IBM MQ와 IBM Cloud Pak® for Integration 의 통합은 Cloud Pak을 사용하는 IBM MQ Console: 싱글 사인온에 대한 새 인증 유형을 추가합니다. (CP4I만 해당)

역할 맵핑:

역할 맵핑 단계에서 인증 단계에 제공된 신임 정보는 대체 사용자 ID에 맵핑될 수 있습니다. 맵핑된 사용자 ID가 진행하도록 허용된 경우(예: 채널 인증 규칙으로 관리 사용자가 차단될 수 있음), 맵핑된 사용자 ID는 IBM MQ 자원에 대한 활동 권한을 부여할 때 최종 단계로 전달됩니다.

권한 부여:

IBM MQ에서는 여러 사용자가 큐, 토픽 및 기타 큐 관리자 오브젝트와 같은 다양한 메시징 자원에 대해 서로 다른 권한을 가질 수 있습니다.

로그 기록 활동:

IBM MQ의 일부 사용자는 MQ 자원에 대한 액세스의 감사 레코드를 작성해야 합니다. 원하는 감사 로그 예에는 요청한 사용자 외에 변경사항에 대한 정보가 포함된 구성 변경사항이 포함될 수 있습니다.

다음 정보 소스를 사용하여 이 요구사항을 구현할 수 있습니다.

1. IBM MQ 큐 관리자는 관리 명령이 성공적으로 실행되었을 때 명령 이벤트를 생성하도록 구성될 수 있습니다.
2. IBM MQ 큐 관리자는 큐 관리자 자원이 작성되거나 대체되거나 삭제되었을 때 구성 이벤트를 생성하도록 구성될 수 있습니다.
3. IBM MQ 큐 관리자는 자원에 대한 권한 검사가 실패한 경우에도 권한을 생성하도록 구성될 수 있습니다.
4. 실패한 권한 검사를 표시하는 오류 메시지는 큐 관리자 오류 로그에 기록됩니다.
5. IBM MQ Console은 인증, 권한 검사가 실패하거나 큐 관리자가 작성되거나 시작되거나 중지되거나 삭제될 때 해당 로그에 감사 메시지를 기록합니다.
6. IBM MQ Appliance는 자신의 로그에 감사 메시지를 기록하여 사용자 로그인 및 시스템 변경사항을 기록합니다.

이러한 유형의 솔루션을 고려할 때 IBM MQ 사용자는 다음과 같은 점을 고려할 수 있습니다.

- 이벤트 메시지는 비지속적이므로 큐 관리자가 재시작될 때 정보는 손실됩니다. 이벤트 모니터는 사용 가능한 메시지를 지속적으로 이용하고 컨텐츠를 지속적 매체로 전송할 수 있도록 구성해야 합니다.
- IBM MQ의 권한이 있는 사용자에게는 이벤트를 사용 안함으로 설정하거나 로그를 지우거나 큐 관리자를 삭제할 수 있는 충분한 권한이 있습니다.

IBM MQ 데이터 액세스 보안 및 감사 추적 제공에 대한 자세한 정보는 다음 주제를 참조하십시오.

- IBM MQ 보안 메커니즘
- 구성 이벤트
- 명령 이벤트
- 오류 로그 사용

데이터 처리

공개 키 인프라를 사용하는 암호화:

연결이 TLS를 사용하도록 지정하여 연결의 시작 측에 대한 상호 인증을 제공할 수도 있으므로 IBM MQ에 대한 네트워크 연결을 보안 설정할 수 있습니다.

전송 메커니즘에서 제공하는 PKI 보안 기능 사용은 IBM MQ를 사용한 데이터 처리 보안을 위한 첫 번째 단계입니다. 하지만 추가 보안 기능을 사용하지 않으면 처리 애플리케이션이 메시지 원본을 유효성 검증하지 않거나 메시지가 전송 시 대체되었는지를 유효성 검증하지 않은 채 전달된 모든 메시지를 처리합니다.

Advanced Message Security (AMS) 기능을 사용하도록 라이센스가 부여된 IBM MQ 의 사용자는 보안 정책의 정의 및 구성을 통해 애플리케이션이 메시지에 보유된 개인 데이터를 처리하는 방법을 제어할 수 있습니다. 보안 정책을 사용하면 디지털 서버 및/또는 암호화를 애플리케이션 간 메시지 데이터에 적용할 수 있습니다.

메시지를 이용하여 메시지가 진짜인지 확인할 때 보안 정책을 사용하여 디지털 서명을 요구하고 유효성 검증할 수 있습니다. AMS 암호화는 메시지 데이터가 의도된 수신인 또는 메시지이고 올바른 복호화 키에 대한 액세스 권한이 있는 경우 읽기 가능한 양식에서 다른 애플리케이션이 디코딩만 할 수 있는 인코딩된 버전으로 변환하는 방법을 제공합니다.

SSL 및 인증서를 사용하여 네트워크 연결을 보호하는 방법에 대한 자세한 정보는 IBM MQ 제품 문서의 다음 주제를 참조하십시오.

- IBM MQ의 TLS 보안 구성
- AMS 개요

데이터 삭제

IBM MQ는 제품에 제공된 데이터를 삭제하기 위해 명령 및 사용자 인터페이스 조치를 제공합니다. 이는 IBM MQ 사용자가 특정 개인과 관련된 데이터를 삭제할 수 있음을 의미합니다.

- GDPR 클라이언트 데이터 삭제를 준수하기 위해 고려할 IBM MQ 동작의 영역
  - 다음을 수행하여 애플리케이션 큐에 저장된 메시지 데이터를 삭제합니다.
    - 메시지 만기를 사용하거나 메시징 API 또는 도구로 개별 메시지를 제거합니다.
    - 비지속적 메시지 클래스가 정상인 큐에 보관된 메시지가 비지속적임을 지정하고 큐 관리자를 재시작합니다.
    - 관리 면에서 큐를 지웁니다.
    - 큐를 삭제합니다.
  - 다음을 수행하여 토픽에 저장한 보유된 발행 데이터를 삭제합니다.
    - 메시지가 비지속적임을 지정하고 큐 관리자를 재시작합니다.
    - 메시지 만기를 사용하거나 보유된 데이터를 새 데이터로 바꿉니다.
    - 관리 면에서 토픽 문자열을 지웁니다.
  - 전체 큐 관리자를 삭제하여 큐 관리자에 저장된 데이터를 삭제하고, 고가용성 또는 재해 복구를 위해 복제된 사본을 삭제합니다.
  - 추적 디렉토리에서 파일을 삭제하여 서비스 추적 명령으로 저장한 데이터를 삭제합니다.
  - 오류 디렉토리에서 파일을 삭제하여 저장된 FFST 데이터를 삭제합니다.
  - 주소 공간 및 커플링 기능 덤프를 삭제합니다(z/OS에서).
  - 이와 같은 데이터의 아카이브, 백업 또는 다른 사본을 삭제합니다.
- GDPR 계정 데이터 삭제를 준수하기 위해 고려할 IBM MQ 동작의 영역
  - 삭제하여 큐 관리자 및 써드파티 서비스에 연결하기 위해 IBM MQ가 저장한 계정 데이터 및 환경 설정을 삭제할 수 있습니다(아카이브, 백업 또는 복제된 사본 포함).
    - 신임 정보를 저장한 큐 관리자 인증 정보 오브젝트.
    - 사용자 ID를 참조하는 큐 관리자 권한 레코드.
    - 특정 IP 주소, 인증서 DN 또는 사용자 ID를 맵핑하거나 차단하는 큐 관리자 채널 인증 규칙.
    - 큐 관리자 및 파일 서버에 대한 인증을 위해 IBM MQ Managed File Transfer 에이전트, 로거 및 MQ Explorer MFT 플러그인에서 사용하는 신임 정보 파일.
    - SSL/TLS 연결 또는 IBM MQ Advanced Message Security(AMS)에서 사용할 수 있는 키 저장소에서 개인에 대한 정보를 표시하거나 포함하는 X.509 디지털 인증서.
    - 시스템 로그 파일의 해당 계정에 대한 참조를 포함한, IBM MQ Appliance의 개별 사용자 계정.

- IBM MQ Explorer 작업공간 메타데이터 및 Eclipse 설정.
- 비밀번호 환경 설정에 지정된 IBM MQ Explorer 비밀번호 저장소.
- IBM MQ 콘솔 및 mqweb 서버 구성 파일.
- IBM MQ Internet Pass-Thru 구성 파일 및 키 저장소.

자세한 내용은 다음을 참조하십시오.

- MFT 및 IBM MQ 연결 인증
- ProtocolBridgeCredentials.xml 파일을 사용하여 파일 서버의 신임 정보 맵핑
- IBM MQ Console 사용자 및 역할 구성

데이터 모니터링

IBM MQ는 사용자가 애플리케이션 및 큐 관리자가 수행하는 방식에 대해 이해하기 위해 사용할 수 있는 다양한 모니터링 기능을 제공합니다.

IBM MQ는 큐 관리자 오류 로그를 관리하는 데 도움이 되는 여러 기능도 제공합니다.

자세한 내용은 다음을 참조하십시오.

- IBM MQ 네트워크 모니터링
- 진단 메시지 서비스
- QMErrorLog 서비스
- IBM MQ Appliance 모니터링 및 보고

개인정보 사용 제한 기능

이 문서에 요약된 기능을 사용하면 IBM MQ에서 일반 사용자가 개인 데이터 사용을 제한할 수 있습니다.

IBM MQ 메시지 큐는 데이터베이스와 동일한 방식으로 영구 데이터 저장소로 사용되면 안 됩니다. 이는 특히 GDPR의 영향을 받는 애플리케이션 데이터를 처리할 때 해당됩니다.

검색 조회를 통해 데이터를 찾을 수 있는 데이터베이스와 달리 큐, 메시지 및 메시지의 상관 ID를 모르면 메시지 데이터를 찾기 어렵습니다.

개인 데이터가 포함된 메시지를 쉽게 식별하고 찾을 수 있는 경우 표준 IBM MQ 메시징 기능을 사용하여 메시지 데이터를 액세스하고 수정할 수 있습니다.

파일 핸들링

1. IBM MQ Managed File Transfer에서는 전송된 파일에 대한 맬웨어 검사를 수행하지 않습니다. 파일 데이터를 전송 중 수정하지 않도록 파일은 있는 그대로 전송되며 무결성 검사를 수행합니다. 소스와 대상 체크섬은 전송 상태 발행의 일부로 발행됩니다. 일반 사용자는 MFT에서 파일을 전송하기 전과 MFT에서 파일을 원격 엔드 포인트로 전송한 후에 환경에 맞게 적절히 맬웨어 검사를 구현하는 것이 좋습니다.
2. IBM MQ Managed File Transfer에서는 MIME 유형이나 파일 확장자에 대한 조치를 수행하지 않습니다. MFT에서는 파일을 읽고 입력 파일에서 읽은 그대로 바이트를 전송합니다.

단일 큐 관리자에 기반한 아키텍처

가장 단순한 IBM MQ 아키텍처는 단일 큐 관리자 구성 및 사용입니다.

IBM MQ 아키텍처를 계획하기 전에 기본 IBM MQ 개념을 숙지하십시오. IBM MQ 기술 개요를 참조하십시오.

단일 큐 관리자를 사용하는 가능한 많은 아키텍처는 다음 절에서 설명됩니다.

- 17 페이지의 『서비스에 액세스하는 로컬 애플리케이션이 있는 단일 큐 관리자』
- 17 페이지의 『클라이언트로서 서비스에 액세스하는 원격 애플리케이션이 있는 단일 큐 관리자』
- 17 페이지의 『발행/구독 구성의 단일 큐 관리자』

서비스에 액세스하는 로컬 애플리케이션이 있는 단일 큐 관리자

단일 큐 관리자에 기반한 첫 번째 아키텍처는 서비스에 액세스하는 애플리케이션이 서비스를 제공하는 애플리케이션과 같은 시스템에서 실행 중입니다. IBM MQ 큐 관리자는 서비스를 요청하는 애플리케이션과 서비스를 제공하는 애플리케이션 간에 비동기 상호통신을 제공합니다. 이는 애플리케이션 중 하나가 장기간 오프라인 상태일 때에도 애플리케이션 간의 통신이 계속될 수 있음을 의미합니다.

클라이언트로서 서비스에 액세스하는 원격 애플리케이션이 있는 단일 큐 관리자

단일 큐 관리자에 기반한 두 번째 아키텍처에는 서비스를 제공하는 애플리케이션에서 원격으로 실행하는 여러 애플리케이션이 있습니다. 원격 애플리케이션은 서비스에 대한 각기 다른 시스템에서 실행 중입니다. 애플리케이션은 단일 큐 관리자에 클라이언트로서 연결합니다. 이는 단일 큐 관리자를 통해 여러 시스템에 한 서비스에 대한 액세스를 제공할 수 있음을 의미합니다.

이 아키텍처의 제한사항은 애플리케이션이 작동하려면 네트워크 연결이 사용 가능해야 한다는 점입니다. 네트워크에서 애플리케이션과 큐 관리자 간의 상호작용은 동기식입니다.

발행/구독 구성의 단일 큐 관리자

단일 큐 관리자를 사용한 대체 아키텍처는 발행/구독 구성을 사용하는 것입니다. 발행/구독 메시징에서는 해당 정보의 이용자로부터 정보 제공자를 분리할 수 있습니다. 이는 메시지를 넣을 큐 이름과 같이 대상 애플리케이션에 대한 정보를 애플리케이션이 알아야 하는 이전에 설명한 아키텍처의 포인트 투 포인트 메시징 스타일과 다릅니다. 송신 애플리케이션이 IBM MQ 발행/구독을 사용하여 정보의 주제에 기초해서 지정된 토픽으로 메시지를 발행합니다. IBM MQ는 구독을 통해 이 주제에 대한 관심을 등록한 애플리케이션으로 메시지 분배를 핸들링합니다. 수신 애플리케이션도 수신할 메시지의 소스에 대한 정보를 알고 있지 않아도 됩니다. 자세한 정보는 발행/구독 메시징 및 단일 큐 관리자 발행/구독 구성 예를 참조하십시오.

관련 개념

IBM MQ 소개

관련 태스크

5 페이지의 『IBM MQ 아키텍처 계획』

IBM MQ 환경을 계획할 때, IBM MQ가 단일 및 다중 큐 관리자 아키텍처와, 포인트-투-포인트 및 발행/구독 메시징 스타일에 대해 제공하는 지원을 고려하십시오. 또한 자원 요구사항과, 로깅 및 백업 기능 사용을 계획하십시오.

Multiplatforms에서 큐 관리자 작성 및 관리

다중 큐 관리자에 기반한 아키텍처

분산 메시징 큐잉 기술을 사용하여 다수의 큐 관리자를 구성하고 사용하는 IBM MQ 아키텍처를 작성할 수 있습니다.

IBM MQ 아키텍처를 계획하기 전에 기본 IBM MQ 개념을 숙지하십시오. IBM MQ 기술 개요를 참조하십시오.

IBM MQ 아키텍처는 추가 큐 관리자를 추가해서 서비스를 제공하는 애플리케이션을 변경하지 않고도 변경이 가능합니다.

애플리케이션은 큐 관리자와 동일한 시스템에서 호스트될 수 있으며 다른 시스템의 또 다른 큐 관리자에서 호스트되는 서비스와 비동기식으로 통신할 수 있습니다. 또는 서비스에 액세스하는 애플리케이션이 클라이언트로서 큐 관리자에 연결한 후 다른 큐 관리자의 서비스에 대한 비동기 액세스를 제공할 수 있습니다.

다른 큐 관리자와 해당 큐에 연결하는 라우트는 분산 큐잉 기술을 사용하여 정의됩니다. 아키텍처 내의 큐 관리자는 채널을 사용하여 연결됩니다. 채널은 큐 관리자의 구성에 따라 한 큐 관리자에서 다른 큐 관리자에 한 방향으로, 메시지를 자동으로 이동시키는 데 사용됩니다.

IBM MQ 네트워크 계획의 상위 레벨 개요는 19 페이지의 『분산 큐 관리자 네트워크 설계』의 내용을 참조하십시오.

IBM MQ 아키텍처의 채널을 계획하는 방법에 대한 정보는 IBM MQ분산 큐잉 기술을 참조하십시오.

분산 큐 관리를 통해 큐 관리자 간의 통신을 작성하고 모니터할 수 있습니다. 분산 큐 관리에 대한 자세한 정보는 분산 큐 관리 소개를 참조하십시오.

관련 태스크

5 페이지의 『IBM MQ 아키텍처 계획』
IBM MQ 환경을 계획할 때, IBM MQ가 단일 및 다중 큐 관리자 아키텍처와, 포인트-투-포인트 및 발행/구독 메시징 스타일에 대해 제공하는 지원을 고려하십시오. 또한 자원 요구사항과, 로깅 및 백업 기능 사용을 계획하십시오.

Multiplatforms에서 큐 관리자 작성 및 관리

분산 큐와 클러스터 계획

분산 큐 관리자에서 호스트되는 큐를 수동으로 연결하거나 큐 관리자 클러스터를 작성하고 제품이 큐 관리자를 연결하도록 할 수 있습니다. 분산 메시징 네트워크에 적합한 토폴로지를 선택하려면 수동 제어, 네트워크 크기, 변경 빈도, 가용성, 확장성에 대한 요구사항을 고려해야 합니다.

시작하기 전에

이 태스크는 사용자가 분산 메시징 네트워크의 개념과 작동 방식을 이해하고 있다고 가정합니다. 기술 개요는 분산 큐잉 및 클러스터를 참조하십시오.

이 태스크 정보

분산 메시징 네트워크를 작성하려면 다른 큐 관리자에서 호스트되는 큐를 연결할 채널을 수동으로 구성하거나 큐 관리자 클러스터를 작성할 수 있습니다. 클러스터링을 통해 큐 관리자는 추가 채널 정의 또는 리모트 큐 정의를 설정할 필요 없이 구성과 관리를 단순화하면서 서로 통신할 수 있습니다.

분산 발행/구독 네트워크에 적합한 토폴로지를 선택하려면 다음의 광범위한 질문을 고려해야 합니다.

- 네트워크의 연결에 대한 수동 제어가 어느 정도 필요합니까?
- 네트워크는 얼마나 큽니까?
- 어느 정도 동적입니까?
- 가용성 및 확장성 요구사항은 무엇입니까?

프로시저

- 네트워크의 연결에 대한 수동 제어가 어느 정도 필요한지 고려하십시오.

  몇 개의 연결만 필요한 경우나 개별 연결을 매우 조밀하게 정의해야 하는 경우에는 아마도 네트워크를 수동으로 작성해야 합니다.

  논리적으로 관련되어 있으며 데이터와 애플리케이션을 공유해야 하는 다수의 큐 관리자가 필요한 경우 큐 관리자 클러스터에 큐 관리자를 전부 그룹화할 것을 고려해야 합니다.
- 필요한 네트워크의 크기를 추정하십시오.

  a) 필요한 큐 관리자의 수를 추정하십시오. 큐는 둘 이상의 큐 관리자에서 호스트할 수 있음을 기억하십시오.

  b) 클러스터 사용을 고려 중이면 전체 저장소 역할을 할 두 개의 큐 관리자를 추가하십시오.

  큰 네트워크에서는 수동 구성과 연결 유지보수가 매우 시간 소모적일 수 있으므로 클러스터 사용을 고려해야 합니다.
- 네트워크 활동이 얼마나 동적인지 고려하십시오.

  사용 중인 큐를 성능 큐 관리자에서 호스트되도록 계획하십시오.

  큐를 자주 작성하고 삭제할 것으로 예상하는 경우 클러스터 사용을 고려하십시오.
- 가용성 및 확장성 요구사항을 고려하십시오.

  a) 큐 관리자의 고가용성을 보장할 필요가 있는지 결정하십시오. 그러한 경우 이 요구사항이 적용된 큐 관리자의 수를 추정하십시오.

  b) 일부 큐 관리자가 다른 큐 관리자보다 덜 기능적인지 고려하십시오.

  c) 일부 큐 관리자에 대한 통신 링크가 다른 링크에 비해 더 취약한지 고려하십시오.

  d) 다수의 큐 관리자에서 큐를 호스트할 것을 고려하십시오.

수동으로 구성된 네트워크와 클러스터는 모두 고가용적이며 확장 가능하도록 구성할 수 있습니다. 클러스터를 사용하면 전체 저장소 역할을 할 두 개의 추가 큐 관리자를 정의해야 합니다. 전체 저장소가 두 개이면 한 전체 저장소가 사용 불가능해도 클러스터가 계속해서 작동하게 됩니다. 전체 저장소 큐 관리자가 강력하고, 성능이 우수하며, 네트워크 연결성이 양호한지 확인하십시오. 다른 작업에는 전체 저장소 큐 관리자를 사용하도록 계획하지 마십시오.

- 이러한 계산을 기반으로, 제공된 링크를 사용하여 큐 관리자 간의 연결을 수동으로 구성할지 또는 클러스터를 사용할지 결정하는 데 도움을 받으십시오.

다음에 수행할 작업

이제 분산 메시징 네트워크를 구성할 준비가 되었습니다.

관련 태스크

분산 큐잉 구성

큐 관리자 클러스터 구성

분산 큐 관리자 네트워크 설계

IBM MQ는 큐 관리자 및 채널을 사용하여 네트워크에서 그리고 애플리케이션 간에 데이터를 보내고 받습니다. 네트워크 계획은 네트워크에서 이 시스템에 연결하기 위한 프레임워크를 작성하는 요구사항 정의를 포함합니다.

채널은 통신해야 하는 다른 시스템과 사용자 시스템 간에 작성할 수 있습니다. 멀티홉 채널을 작성하여 직접 연결이 없는 시스템에 연결할 수 있습니다. 시나리오에 설명된 메시지 채널 연결은 20 페이지의 그림 1에 네트워크 다이어그램으로 표시됩니다.

서로 다른 물리적 네트워크에 있는 시스템 간에 채널을 작성하거나, 방화벽을 통해 통신하는 채널을 작성하는 경우에는 IBM MQ Internet Pass-Thru를 사용하면 구성을 간소화할 수 있습니다. 자세한 정보는 IBM MQ Internet Pass-Thru의 내용을 참조하십시오.

채널 및 전송 큐 이름

전송 큐에 어느 이름이나 지정할 수 있습니다. 하지만 해당하는 경우 혼란을 피하기 위해 목적지 큐 관리자 이름 또는 큐 관리자 알리어스 이름과 동일한 이름을 제공할 수 있습니다. 이는 사용하는 라우트와 전송 큐를 연관시켜서 중간(멀티호핑된) 큐 관리자를 통해 작성된 병렬 라우트의 명확한 개요를 제공합니다.

이는 채널 이름에는 명확하지 않습니다. 예를 들어, QM2에 대한 20 페이지의 그림 1의 채널 이름은 들어오는 채널과 나가는 채널에 따라 달라야 합니다. 모든 채널 이름은 전송 큐 이름을 여전히 포함할 수 있지만 고유한 이름이 되도록 규정해야 합니다.

예를 들어, QM2에는 QM1에서 들어오는 QM3 채널이 하나 있고 QM3으로 나가는 QM3 채널이 하나 있습니다. 이름을 고유하게 만들기 위해 첫 번째 이름을 QM3_from_QM1으로 지정하고 두 번째 이름을 QM3_from_QM2로 지정합니다. 이러한 방식으로 채널 이름은 이름의 첫 번째 부분에 전송 큐 이름을 표시합니다. 방향 및 인접 큐 관리자 이름은 이름의 두 번째 부분에 표시됩니다.

20 페이지의 그림 1의 제안된 채널 이름 표는 20 페이지의 표 1에 제공됩니다.

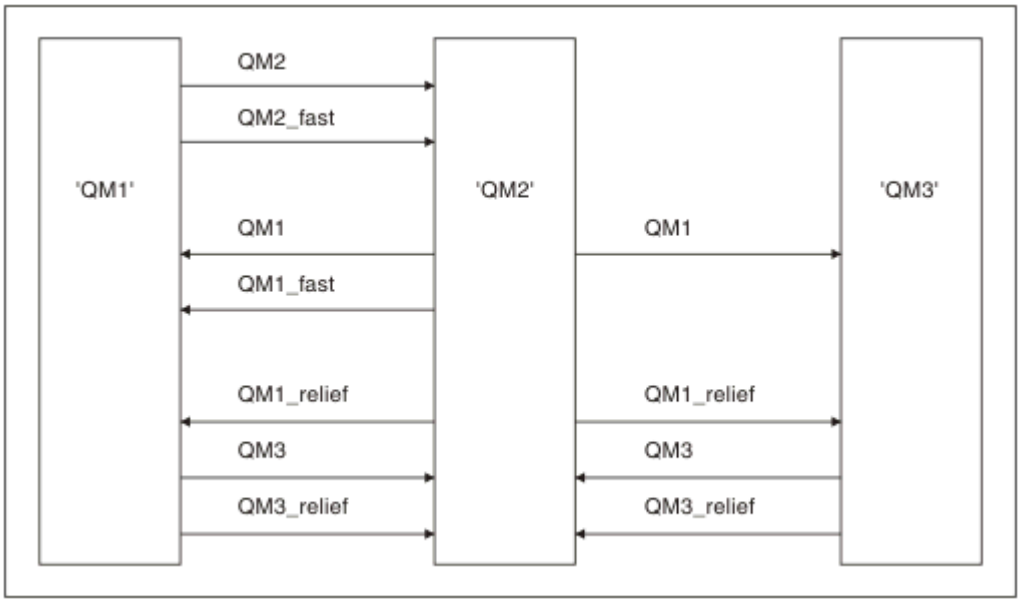


그림 *1.* 모든 채널이 표시된 네트워크 다이어그램

표 *1.* 채널 이름 예			
라우트 이름	**채널을 호스트하는 큐 관리자**	**전송 큐 이름**	**제안된 채널 이름**
QM1	QM1 & QM2	QM1(QM2에서)	QM1.from.QM2
QM1	QM2 & QM3	QM1(QM3에서)	QM1.from.QM3
QM1_fast	QM1 & QM2	QM1_fast(QM2에서)	QM1_fast.from.QM2
QM1_relief	QM1 & QM2	QM1_relief(QM2에서)	QM1_relief.from.QM2
QM1_relief	QM2 & QM3	QM1_relief(QM3에서)	QM1_relief.from.QM3
QM2	QM1 & QM2	QM2(QM1에서)	QM2.from.QM1
QM2_fast	QM1 & QM2	QM2_fast(QM1에서)	QM2_fast.from.QM1
QM3	QM1 & QM2	QM3(QM1에서)	QM3.from.QM1
QM3	QM2 & QM3	QM3(QM2에서)	QM3.from.QM2
QM3_relief	QM1 & QM2	QM3_relief(QM1에서)	QM3_relief.from.QM1
QM3_relief	QM2 & QM3	QM3_relief(QM2에서)	QM3_relief.from.QM2

참고:

1. z/OS IBM MQ for z/OS에서는 큐 관리자 이름이 4자로 제한됩니다.
2. 네트워크의 모든 채널 이름을 고유하게 지정하십시오. 20 페이지의 표 1에 표시된 대로, 채널 이름에 소스 및 대상 큐 관리자 이름을 포함하는 것이 좋은 방법입니다.

네트워크 계획표

네트워크 작성 시에는 팀의 다른 멤버가 계획을 구현하는 네트워크 계획표의 상위 레벨 기능이 하나 더 있다고 가정합니다.

광범위하게 사용되는 애플리케이션의 경우, 21 페이지의 그림 2에 표시된 대로 로컬 액세스 사이트 간의 광대역 링크를 사용하여 메시지 트래픽의 집중을 위한 로컬 액세스 사이트의 관점에서 생각하는 것이 보다 경제적입니다.

이 예에는 두 개의 기본 시스템 및 여러 위성 시스템이 있습니다. 실제 구성은 비즈니스 고려사항에 따라 다릅니다. 두 개의 집선기 큐 관리자는 편의상 중앙에 위치해 있습니다. 각 QM 집선기에는 로컬 큐 관리자로 향하는 메시지 채널이 있습니다.

- QM 집선기 1에는 세 개의 로컬 큐 관리자 QM1, QM2, QM3 각각에 대한 메시지 채널이 있습니다. 이 큐 관리자를 사용하는 애플리케이션은 QM 집선기를 통해 서로 통신할 수 있습니다.
- QM 집선기 2에는 세 개의 로컬 큐 관리자 QM4, QM5, QM6 각각에 대한 메시지 채널이 있습니다. 이 큐 관리자를 사용하는 애플리케이션은 QM 집선기를 통해 서로 통신할 수 있습니다.
- QM 집선기에는 서로 간에 메시지 채널이 있어서 큐 관리자의 애플리케이션이 다른 큐 관리자의 다른 애플리케이션과 메시지를 교환할 수 있습니다.

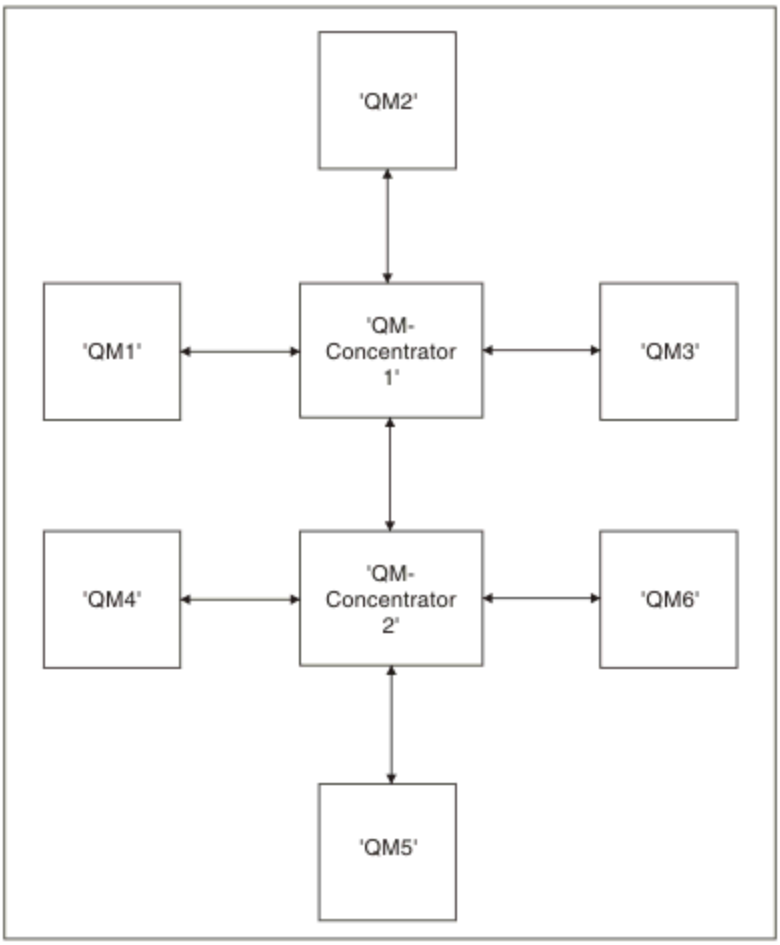


그림 *2. QM* 집선기가 표시된 네트워크 다이어그램

클러스터 설계

클러스터는 초기 구성 및 진행 중인 관리를 모두 단순화하는 방식으로 큐 관리자를 상호 연결하기 위한 메커니즘을 제공합니다. 클러스터가 제대로 기능하고 가용성 및 응답성의 필수 레벨을 달성할 수 있도록 클러스터를 주의해서 설계해야 합니다.

시작하기 전에

클러스터링 개념 소개는 다음 주제를 참조하십시오.

- 분산 큐잉 및 클러스터
- 27 페이지의 『클러스터링과 분산 큐잉의 비교』

- 클러스터의 컴포넌트

큐 관리자 클러스터를 설계할 때 몇 가지를 결정해야 합니다. 먼저 클러스터에서 클러스터 정보의 전체 저장소를 보유할 큐 관리자를 결정해야 합니다. 작성한 큐 관리자가 클러스터에서 작업할 수 있습니다. 이 용도로 임의의 수의 큐 관리자를 선택할 수 있지만 이상적인 수는 2입니다. 전체 저장소를 보유할 큐 관리자 선택에 대한 정보는 29 페이지의 『전체 저장소를 보유할 클러스터 큐 관리자 선택 방법』의 내용을 참조하십시오.

클러스터 설계에 대한 자세한 정보는 다음 주제를 참조하십시오.

- 34 페이지의 『클러스터 예』
- 30 페이지의 『클러스터 조직화』
- 30 페이지의 『클러스터 이름 지정 규칙』
- z/OS 31 페이지의 『Queue sharing groups and clusters』
- 32 페이지의 『중첩 클러스터』

다음에 수행할 작업

클러스터 구성 및 작업에 대한 자세한 정보는 다음 주제를 참조하십시오.

- 클러스터에서 통신 설정
- 큐 관리자 클러스터 구성
- 클러스터로(부터) 메시지 라우팅
- 워크로드 관리에 클러스터 사용

클러스터 구성에 대한 자세한 정보는 33 페이지의 『클러스터링 팁』의 내용을 참조하십시오.

다중 클러스터 전송 큐 사용 방법 계획

명시적으로 전송 큐를 정의하거나 시스템이 전송 큐를 생성하도록 할 수 있습니다. 사용자가 직접 전송 큐를 정의할 경우 큐 정의에 대해 추가 제어 권한을 가집니다. z/OS z/OS에서는 메시지가 보관되는 페이지 세트에 대해 더 많은 제어를 할 수도 있습니다.

전송 큐 정의

다음은 전송 큐를 정의하는 두 가지 방법입니다.

- 다음과 같이 자동으로 새 큐 관리자 속성 DEFCLXQ 사용합니다.

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

  DEFCLXQ(SCTQ)은 기본적으로 클러스터-전송자 채널이 SYSTEM.CLUSTER.TRANSMIT.QUEUE임을 표시합니다. 이는 기본값입니다.

  DEFCLXQ(CHANNEL)은 기본적으로 클러스터-전송자 채널이 별도의 전송 큐 SYSTEM.CLUSTER.TRANSMIT.*channel name*를 사용함을 나타냅니다. 전송 큐는 큐 관리자가 자동으로 정의합니다. 자세한 정보는 23 페이지의 『자동 정의 클러스터 전송 큐』의 내용을 참조하십시오.
- 수동으로 CLCHNAME 속성에 대해 지정된 값을 사용하여 전송 큐를 정의합니다. CLCHNAME 속성은 클러스터-전송자 채널에서 사용하는 전송 큐를 표시합니다. z/OS 전송 큐를 수동으로 정의하는 경우z/OS , 보다25 페이지의 『수동으로 정의된 클러스터 전송 큐 계획』 자세한 내용은.

필요한 보안

스위치를 자동이나 수동으로 시작하려면 채널을 시작할 권한이 있어야 합니다.

전송 큐로 사용된 큐를 정의하려면 큐를 정의할 수 있는 표준 IBM MQ 권한이 필요합니다.

변경사항을 구현하기에 적합한 시간

클러스터-전송자 채널에서 사용한 전송 큐를 변경할 경우 다음 사항을 고려하여 업데이트 시간을 할당해야 합니다.

- 채널이 전송 큐를 전환하는 데 필요한 시간은 이전 전송 큐의 총 메시지 수, 이동해야 하는 메시지 수 및 메시지 크기에 따라 달라집니다.
- 변경이 수행되는 동안에도 애플리케이션은 계속해서 메시지를 전송 큐에 넣을 수 있습니다. 이렇게 되면 상태 전이 시간이 증가될 수 있습니다.
- 언제든지, 가능하면 워크로드가 낮을 때 CLCHNAME 또는 전송 큐의 CLCHNAME 매개변수를 변경할 수 있습니다.

  직후에는 아무 일도 일어나지 않습니다.
- 채널이 시작 또는 재시작할 때만 변경이 이루어집니다. 채널이 시작하면 현재 구성을 확인하고 필요한 경우 새 전송 큐로 전환합니다.
- 다음은 전송 큐와 클러스터-송신자 채널의 연관을 대체할 수 있는 몇 가지 변경사항입니다.
  - 전송 큐의 CLCHNAME 속성 값을 대체하여 덜 특정적이거나 비어 있도록 합니다.
  - 전송 큐의 CLCHNAME 속성 값을 대체하여 CLCHNAME를 더 특정적으로 만듭니다.
  - CLCHNAME이 지정된 큐를 삭제합니다.
  - 큐 관리자 속성 DEFCLXQ를 변경합니다.

전환 수행 기간

상태 전이 주기 동안 채널에 대한 메시지는 한 전송 큐에서 다른 전송 큐로 이동됩니다. 채널이 전송 큐를 전환하는 데 필요한 시간은 이전 전송 큐의 총 메시지 수 및 이동해야 하는 메시지 수에 따라 달라집니다.

몇 천 개의 메시지가 포함된 큐의 경우 메시지를 이동하는 데 1초 미만이 걸려야 합니다. 실제 시간은 메시지의 수와 크기에 따라 달라집니다. 큐 관리자는 매초마다 몇 MB의 메시지를 이동할 수 있어야 합니다.

변경이 수행되는 동안에도 애플리케이션은 계속해서 메시지를 전송 큐에 넣을 수 있습니다. 이렇게 되면 상태 전이 시간이 증가될 수 있습니다.

변경사항을 적용하려면 해당하는 클러스터-송신자 채널을 다시 시작해야 합니다. 큐 관리자가 사용 중이지 않고 클러스터 전송 큐에 메시지가 거의 없을 때 전송 큐 구성을 변경하는 것이 가장 좋습니다.

그만큼**runswchl** 명령 z/OS 아니면 그 CSQUTIL의 SWITCH CHANNEL(*) STATUS 명령 ~에z/OS 클러스터 송신자 채널의 상태와 전송 큐 구성에 미해결된 보류 중인 변경사항을 쿼리하는 데 사용할 수 있습니다.

변경 구현 방법

자동 또는 수동으로 다중 클러스터 전송 큐에 대해 변경을 수행하는 방법에 대한 자세한 내용은 다중 클러스터 전송 큐를 사용하는 시스템 구현을 참조하십시오.

변경 실행 취소

z/OS

문제점이 발생하는 경우 변경사항을 백아웃하는 방법에 대한 세부사항은 z/OS에서 전송 큐에 대한 변경 실행 취소 를 참조하십시오.

자동 정의 클러스터 전송 큐
시스템이 전송 큐를 생성하도록 할 수 있습니다.

시작하기 전에

z/OS z/OS에서 클러스터 전송 큐를 수동으로 설정하려면 25 페이지의 『수동으로 정의된 클러스터 전송 큐 계획』의 내용을 참조하십시오.

이 태스크 정보

채널에 연관된 수동으로 정의된 클러스터 전송 큐가 없고 DEFCLXQ(CHANNEL)를 지정하는 경우, 채널이 큐 관리자를 시작하면 클러스터 송신자 채널에 대해 영구적 동적 큐가 자동으로 정의됩니다. SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 모델 큐는 이름이 SYSTEM.CLUSTER.TRANSMIT.ChannelName인 영구 동적 클러스터 전송 큐를 자동으로 정의하는 데 사용됩니다.

중요사항: z/OS IBM MQ 8.0에서 큐 관리자에는 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE가 없습니다. IBM MQ 8.0 에서 이 버전으로 직접 마이그레이션할 수 없습니다. IBM MQ 8.0에서 마이그레이션된 큐 관리자에 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 를 추가하는 방법에 대한 정보는 큐 관리자를 마이그레이션하는 데 사용한 임시 버전에 대한 문서에서 이 주제를 참조하십시오.

프로시저

1. *DEFCLXQ* 큐 관리자 속성을 사용하십시오.
   이 속성에 대한 자세한 정보는 ALTER QMGR을 참조하십시오.

   두 개의 옵션이 있습니다.

   SCTQ
   이 옵션은 기본값이며 이는 단일 SYSTEM.CLUSTER.TRANSMIT.QUEUE를 사용하는 것을 의미합니다.

   CHANNEL
   다중 클러스터 전송 큐 사용을 의미합니다.
2. 새 연관으로 전환하려면 다음을 수행하십시오.
   - 채널을 중지하고 다시 시작하십시오.
   - 채널은 새 전송 큐 정의를 사용합니다.
   - 메시지는 이전 큐에서 새 전송 큐로 상태 전이 스위치 프로세스로 전송됩니다.

   모든 애플리케이션 메시지는 이전 정의로 넣어집니다.

   이전 큐의 메시지 수가 0이 되면 새 메시지가 새 전송 큐에 직접 배치됩니다.
3. 전환 프로세스가 완료되는 시기를 모니터하려면 다음을 수행하십시오.
   a) 채널로 시작되는 전송 큐의 스위치는 백그라운드에서 실행되고 관리자는 완료되는 시기를 판별하기 위해 큐 관리자 작업 로그를 모니터할 수 있습니다.
   b) 스위치의 진행 상태를 표시하기 위해 작업 로그에서 메시지를 모니터하십시오.
   c) 원하는 채널만 이 전송 큐를 사용하는지 확인하려면, DIS CLUSQMGR(*) 명령을 실행하십시오. 여기서, 전송 큐를 정의하는 전송 큐 특성의 예로는 APPQMGR.CLUSTER1.XMITQ이(가) 있습니다.
   d) z/OS
   CSQUTIL에서 SWITCH CHANNEL (*) STATUS 명령을 사용하십시오.
   이 옵션은 보류 중인 변경 중에서 미해결인 변경 및 전송 큐 사이에서 이동해야 하는 메시지 수를 알려줍니다.

결과

하나 이상의 클러스터 전송 큐를 설정했습니다.

관련 태스크

25 페이지의 『수동으로 정의된 클러스터 전송 큐 계획』
~에IBM MQ for z/OS , 전송 큐를 직접 정의하는 경우 정의 및 메시지가 보관되는 페이지 세트를 더 잘 제어할 수 있습니다.

관련 참조

ALTER QMGR
DISPLAY CLUSQMGR

z/OS 수동으로 정의된 클러스터 전송 큐 계획
~에IBM MQ for z/OS , 전송 큐를 직접 정의하는 경우 정의 및 메시지가 보관되는 페이지 세트를 더 잘 제어할 수 있습니다.

시작하기 전에

클러스터 전송 큐를 자동으로 설정하려면 23 페이지의 『자동 정의 클러스터 전송 큐』의 내용을 참조하십시오.

이 태스크 정보

관리자는 전송 큐를 수동으로 정의하고 큐 속성 CLCHNAME을 사용하여 이 큐를 전송 큐로 사용할 클러스터 송신자 채널을 정의합니다.

CLCHNAME의 시작이나 끝에 와일드카드 문자를 포함시키면 단일 큐를 여러 채널에 사용할 수 있습니다.

프로시저

1. 예를 들어, 다음을 입력하십시오.

```
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)
```

   팁: 전송 큐에 사용하려는 페이지 세트(및 버퍼 풀)을 계획해야 합니다. 서로 다른 큐에 대해 서로 다른 페이지 세트를 가질 수 있으며 이들 사이에 격리를 제공할 수 있으므로 하나의 페이지 세트가 채워져도 다른 페이지 세트의 전송 큐에 영향을 주지 않습니다.

   각 채널이 적절한 큐를 선택하는 방법에 대한 정보는 클러스터 전송 큐 및 클러스터 송신자 채널에 대한 작업을 참조하십시오.

   채널이 시작되면 이는 해당 연관을 새 전송 큐로 스위치합니다. 메시지가 손실되지 않도록 하기 위해 큐 관리자는 이전 클러스터 전송 큐에서 새 전송 큐로 순서대로 메시지를 자동으로 전송합니다.
2. CSQUTIL SWITCH 함수를 사용하여 새 연관으로 변경하십시오.
   추가 정보는 클러스터 송신자 채널에 연관된 전송 큐 스위치(SWITCH)를 참조하십시오.
   a) 전송 큐가 변경되는 하나 이상의 채널이 STOPPED 상태가 되도록 이를 STOP하십시오.
      예를 들면, 다음과 같습니다.

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

   b) 전송 큐에서 CLCHNAME(XXXX) 속성을 변경하십시오.
   c) SWITCH 함수를 사용하여 발생 중인 메시지 또는 모니터를 전환하십시오.
      명령

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

      채널을 시작하지 않고 메시지를 이동하십시오.
   d) 채널이 올바른 큐를 사용 중인지 검사하기 위해 하나 이상의 채널을 시작하십시오.
      예를 들면, 다음과 같습니다.

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

   팁: 다음 프로세스에서는 CSQUTIL SWITCH 함수를 사용합니다. 자세한 내용은 다음을 참조하세요. 클러스터 송신자 채널과 연관된 전송 큐 전환(SWITCH) .

이 함수를 사용할 필요는 없지만 이 기능을 사용하는 경우 추가 옵션이 제공됩니다.

- SWITCH CHANNEL (*) STATUS를 사용하면 클러스터 송신자 채널의 스위치 상태를 더 쉽게 식별할 수 있습니다. 이를 통해 관리자는 현재 스위치 중인 채널을 확인하고 해당 채널이 다음에 시작될 때 적용되는 보류 중인 스위치가 포함된 채널을 확인할 수 있습니다.

  이 기능을 사용하지 않는 경우 관리자는 여러 개의 DISPLAY 명령을 사용하고 이 정보를 확인하기 위해 결과 출력을 처리해야 합니다. 관리자는 구성 변경으로 필요한 결과를 얻었는지 확인할 수도 있습니다.
- CSQUTIL을 사용하여 스위치를 시작하면 CSQUTIL은 이 조작의 진행 상태를 계속 모니터하고 스위치가 완료될 때만 종료합니다.

  이를 통해 해당 조작을 배치로 수행하기가 더 쉬워집니다. 또한, CSQUTIL이 다중 채널 스위치를 위해 실행되면 CSQUTIL은 해당 조치를 순차적으로 수행합니다. 이 경우, 여러 개의 스위치가 병렬로 실행되는 것보다 엔터프라이즈에 대해 더 적은 영향을 주게 됩니다.

결과

클러스터 전송 큐 또는 큐를 설정했습니다.z/OS .

액세스 제어 및 다중 클러스터 전송 큐

애플리케이션이 리모트 클러스터 큐에 메시지를 넣을 때 검사하는 세 가지 모드 중에서 선택하십시오. 모드에서는 클러스터 큐에 대해 원격으로 검사하거나 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 대해 로컬로 검사하거나 클러스터 큐 또는 클러스터 큐 관리자의 로컬 프로파일에 대해 검사합니다.

IBM MQ는 사용자가 리모트 큐에 메시지를 넣을 권한이 있는지에 대한 로컬 또는 로컬 및 리모트 검사의 선택사항을 제공합니다. 일반 IBM MQ 애플리케이션은 로컬 검사만을 사용하며 로컬 큐 관리자에 대한 액세스 검사를 신뢰하는 리모트 큐 관리자에 의존합니다. 원격 검사를 사용하지 않으면 원격 메시지 채널 프로세스 권한이 있는 대상 큐에 메시지를 넣습니다. 원격 검사를 사용하려면 수신 채널의 넣기 권한을 컨텍스트 보안에 설정해야 합니다.

애플리케이션이 여는 큐에 대한 로컬 검사가 수행됩니다. 분산 큐잉에서는 애플리케이션이 일반적으로 리모트 큐 정의를 열고 리모트 큐 정의에 대한 액세스 검사가 수행됩니다. 전체 라우팅 헤더에 메시지를 넣는 경우에는 전송 큐에 대해 검사합니다. 애플리케이션이 로컬 큐 관리자에 있지 않은 클러스터 큐를 여는 경우 검사할 로컬 오브젝트가 없습니다. 액세스 제어는 클러스터 전송 큐, SYSTEM.CLUSTER.TRANSMIT.QUEUE을(를) 대상으로 검사합니다. 다중 클러스터 전송 큐의 경우에도 리모트 클러스터 큐에 대한 로컬 액세스 제어 검사가 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 대해 수행됩니다.

로컬 또는 원격 검사의 선택은 두 가지 극단 상황의 선택입니다. 원격 검사는 세분화되어 있습니다. 모든 사용자가 클러스터 큐에 넣으려면 클러스터의 모든 큐 관리자에 대한 액세스 제어 프로파일을 보유해야 합니다. 로컬 검사는 대략적입니다. 모든 사용자가 연결된 큐 관리자의 클러스터 전송 큐에 대한 액세스 제어 프로파일만 있으면 됩니다. 이 프로파일로 큐 관리자는 클러스터의 어느 큐 관리자에 있는 클러스터 큐에나 메시지를 넣을 수 있습니다.

관리자는 클러스터 큐에 대한 액세스 제어를 설정할 수 있는 다른 방법이 있습니다. **setmqaut** 명령을 사용하여 클러스터의 큐 관리자에 클러스터 큐에 대한 보안 프로파일을 작성할 수 있습니다. 큐 이름만 지정한 채 리모트 클러스터 큐를 로컬로 여는 경우 프로파일이 적용됩니다. 리모트 큐 관리자에 대한 프로파일을 설정할 수도 있습니다. 그렇게 되면 큐 관리자가 완전한 이름을 제공하여 클러스터 큐를 여는 사용자의 프로파일을 검사할 수 있습니다.

큐 관리자 스탠자, **ClusterQueueAccessControl**을(를) RQMName(으)로 변경하는 경우에만 새 프로파일이 작동됩니다. 기본값은 Xmitq입니다. 기존 애플리케이션이 클러스터 큐를 사용하는 모든 클러스터 큐에 대한 프로파일을 작성해야 합니다. 프로파일을 작성하지 않고 스탠자를 RQMName으로 변경할 경우 애플리케이션이 실패할 수 있습니다.

팁: 클러스터 큐 액세스 검사는 리모트 큐잉에 적용되지 않습니다. 액세스 검사는 여전히 로컬 정의에 대해 수행됩니다. 변경사항은 동일한 접근법에 따라 클러스터 큐와 클러스터 토픽에 대한 액세스 검사를 구성할 수 있음을 의미합니다. z/OS 변경사항은 클러스터 큐에 대한 액세스 검사 접근 방법을 z/OS에서와 거의 유사하게 조정하기도 합니다. z/OS에서 액세스 검사를 설정하는 명령은 다르지만 액세스 검사는 모두 오브젝트 자체가 아닌 프로파일에 대해 수행합니다.

관련 개념

42 페이지의 『클러스터링: 다중 클러스터 전송 큐를 사용하여 애플리케이션 격리』

클러스터의 큐 관리자 간에 메시지 플로우를 분리시킬 수 있습니다. 다른 클러스터 송신자 채널에서 전송되는 메시지를 상이한 클러스터 전송 큐에 둘 수 있습니다. 단일 클러스터 또는 중첩 클러스터에 이 접근법을 사용할 수 있습니다. 이 주제는 사용할 접근법을 선택하는 데 유용한 예와 몇 가지 우수 사례를 제공합니다.

관련 태스크

ClusterQueueAccessControl 설정

클러스터링과 분산 큐잉의 비교

분산 큐잉 및 클러스터링을 사용하여 큐 관리자를 연결하기 위해 정의해야 하는 컴포넌트를 비교합니다.

클러스터를 사용하지 않는 경우 큐 관리자는 독립적이며 분산 큐잉을 사용하여 통신합니다. 한 큐 관리자가 다른 큐 관리자에 메시지를 보내야 하는 경우 다음을 정의해야 합니다.

- 전송 큐
- 리모트 큐 관리자로 향하는 채널

27 페이지의 그림 3에서는 분산 큐잉에 필요한 컴포넌트를 보여줍니다.

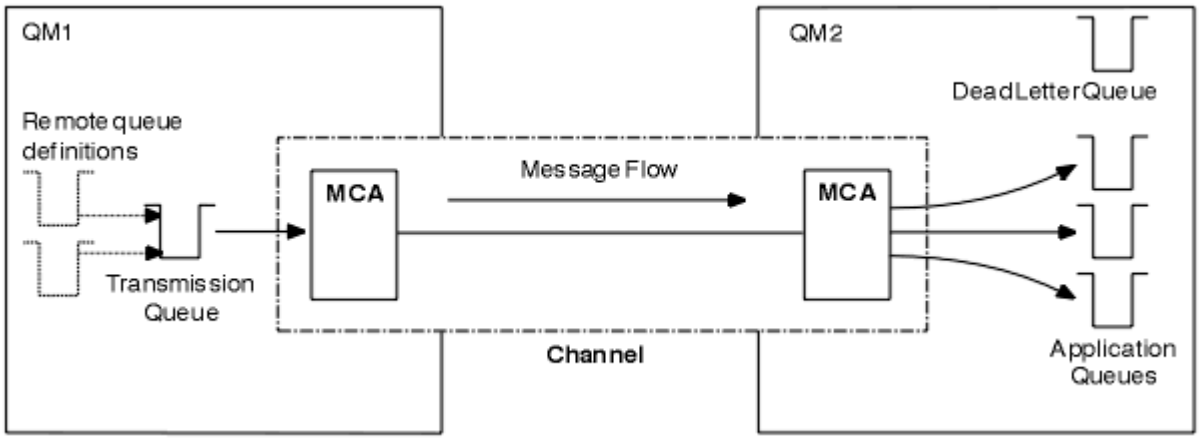


그림 *3.* 분산 큐잉

클러스터의 큐 관리자를 그룹화할 경우 클러스터의 다른 모든 큐 관리자가 큐 관리자의 큐를 사용할 수 있습니다. 어느 큐 관리자나 명확한 정의 없이도 동일한 클러스터의 다른 큐 관리자에 메시지를 보낼 수 있습니다. 각 목적지에 대한 채널 정의, 리모트 큐 정의 또는 전송 큐는 제공하지 않습니다. 클러스터의 모든 큐 관리자마다 클러스터의 다른 큐 관리자에 메시지를 전송할 수 있는 단일 전송 큐가 있습니다. 클러스터의 각 큐 관리자는 다음 항목만 정의하면 됩니다.

- 메시지를 수신할 하나의 클러스터 수신자 채널
- 자신을 소개하고 클러스터에 대해 배우는 하나의 클러스터 송신자 채널

클러스터 대 분산 큐잉 설정을 위한 정의

각각 2개의 큐가 포함된 4개의 큐 관리자를 보여주는 28 페이지의 그림 4를 살펴보십시오. 분산 큐잉을 사용하여 이러한 큐 관리자를 연결하기 위해 필요한 정의의 수를 고려하십시오. 클러스터로서 동일한 네트워크를 설정하는 데 필요한 정의의 수를 비교하십시오.

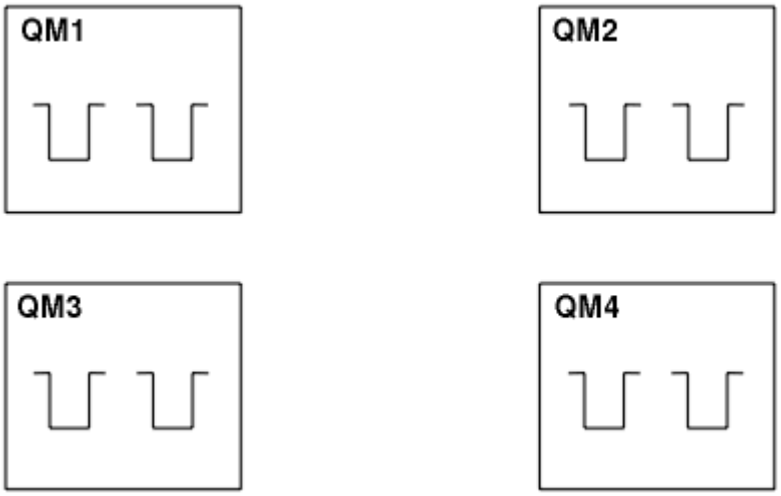


그림 *4.* 네 개의 큐 관리자 네트워크

분산 큐잉을 사용하여 네트워크를 설정하기 위한 정의

분산 큐잉을 사용하여 27 페이지의 그림 3에 표시된 네트워크를 설정하려면 다음과 같은 정의가 필요할 수 있습니다.

표 *2.* 분산 큐잉에 대한 정의		
설명	**큐 관리자당 개수**	**총 수**
다른 모든 큐 관리자에 메시지를 보낼 채널에 대한 송신자 채널 정의	3	12
다른 모든 큐 관리자로부터 메시지를 받을 채널에 대한 수신자 채널 정의	3	12
다른 모든 큐 관리자로 향하는 전송 큐에 대한 전송 큐 정의	3	12
각 로컬 큐에 대한 로컬 큐 정의	2	8
이 큐 관리자가 메시지를 넣을 각 리모트 큐에 대한 리모트 큐 정의	6	24

일반 수신자 채널 정의를 사용하여 이 정의의 수를 줄일 수 있습니다. 최대 정의 수는 각 큐 관리자당 17개를 초과할 수 없으며 이 네트워크에서는 총 68입니다.

클러스터를 사용하여 네트워크를 설정하기 위한 정의

클러스터를 사용하여 27 페이지의 그림 3에 표시된 네트워크를 설정하려면 다음과 같은 정의가 필요할 수 있습니다.

표 *3.* 클러스터링에 대한 정의		
설명	**큐 관리자당 개수**	**총 수**
저장소 큐 관리자에 메시지를 보낼 채널에 대한 클러스터 송신자 채널 정의	1	4
클러스터의 다른 큐 관리자로부터 메시지를 받을 채널에 대한 클러스터 수신자 채널 정의	1	4
각 로컬 큐에 대한 로컬 큐 정의	2	8

큐 관리자의 이 클러스터(전체 저장소 2개 포함)를 설정하려면 큐 관리자당 4개의 정의가 필요하며 총 16개의 정의입니다. 두 개의 큐 관리자의 경우에는 클러스터의 전체 저장소 큐 관리자로 지정하기 위해 큐 관리자 정의를 대체해야 합니다.

하나의 CLUSSDR 및 하나의 CLUSRCVR 채널 정의만 필요합니다. 클러스터가 정의될 때 다른 저장소 큐 관리자를 방해하지 않고 큐 관리자(저장소 큐 관리자 제외)를 추가하거나 제거할 수 있습니다.

클러스터를 사용하면 여러 큐 관리자를 포함한 네트워크를 설정하는 데 필요한 정의 수가 감소합니다.

필요한 정의 수가 더 적으므로 다음과 같은 오류가 발생할 위험도 낮습니다.

- 오브젝트 이름이 항상 일치합니다(예를 들어, 송신자-수신자 쌍의 채널 이름).
- 채널 정의에 지정된 전송 큐 이름이 항상 올바른 전송 큐 정의 또는 전송 큐 정의에 지정된 전송 큐 이름과 일치합니다.
- QREMOTE 정의가 항상 리모트 큐 관리자의 올바른 큐를 가리킵니다.

클러스터가 일단 설정되면 다른 큐 관리자에서 시스템 관리 작업을 수행할 필요 없이 클러스터의 한 큐 관리자에서 다른 큐 관리자로 클러스터 큐를 이동시킬 수 있습니다. 채널, 리모트 큐 또는 전송 큐 정의를 삭제하거나 수정하는 것을 잊어버릴 가능성이 없습니다. 기존 네트워크를 방해하지 않고 새 큐 관리자를 클러스터에 추가할 수 있습니다.

전체 저장소를 보유할 클러스터 큐 관리자 선택 방법

각 클러스터에서 전체 저장소를 보유할 최소 하나, 가급적이면 두 개의 큐 관리자를 선택해야 합니다. 가장 예외적 상황을 제외하면 두 개의 전체 저장소가 충분합니다. 가능하면 강력하고 영구적으로 연결된 플랫폼에서 호스트되고, 동시 가동 중단이 없으며, 지리적으로 중앙에 위치한 큐 관리자를 선택하십시오. 또한 전체 저장소 호스트로 사용할 시스템을 다른 태스크에 사용하지 않고 전용 시스템으로 사용할 것을 고려하십시오.

전체 저장소는 클러스터 상태의 전체 그림을 유지보수하는 큐 관리자입니다. 이 정보를 공유하기 위해 각 전체 저장소는 CLUSSDR 채널(및 해당 CLUSRCVR 정의)을 통해 클러스터의 다른 모든 전체 저장소에 연결됩니다. 이 채널을 수동으로 정의해야 합니다.

그림 *5.* 연결된 두 개의 전체 저장소.

클러스터의 다른 모든 큐 관리자는 부분 저장소에서 클러스터 상태에 대해 현재 알고 있는 사항을 유지합니다. 이 큐 관리자는 사용 가능한 두 개의 전체 저장소를 사용하여 자신에 대한 정보를 발행하고 다른 큐 관리자에 대한 정보를 요청합니다. 선택된 전체 저장소가 사용 불가능하면 다른 전체 저장소가 사용됩니다. 선택된 전체 저장소가 다시 사용 가능하게 되면 보조를 맞추기 위해 다른 큐 관리자로부터 새로운 최신 정보와 변경된 정보를 수집합니다. 모든 전체 저장소가 서비스 불가능 상태가 되면 다른 큐 관리자는 부분 저장소의 정보를 사용합니다. 하지만 보유한 정보만을 사용하도록 제한되며 새 정보 및 업데이트에 대한 요청은 처리 불가능합니다. 전체 저장소가 네트워크에 다시 연결되면 메시지가 교환되어 모든 저장소(전체 및 부분 모두)는 최신 상태가 됩니다.

전체 저장소 할당을 계획할 때에는 다음을 고려하십시오.

- 전체 저장소를 보유하도록 선택한 큐 관리자는 신뢰할 수 있으며 관리되어야 합니다. 강력하고 영구적으로 연결된 플랫폼에서 호스트되는 큐 관리자를 선택하십시오.
- 전체 저장소를 호스트하는 시스템의 계획된 가동 중단을 고려하고 동시 가동 중단이 없는지 확인하십시오.
- 네트워크 성능을 고려하십시오. 지리적으로 중앙에 위치하거나 클러스터의 다른 큐 관리자와 동일한 시스템을 공유하는 큐 관리자를 선택하십시오.
- 큐 관리자가 둘 이상 클러스터의 멤버인지 여부를 고려하십시오. 동일한 큐 관리자를 사용하여 여러 클러스터에 대한 전체 저장소를 호스트하는 것이 관리상 편리할 수 있습니다. 단 이는 큐 관리자가 얼마나 바쁠지 예상한 바와 이 장점이 밸런스를 이룰 때입니다.
- 전체 저장소만을 포함한 일부 시스템을 다른 태스크에 사용하지 않고 전용 시스템으로 사용할 것을 고려하십시오. 그러면 이 시스템은 큐 관리자 구성에 대한 유지보수만을 수행하며 기타 비즈니스 애플리케이션의 유지보수를 위해 서비스에서 제거되지 않습니다. 또한 저장소를 유지보수하는 태스크가 시스템 자원을 두고 애플리케이션과 경쟁하지 않습니다. 이는 특히, 전체 저장소에 클러스터 상태를 유지보수하는 워크로드가 훨씬 많은 대형 클러스터(즉, 수천 개 이상의 큐 관리자를 포함한 클러스터)에 유익할 수 있습니다.

셋 이상의 전체 저장소도 가능하지만 거의 권장되지 않습니다. 오브젝트 정의(즉, 큐, 토픽, 채널)는 사용 가능한 모든 전체 저장소로 플로우되지만 요청은 부분 저장소에서 최대 두 개의 전체 저장소로만 플로우됩니다. 이는 셋 이상의 전체 저장소가 정의되고 두 개의 전체 저장소가 사용 불가능하게 되면 일부 부분 저장소가 예상하는 업데이트를 수신하지 못할 수 있음을 의미합니다. MQ 클러스터: 왜 전체 저장소는 두 개뿐입니까?를 참조하십시오.

셋 이상의 전체 저장소를 정의하는 것이 유용할 수 있는 유일한 상황은 기존 전체 저장소를 새 하드웨어나 새 큐 관리자로 마이그레이션할 때입니다. 이 경우에는 대체 전체 저장소를 도입하고 이전 전체 저장소를 제거하기 전에 이 저장소가 완전히 채워졌는지 확인해야 합니다. 전체 저장소를 추가할 때마다 CLUSSDR 채널로 다른 모든 전체 저장소에 이 전체 저장소를 직접 연결해야 함을 기억하십시오.

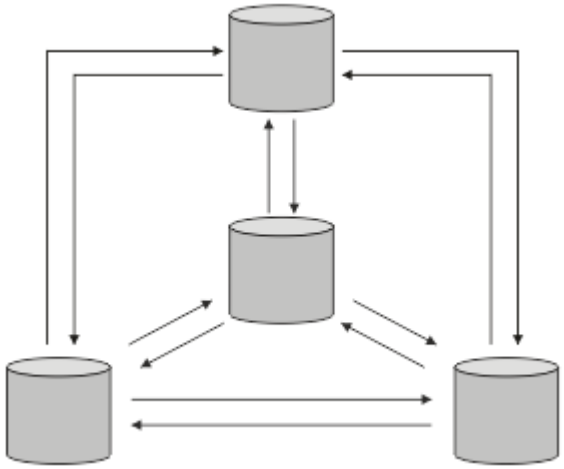


그림 6. 연결된 셋 이상의 전체 저장소

관련 정보

MQ 클러스터: 왜 전체 저장소는 두 개뿐입니까?
MQ 클러스터의 가능한 크기

클러스터 조직화

전체 저장소로 링크할 큐 관리자를 선택하십시오. 성능 영향, 큐 관리자 버전, 다수의 CLUSSDR 채널을 원하는지 여부를 고려하십시오.

전체 저장소를 보유할 큐 관리자를 선택했으면 전체 저장소로 링크할 큐 관리자를 결정해야 합니다. CLUSSDR 채널 정의는 클러스터의 다른 전체 저장소를 알아내는 전체 저장소로 큐 관리자를 링크합니다. 이 때부터 큐 관리자는 두 개의 전체 저장소로 메시지를 보냅니다. 항상 CLUSSDR 채널 정의가 있는 전체 저장소를 먼저 사용하려 시도합니다. 큐 관리자를 두 전체 저장소 중 하나로 링크하도록 선택할 수 있습니다. 선택 시 구성의 토폴로지와 큐 관리자의 물리적 또는 지리적 위치를 고려하십시오.

모든 클러스터 정보가 두 개의 전체 저장소에 송신되기 때문에 두 번째 CLUSSDR 채널 정의를 작성하려는 경우가 있을 수 있습니다. 광범위한 영역에 많은 전체 저장소가 분산된 클러스터에 두 번째 CLUSSDR 채널을 정의할 수 있습니다. 그런 다음 정보가 송신되는 두 개의 전체 저장소를 제어할 수 있습니다.

클러스터 이름 지정 규칙

큐 관리자가 속한 클러스터를 식별하는 이름 지정 규칙을 사용하여 동일한 클러스터의 큐 관리자 이름을 지정할 것을 고려하십시오. 채널 이름에 유사한 이름 지정 규칙을 사용하고 채널 특성을 설명하도록 확장하십시오.

MQ 클러스터 이름 지정 시 우수 사례

클러스터 이름은 최대 48자까지 가능하지만 상대적으로 짧은 클러스터 이름은 다른 오브젝트에 이름 지정 규칙을 적용할 때 유용합니다. 31 페이지의 『클러스터 채널 이름 선택 시 우수 사례』의 내용을 참조하십시오.

클러스터 이름을 선택할 때 일반적으로 '컨텐츠' 가 아닌 클러스터의 '목적' (수명이 길 수 있음) 을 표시하는 것이 유용합니다. 예를 들어, 'QM1_QM2_QM3_CLUS' 대신 'B2BPROD' 또는 'ACTTEST' 입니다.

클러스터 큐 관리자 이름 선택 시 우수 사례

처음부터 새 클러스터 및 해당 멤버를 작성하는 경우 해당 클러스터 사용량을 반영하는 큐 관리자에 대한 이름 지정 규칙을 고려하십시오. 모든 큐 관리자의 이름이 서로 달라야 합니다. 그러나 논리 그룹을 식별하고 기억하는 데 도움이 되도록 클러스터의 큐 관리자에 유사한 이름 세트를 제공할 수 있습니다 (예: 'ACTTQM1, ACTTQM2).

상대적으로 짧은 큐 관리자 이름 (예: 8자미만) 은 다음 절에 설명된 규칙을 사용하거나 채널 이름에 대해 유사한 규칙을 사용하도록 선택하는 경우에 도움이 됩니다.

클러스터 채널 이름 선택 시 우수 사례

큐 관리자 및 클러스터는 최대 48자의 이름을 가질 수 있고 채널 이름은 20자로 제한되므로 프로젝트를 통해 중간에 이름 지정 규칙을 변경하지 않도록 오브젝트의 이름을 처음 지정할 때 주의하십시오 (이전 절 참조).

채널을 정의할 때 클러스터에 있는 임의의 큐 관리자에서 자동으로 작성된 클러스터 송신자 채널은 클러스터의 수신 큐 관리자에 구성된 해당 클러스터 수신자 채널에서 해당 이름을 사용하므로 이러한 채널은 고유해야 하며 클러스터의 리모트 큐 관리자에서 의미가 있어야 합니다.

한 가지 일반적인 방법은 클러스터 이름이 앞에 오는 큐 관리자 이름을 사용하는 것입니다. 예를 들어, 클러스터 이름이 CLUSTER1이고 큐 관리자가 QM1, QM2이면 클러스터 수신자 채널은 CLUSTER1.QM1, CLUSTER1.QM2 입니다.

채널에 다른 우선순위가 있거나 다른 프로토콜을 사용하는 경우 이 규칙을 확장할 수 있습니다. 예를 들면, 다음과 같습니다.

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

이 예에서 S1 는 첫 번째 SNA 채널일 수 있고, N3 는 네트워크 우선순위가 3인 NetBIOS 채널일 수 있으며, T4 는 IPV4 네트워크를 사용하는 TCP IP일 수 있습니다.

공유 채널 정의 이름 지정

단일 채널 정의를 여러 클러스터에서 공유할 수 있습니다. 이 경우 여기에 제안된 이름 지정 규칙을 수정해야 합니다. 그러나 채널 정의 관리 에 설명된 대로 일반적으로 어떤 경우에도 각 클러스터에 대해 개별 채널을 정의하는 것이 좋습니다.

이전 채널 이름 지정 규칙

클러스터 환경 외부에서는 'FROMQM.TO.TARGETQM' 이름 지정 규칙을 사용하는 것이 일반적이므로 기존 클러스터에서 유사한 규칙 (예: CLUSTER.TO.TARGET) 을 사용했음을 알 수 있습니다. 이는 채널 이름 내에서 '유용한' 정보를 전달하기 위해 사용 가능한 문자를 더 줄이기 때문에 새 클러스터 이름 지정 스킴의 일부로 권장되지 않습니다.

z/OS IBM MQ for z/OS 의 채널 이름

VTAM 일반 자원 또는 DDNS (*Dynamic Domain Name Server*) 일반 이름을 정의할 수 있습니다. 일반 이름을 사용하여 연결 이름을 정의할 수 있습니다. 하지만 클러스터 수신자 정의를 작성할 때에는 일반 연결 이름을 사용하지 마십시오.

클러스터 수신자 정의에 일반 연결 이름을 사용하는 문제점은 다음과 같습니다. 일반 CONNAME 로 CLUSRCVR 를 정의하는 경우 CLUSSDR 채널이 사용자가 원하는 큐 관리자를 가리킨다는 보장이 없습니다. 초기 CLUSSDR은 반드시 전체 저장소를 호스팅하는 큐 관리자가 아닌, 결국 큐 공유 그룹의 큐 관리자를 가리킬 수 있습니다. 채널이 연결을 다시 시도하기 시작하면 동일한 일반 이름을 가진 다른 큐 관리자에 다시 연결하여 메시지 플로우를 방해할 수 있습니다.

z/OS *Queue sharing groups and clusters*

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

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

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

VTAM 일반 자원 또는 DDNS (*Dynamic Domain Name Server*) 일반 이름을 정의할 수 있습니다. 일반 이름을 사용하여 연결 이름을 정의할 수 있습니다. 하지만 클러스터 수신자 정의를 작성할 때에는 일반 연결 이름을 사용하지 마십시오.

클러스터 수신자 정의에 일반 연결 이름을 사용하는 문제점은 다음과 같습니다. 일반 CONNAME 로 CLUSRCVR 를 정의하는 경우 CLUSSDR 채널이 사용자가 원하는 큐 관리자를 가리킨다는 보장이 없습니다. 초기 CLUSSDR은 반드시 전체 저장소를 호스팅하는 큐 관리자가 아닌, 결국 큐 공유 그룹의 큐 관리자를 가리킬 수 있습니다. 채널이 연결을 다시 시도하기 시작하면 동일한 일반 이름을 가진 다른 큐 관리자에 다시 연결하여 메시지 플로우를 방해할 수 있습니다.

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

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

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

중첩 클러스터

중첩 클러스터는 추가 관리 기능을 제공합니다. 이름 목록을 사용하여 중첩 클러스터를 관리하는 데 필요한 명령의 수를 줄이십시오.

중첩되는 클러스터를 작성할 수 있습니다. 중첩 클러스터를 정의할 수 있는 많은 이유가 있습니다. 예를 들어, 다음과 같습니다.

- 서로 다른 조직이 각자 고유의 관리를 할 수 있도록 하기 위해
- 독립 애플리케이션을 개별적으로 관리할 수 있도록 하기 위해
- 서비스 클래스를 작성하기 위해

33 페이지의 그림 7에서 STF2 큐 관리자는 두 클러스터 모두의 멤버입니다. 한 큐 관리자가 둘 이상의 클러스터의 멤버이면 이름 목록을 사용하여 필요한 정의 수를 줄일 수 있습니다. 이름 목록은 예를 들어, 클러스터 이름 등의 이름 목록을 포함합니다. 클러스터의 이름을 지정하는 이름 목록을 작성할 수 있습니다. STF2에 대한 ALTER QMGR 명령에 이름 목록을 지정하여 두 클러스터 모두에 대한 전체 저장소 큐 관리자로 지정하십시오.

네트워크에 둘 이상의 클러스터가 있으면 각각 다른 이름을 지정해야 합니다. 동일한 이름의 두 클러스터가 병합되는 경우 다시 클러스터를 분리할 수 없습니다. 클러스터 및 채널에 서로 다른 이름을 지정하는 것도 좋은 방법입니다. DISPLAY 명령의 출력을 보면 더 쉽게 구별됩니다. 큐 관리자가 제대로 작동하려면 클러스터 내에서 큐 관리자 이름이 고유해야 합니다.

서비스 클래스 정의

각 직원 구성원과 각 학생별로 하나의 큐 관리자가 있는 대학교를 상상해 보십시오. 직원 구성원 간의 메시지는 우선순위가 높은 고대역폭의 채널로 이동합니다. 학생 간의 메시지는 보다 저렴하고 느린 채널에서 이동합니다. 일반적인 분산 큐잉 기술을 사용하여 이 네트워크를 설정할 수 있습니다. IBM MQ는 목적지 큐와 큐 관리자 이름을 보고 사용할 채널을 선택합니다.

직원과 학생을 명확하게 구분하기 위해 33 페이지의 그림 7에 표시된 대로 큐 관리자를 두 개의 클러스터로 그룹화할 수 있습니다. IBM MQ는 이 클러스터에 정의된 채널을 통해서만 Staff 클러스터의 meetings 큐로 메시지를 이동시킵니다. Students 클러스터의 gossip 큐 메시지는 이 클러스터에 정의된 채널을 통해 이동하고 적절한 서비스 클래스를 수신합니다.

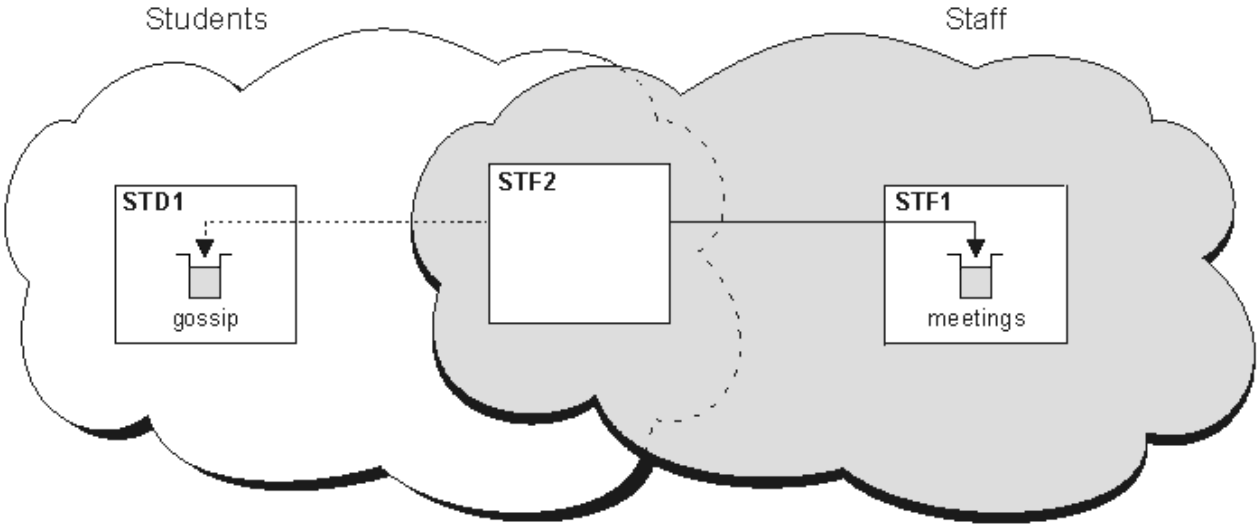


그림 7. 서비스 클래스

클러스터링 팁

클러스터링을 사용하기 전에 시스템 또는 애플리케이션을 약간 수정해야 할 수 있습니다. 분산 큐잉 작동의 차이점 및 유사성이 모두 존재합니다.

- 이러한 큐 관리자가 클러스터 큐에 액세스하도록 클러스터 외부의 큐 관리자에 수동 구성 정의를 추가해야 합니다.
- 동일한 이름의 두 클러스터를 병합하는 경우 이 클러스터를 다시 분리할 수 없습니다. 따라서 모든 클러스터에 고유 이름을 지정하는 것이 좋습니다.
- 메시지가 큐 관리자에 도달하지만 수신할 큐가 없으면 데드-레터 큐에 메시지를 넣습니다. 데드-레터 큐가 없으면 채널이 실패하고 다시 시도합니다. 데드-레터 큐의 사용은 분산 큐잉의 경우와 동일합니다.
- 지속 메시지의 무결성이 유지됩니다. 클러스터 사용으로 인해 메시지가 복제 또는 손실되지 않습니다.
- 클러스터를 사용하면 시스템 관리가 감소합니다. 클러스터는 분산 큐잉을 사용하여 고려할 수 있는 것보다 더 많은 큐 관리자가 있는 보다 큰 네트워크에 쉽게 연결합니다. 클러스터의 모든 큐 관리자 간에 통신을 사용하려 시도하는 경우에는 과도한 네트워크 자원을 소모할 위험이 있습니다.
- 트리 구조로 큐 관리자를 표시하는 IBM MQ Explorer를 사용하면 큰 클러스터의 보기가 번잡할 수 있습니다.
- Multi 분배 목록의 용도는 단일 MQPUT 명령을 사용하여 여러 목적지에 동일한 메시지를 보내기 위해서입니다. 분배 목록은 IBM MQ for Multiplatforms에서 지원됩니다. 분배 목록을 큐 관리자 클러스터와 함께 사용할 수 있습니다. 클러스터에서 모든 메시지는 MQPUT 시에 확장됩니다. 네트워크 트래픽의 관점에서 보면 비클러스터링 환경에서처럼 장점이 그리 많지 않습니다. 분배 목록의 장점은 수많은 채널과 전송 큐를 수동으로 정의할 필요가 없다는 점입니다.
- 클러스터를 사용하여 워크로드 밸런싱을 조절하려는 경우 애플리케이션을 검사하십시오. 특정 큐 관리자로 또는 특정 순서로 메시지를 처리해야 하는지 확인하십시오. 이러한 애플리케이션은 메시지 연관관계가 있습니다. 복잡한 클러스터에서 애플리케이션을 사용하려면 먼저 수정해야 할 수 있습니다.
- 특정 목적지로 메시지를 강제로 보내기 위해 MQOPEN에 MQOO_BIND_ON_OPEN 옵션 사용을 선택할 수 있습니다. 목적지 큐 관리자가 사용 불가능하면 큐 관리자가 다시 사용 가능하게 될 때까지 메시지가 전달되지 않습니다. 복제 위험으로 인해 다른 큐 관리자로 메시지가 라우팅되지 않습니다.
- 큐 관리자가 클러스터 저장소를 호스팅할 경우에는 호스트 이름 또는 IP 주소를 알고 있어야 합니다. 클러스터를 조인하는 기타 큐 관리자에 대한 CLUSSDR 정의를 작성할 때 CONNAME 매개변수에 이 정보를 지정해야 합니다. DHCP를 사용하면 시스템을 재시작할 때마다 DHCP가 새 IP 주소를 할당할 수 있어서 IP 주소가 변경되기 쉽습니다. 따라서 CLUSSDR 정의에 IP 주소를 지정하면 안됩니다. 모든 CLUSSDR 정의가 IP 주소가 아닌 호스트 이름을 지정해도 정의는 여전히 신뢰할 수 없습니다. DHCP는 호스트의 DNS 디렉토리 입력 항목을 반드시 새 주소로 업데이트하지는 않습니다. DHCP를 사용하는 시스템의 전체 저장소로 큐 관리자를 지정해야 하는 경우에는 DNS 디렉토리를 최신 상태로 유지함이 보장되는 소프트웨어를 설치하십시오.

- VTAM 일반 자원 또는 DDNS(Dynamic Domain Name Server) 일반 이름과 같은 일반 이름을 채널의 연결 이름으로 사용하지 마십시오. 그러할 경우 채널이 예상과 다른 큐 관리자에 연결할 수 있습니다.
- 로컬 클러스터 큐에서만 메시지를 가져올 수 있지만 메시지 넣기는 클러스터의 어느 큐에나 가능합니다. MQGET 명령을 사용하기 위해 큐를 여는 경우 큐 관리자는 로컬 큐를 엽니다.
- 단순 IBM MQ 클러스터를 설정하면 애플리케이션을 변경할 필요가 없습니다. 애플리케이션이 MQOPEN 호출에 대상 큐의 이름을 지정할 수 있으며 큐 관리자의 위치에 대해 알 필요가 없습니다. 워크로드 관리를 위한 클러스터를 설정하면 애플리케이션을 검토하고 필요에 따라 수정해야 합니다.
- DISPLAY CHSTATUS 및 DISPLAY QSTATUS **runmqsc** 명령을 사용하여 채널이나 큐에 대한 현재 모니터링 및 상태 데이터를 볼 수 있습니다. 모니터링 정보를 사용하여 시스템의 성능 및 상태를 파악할 수 있습니다. 모니터링은 큐 관리자, 큐, 채널 속성으로 제어됩니다. 자동 정의된 클러스터 송신자 채널의 모니터링은 MONACLS 큐 관리자 속성으로 가능합니다.

관련 개념

클러스터

27 페이지의 『클러스터링과 분산 큐잉의 비교』

분산 큐잉 및 클러스터링을 사용하여 큐 관리자를 연결하기 위해 정의해야 하는 컴포넌트를 비교합니다.

클러스터의 컴포넌트

관련 태스크

큐 관리자 클러스터 구성

새 클러스터 설정

큐 관리자 저장소가 정보를 보유하는 기간

큐 관리자 저장소는 정보를 30일 동안 보유합니다. 자동 프로세스가 효율적으로 사용 중인 정보를 새로 고칩니다.

큐 관리자가 자체적인 일부 정보를 전송하는 경우 전체 및 부분 저장소 큐 관리자가 정보를 30일 동안 저장합니다. 정보는 예를 들어 큐 관리자가 새 큐 작성을 광고하는 경우에 전송됩니다. 이 정보가 만기되지 않도록 하기 위해 큐 관리자는 27일 후에 자체에 대한 모든 정보를 자동으로 다시 전송합니다. 부분 저장소가 30일의 지속 기간 동안 정보 파트에 대한 새 요청을 전송하면 만기 시간이 원래 30일로 유지됩니다.

정보가 만기되면 저장소에서 즉시 제거되지는 않습니다. 대신, 60일의 유예 기간 동안 보유됩니다. 유예 기간 중에 업데이트가 수신되지 않으면 정보가 제거됩니다. 유예 기간을 사용하면 큐 관리자가 만기 날짜에 임시로 서비스되지 않을 수도 있습니다. 큐 관리자가 90일 이상 클러스터에 연결되지 않으면 더 이상 클러스터의 파트가 아닙니다. 그렇지만 네트워크에 다시 연결되는 경우 다시 클러스터의 파트가 됩니다. 전체 저장소는 다른 큐 관리자의 새 요청을 충족시키기 위해 만기된 정보를 사용하지 않습니다.

유사하게 큐 관리자가 전체 저장소에서의 최신 정보에 대한 요청을 전송하면 요청이 30일 동안 계속됩니다. 27일 후에 IBM MQ는 요청을 검사합니다. 27일 동안 참조된 경우에는 자동으로 새로 고쳐집니다. 그렇지 않은 경우, 만기된 상태가 되고 다시 필요하게 될 때 큐 관리자가 이를 새로 고칩니다. 요청 만기를 통해 휴면 중인 큐 관리자의 정보 요청이 증가되는 것을 방지합니다.

참고: APAR PH43191의 PTF를 다운로드하여 설치해야 합니다. 이는 등록의 만기 시간을 계산할 때 시스템 오류를 수정합니다. 이러한 오류로 인해 구독이 조기에 만료되거나 (CSQX456I 메시지가 발행됨) 오브젝트가 만료된 후에 만료될 수 있습니다 (오류가 있는 MQRC 2085 (MQRC_UNKNOWN_OBJECT) 오류가 발생함).

대형 클러스터인 경우, 동시에 자체적인 모든 정보를 다수의 큐 관리자가 자동으로 재전송하는 경우 문제가 될 수 있습니다. 대형 클러스터를 새로 고치면 클러스터의 성능 및 가용성에 영향을 줄 수 있음을 참조하십시오.

관련 개념

61 페이지의 『클러스터링: REFRESH CLUSTER 사용 우수 사례』

REFRESH CLUSTER 명령을 사용하여 로컬에 보유된 클러스터에 대한 모든 정보를 제거하고 클러스터의 전체 저장소에서 이 정보를 다시 빌드합니다. 예외 상황을 제외하고는 이 명령을 사용해서는 안됩니다. 사용해야 하는 경우 사용 방법에 대한 특수 고려사항이 있습니다. 이 정보는 고객의 피드백과 테스트에 기반을 둔 지침입니다.

클러스터 예

첫 번째 예는 두 가지 큐 관리자의 가능한 최소 클러스터를 보여줍니다. 두 번째와 세 번째 예는 세 가지 큐 관리자 클러스터의 두 개 버전을 보여줍니다.

가능한 최소 클러스터는 두 개의 큐 관리자만 포함합니다. 이 경우에는 두 큐 관리자가 모두 전체 저장소를 보유합니다. 클러스터를 설정하기 위해 몇 개의 정의만 필요하며 각 큐 관리자의 자율도가 아직은 높습니다.

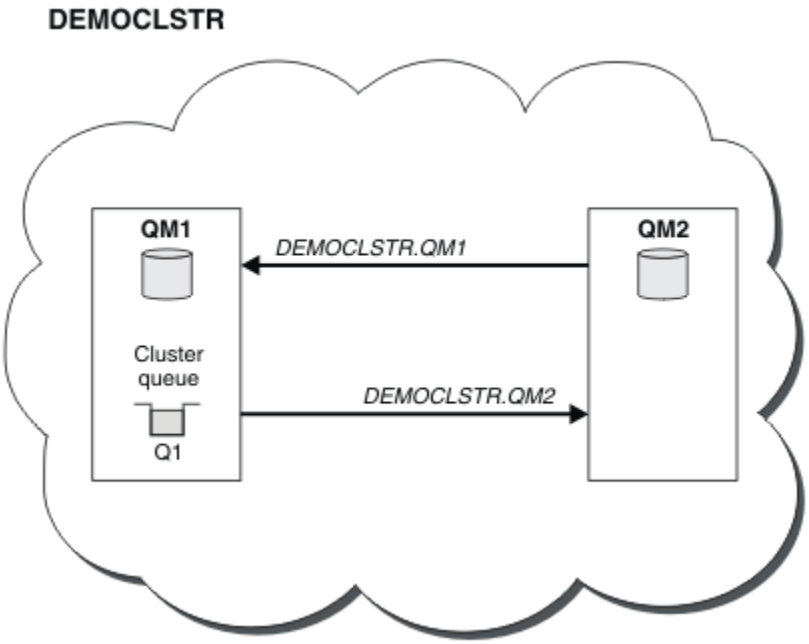


그림 *8.* 두 큐 관리자의 작은 클러스터

- 큐 관리자는 긴 이름(예: LONDON 및 NEWYORK)을 가질 수 있습니다. z/OS IBM MQ for z/OS에서 큐 관리자 이름은 4자로 제한됩니다.
- 각 큐 관리자는 일반적으로 별도의 시스템에 구성됩니다. 하지만 동일한 시스템에 여러 큐 관리자를 보유할 수 있습니다.

유사한 예 클러스터 설정에 대한 지시사항은 새 클러스터 설정을 참조하십시오.

36 페이지의 그림 9에서는 CLSTR1(이)라는 클러스터의 구성요소를 표시합니다.

- 이 클러스터에는 세 개의 큐 관리자(QM1, QM2 및 QM3)가 있습니다.
- QM1 및 QM2에서는 클러스터의 모든 큐 관리자 및 클러스터 관련 오브젝트에 대한 정보의 저장소를 호스팅합니다. 이를 전체 저장소 큐 관리자라 부릅니다. 저장소는 다이어그램에서 음영이 있는 실린더로 표시됩니다.
- QM2 및 QM3에서는 클러스터의 다른 큐 관리자에 액세스할 수 있는 일부 큐를 호스팅합니다. 클러스터의 다른 큐 관리자에 액세스할 수 있는 큐를 클러스터 큐라 합니다. 클러스터 큐는 다이어그램에서 음영이 있는 큐로 표시됩니다. 클러스터 큐는 클러스터의 어디에서나 액세스할 수 있습니다. IBM MQ 클러스터링 코드는 클러스터 큐에 대한 리모트 큐 정의가 이 정의를 참조하는 모든 큐 관리자에 작성되도록 합니다.

  분산 큐잉과 마찬가지로 애플리케이션은 MQPUT 호출을 사용하여 클러스터의 모든 큐 관리자에 있는 클러스터 큐에 메시지를 넣습니다. 애플리케이션은 MQGET 호출을 사용하여 큐가 상주하는 큐 관리자의 클러스터 큐에서만 메시지를 검색합니다.
- 각 큐 관리자에는 메시지를 수신할 수 있는 *cluster_name. queue_manager_name*이라는 채널의 수신 측에 대해 수동으로 작성된 정의가 있습니다. 수신 큐 관리자에서 *cluster_name. queue_manager_name*은 클러스터 수신자 채널입니다. 클러스터 수신자 채널은 분산 큐잉에서 사용되는 수신자 채널과 유사하며 큐 관리자에 대한 메시지를 수신합니다. 클러스터에 관한 정보도 수신합니다.

•

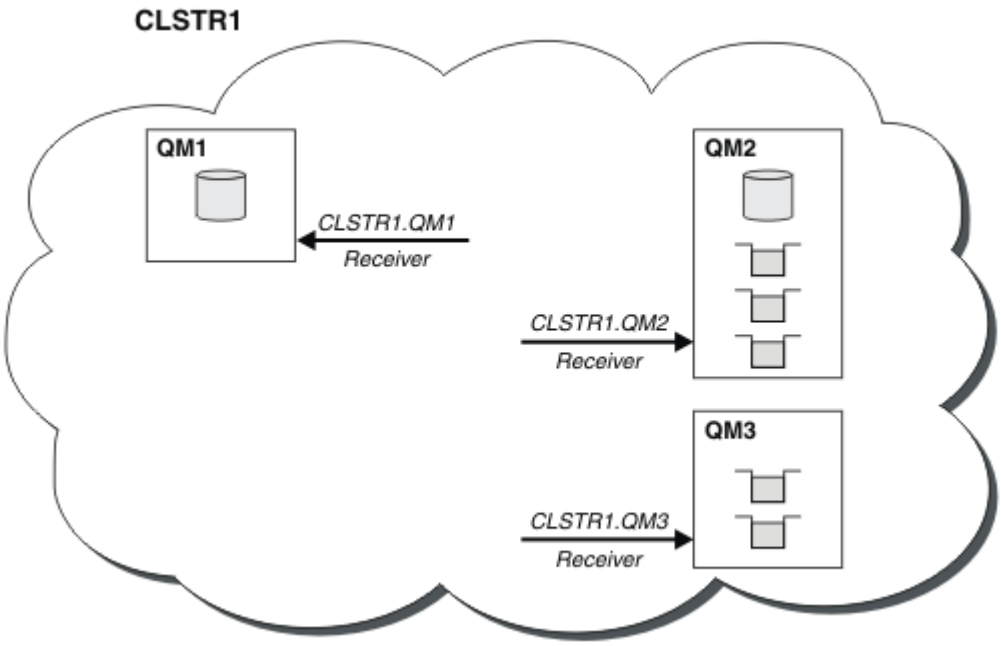


그림 *9.* 큐 관리자의 클러스터

- 36 페이지의 그림 10에는 각 큐 관리자마다 채널의 송신 측에 대한 정의도 있습니다. 각 큐 관리자는 전체 저장소 큐 관리자 중 하나의 클러스터 수신자 채널에 연결합니다. 송신 큐 관리자에서 *cluster_name.queue_manager_name*은(는) 클러스터 전송자 채널입니다. QM1 및 QM3에 CLSTR1.QM2에 연결하는 클러스터 송신자 채널이 있습니다(점선 "2" 참조).

  QM2에는 CLSTR1.QM1에 연결하는 클러스터 송신자 채널이 있습니다(점선 "3" 참조). 클러스터-송신자 채널은 분산 큐잉에서 사용되는 송신자 채널과 비슷하며, 수신 큐 관리자로 메시지를 송신합니다. 클러스터에 관한 정보도 보냅니다.

  클러스터 수신자 채널의 끝과 클러스터 송신자 채널의 끝이 모두 정의되고 나면 채널이 자동으로 시작됩니다.

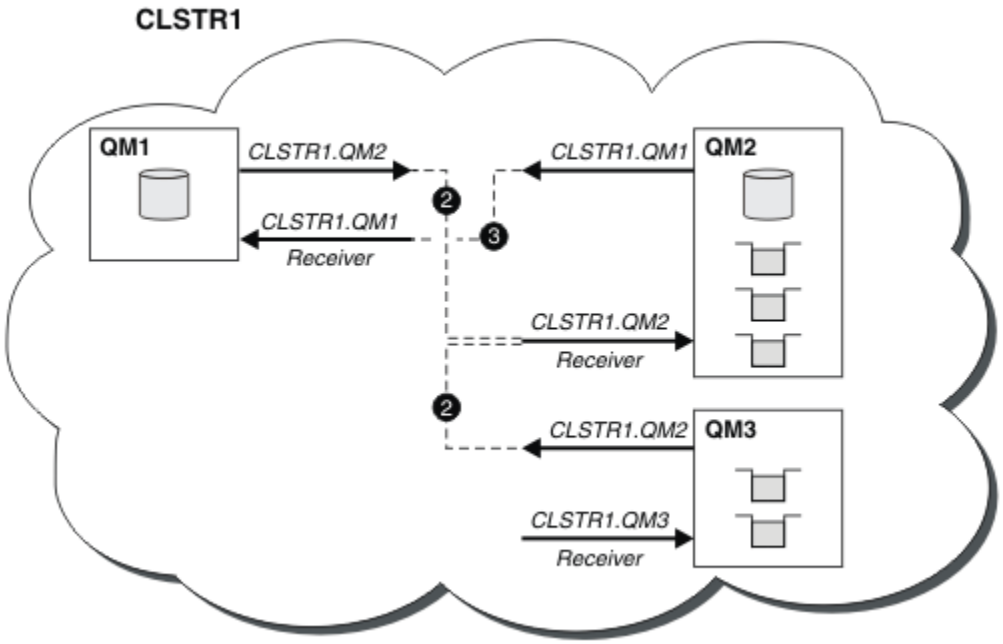


그림 *10.* 송신자 채널이 있는 큐 관리자의 클러스터

로컬 큐 관리자에 클러스터 송신자 채널을 정의하면 해당 큐 관리자가 전체 저장소 큐 관리자 중 하나에 도입됩니다. 전체 저장소 큐 관리자는 그에 따라 전체 저장소에서 정보를 업데이트합니다. 그런 다음 자동으로 원래 큐 관리자에 다시 클러스터 송신자 채널을 작성하고 클러스터에 대한 이 큐 관리자 정보를 보냅니다. 따라서 큐 관리자는 클러스터에 대해 학습하고 클러스터는 큐 관리자에 대해 학습합니다.

36 페이지의 그림 9를 다시 보십시오. 큐 관리자 QM3에 연결된 애플리케이션이 QM2의 큐에 일부 메시지를 보내려 한다고 가정합니다. QM3이(가) 해당 큐에 처음으로 액세스해야 하는 경우 전체 저장소를 참조하여 해당 큐를 발견합니다. 이 경우 전체 저장소는 송신자 채널 CLSTR1.QM2을(를) 사용하여 액세스하는 QM2입니다. 저장소의 정보를 사용하여 해당 큐에 대한 리모트 정의를 자동으로 작성할 수 있습니다. 큐가 QM1에 있으면 QM2은(는) 전체 저장소이므로 이 메커니즘은 계속 작동합니다. 전체 저장소에 클러스터의 모든 오브젝트에 대한 전체 레코드가 있습니다. 후자의 경우, QM3에서 QM1의 클러스터 수신자 채널에 해당하는 클러스터 송신자 채널을 자동으로 작성하므로 둘 간에 직접 통신할 수도 있습니다.

37 페이지의 그림 11는 자동으로 작성된 두 개의 클러스터 송신자 채널이 있는 동일한 클러스터를 보여줍니다. 클러스터 송신자 채널은 클러스터 수신자 채널 CLSTR1.QM3과(와) 결합하는 두 개의 점선으로 표시됩니다. QM1이(가) 메시지를 전송하는 데 사용하는 클러스터 전송 큐, SYSTEM.CLUSTER.TRANSMIT.QUEUE도 표시합니다. 클러스터의 모든 큐 관리자에는 클러스터 전송 큐가 있어서 여기에서 동일한 클러스터의 다른 모든 큐 관리자로 메시지를 보낼 수 있습니다.

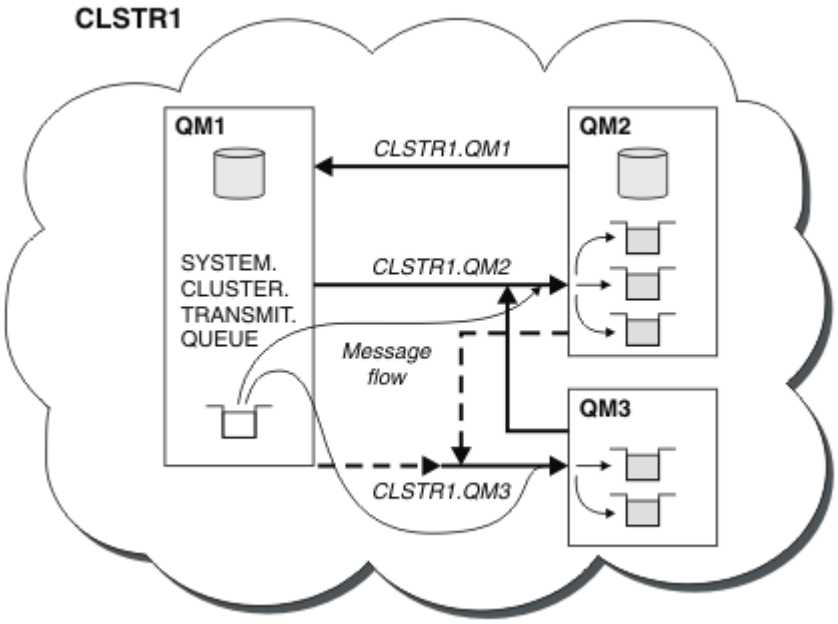


그림 *11.* 자동 정의된 채널이 표시된 큐 관리자의 클러스터

참고: 기타 다이어그램은 수동 정의하는 채널의 수신 측만을 표시합니다. 송신 끝은 대부분 필요할 때 자동으로 정의되므로 생략되었습니다. 대부분의 클러스터 송신자 채널의 자동 정의는 클러스터의 기능 및 효율에 결정적 요소입니다.

관련 개념

27 페이지의 『클러스터링과 분산 큐잉의 비교』
분산 큐잉 및 클러스터링을 사용하여 큐 관리자를 연결하기 위해 정의해야 하는 컴포넌트를 비교합니다.

클러스터의 컴포넌트

관련 태스크

큐 관리자 클러스터 구성
새 클러스터 설정

클러스터링: 우수 사례

클러스터는 큐 관리자를 상호 연결하기 위한 메커니즘을 제공합니다. 이 절에 설명된 우수 사례는 테스트 및 고객의 피드백에 기반을 둔 것입니다.

성공적인 클러스터 설정은 양질의 애플리케이션 관리 및 네트워크 설계와 같은 IBM MQ 기본사항에 대한 완벽한 이해와 우수한 계획에 의존합니다. 계속하기 전에 관련 항목의 정보를 반드시 숙지하십시오.

관련 개념

분산 큐잉 및 클러스터

클러스터

관련 태스크

21 페이지의 『클러스터 설계』

클러스터는 초기 구성 및 진행 중인 관리를 모두 단순화하는 방식으로 큐 관리자를 상호 연결하기 위한 메커니즘을 제공합니다. 클러스터가 제대로 기능하고 가용성 및 응답성의 필수 레벨을 달성할 수 있도록 클러스터를 주의해서 설계해야 합니다.

클러스터 모니터링

클러스터링: 중첩 클러스터에 대한 특수 고려사항

이 주제는 IBM MQ 클러스터 계획 및 관리를 위한 지침을 제공합니다. 이 정보는 고객의 피드백과 테스트에 기반을 둔 지침입니다.

클러스터 소유권

다음 정보를 읽기 전에 중첩 클러스터를 숙지하십시오. 필요한 정보는 32 페이지의 『중첩 클러스터』 및 클러스터 간 메시지 경로 구성을 참조하십시오.

중첩 클러스터로 이루어진 시스템을 구성하고 관리할 때에는 다음 사항을 준수하는 것이 가장 좋습니다.

- IBM MQ 클러스터가 이전에 설명된 대로 '느슨하게 결합'되어 있어도 클러스터를 단일 관리 단위로 간주하는 것이 유용합니다. 개별 큐 관리자의 정의 간 상호작용이 클러스터의 원활한 기능에 중요하기 때문에 이 개념이 사용됩니다. 예를 들어, 워크로드 밸런스가 조절된 클러스터 큐를 사용할 때에는 단일 관리자나 팀이 클러스터 전반에 퍼진 정의에 따라 달라지는 메시지의 가능한 전체 목적지 세트를 이해하는 것이 중요합니다. 보다 구체적으로는 클러스터 송신자/수신자 채널 쌍이 도처에서 호환 가능해야 합니다.
- 여러 클러스터(개별 팀/개인이 관리할)가 모이는 이전 개념을 고려할 경우 게이트웨이 큐 관리자의 관리를 제어할 명확한 정책을 적절하게 보유하는 것이 중요합니다.
- 중첩 클러스터를 단일 네임스페이스로 처리하면 유용합니다. 즉, 채널 이름과 큐 관리자 이름이 단일 클러스터 전반에서 고유해야 합니다. 전체 토폴로지에서 이름이 고유하면 관리가 훨씬 용이합니다. 최상의 방법은 적합한 이름 지정 규칙을 따르는 것입니다. 30 페이지의 『클러스터 이름 지정 규칙』에 가능한 규칙이 설명되어 있습니다.
- 때로는 관리 및 시스템 관리 협력이 필수적이기도 합니다. 예를 들어, 겹치는 데 필요한 서로 다른 클러스터를 소유하는 조직 간의 협업입니다. 누가 무엇을 소유하고 있으며 적용 가능한 규칙 및 규칙을 명확하게 이해하면 클러스터가 겹치는 경우 클러스터링이 원활하게 실행되는 데 도움이 됩니다.

중첩 클러스터: 게이트웨이

일반적으로 단일 클러스터는 다중 클러스터보다 관리가 더 용이합니다. 따라서 많은 수의 작은 클러스터를 작성(예를 들어, 모든 애플리케이션에 하나씩)하는 방식은 대개 피하는 방법입니다.

하지만 서비스 클래스를 제공하려면 중첩 클러스터를 구현할 수 있습니다. 예를 들면, 다음과 같습니다.

- 발행/구독에 대해 더 작은 클러스터가 있는 동심 클러스터가 있는 경우. 자세한 정보는 시스템 크기 조절 방법을 참조하십시오.
- 일부 큐 관리자를 다른 팀에서 관리하는 경우. 자세한 정보는 이전 38 페이지의 『클러스터 소유권』 절을 참조하십시오.
- 조직 또는 지리적 관점에서 타당할 경우.
- 동등한 클러스터가 이름 해석을 사용하여 작동하는 경우 (예: 기존 클러스터에서 TLS를 구현하는 경우).

겹치는 클러스터의 보안 이점은 없습니다. 두 개의 서로 다른 팀에서 관리하는 클러스터가 겹치도록 허용하여 토폴로지뿐만 아니라 팀을 효과적으로 결합할 수 있습니다.

- 이러한 클러스터에 알려진 모든 이름은 다른 클러스터에 액세스할 수 있습니다.
- 한 클러스터에서 광고되는 모든 이름을 다른 클러스터에서 광고하여 적합한 메시지를 끌어낼 수 있습니다.

- 게이트웨이에 인접한 큐 관리자의 광고되지 않은 오브젝트는 게이트웨이가 구성원인 클러스터에서 해석될 수 있습니다.

네임스페이스는 두 클러스터 모두의 유니온이며 단일 네임스페이스로 처리해야 합니다. 따라서 중첩 클러스터의 소유권은 두 클러스터의 모든 관리자가 공유합니다.

시스템에 여러 클러스터가 있으면 한 클러스터의 큐 관리자에서 다른 클러스터의 큐 관리자로 메시지를 라우팅하기 위한 요구사항이 있을 수도 있습니다. 이 경우 다수의 클러스터를 일정한 방식으로 서로 연결시켜야 합니다. 우수한 패턴은 클러스터 간 게이트웨이 큐 관리자를 사용하는 것입니다. 이렇게 배열하면 관리하기 어려운 포인트-투-포인트 채널의 메시가 빌드되지 않고 보안 정책으로서 이러한 문제를 관리할 좋은 위치가 제공됩니다. 이 배열을 완성하는 두 가지 특징적 방법이 있습니다.

1. 두 번째 클러스터 수신자 정의를 사용하여 하나(이상)의 큐 관리자를 두 클러스터 모두에 두십시오. 이 배열은 관리 정의 수가 더 적지만 이전에 언급한 것처럼, 중첩 클러스터의 소유권을 두 클러스터의 모든 관리자가 공유함을 의미합니다.
2. 일반적인 포인트-투-포인트 채널을 사용하여 클러스터 1의 큐 관리자와 클러스터 2의 큐 관리자가 쌍이 되게 하십시오.

둘 중 어느 경우든, 다양한 도구를 사용하여 트래픽을 적절하게 라우팅할 수 있습니다. 특히, 큐 또는 큐 관리자 알리어스를 사용하여 다른 클러스터로 라우팅할 수 있으며, 비어 있는 **RQMNAME** 특성이 있는 큐 관리자 알리어스는 원하는 경우 워크로드 밸런싱을 다시 구동합니다.

관련 개념

30 페이지의 『클러스터 이름 지정 규칙』
큐 관리자가 속한 클러스터를 식별하는 이름 지정 규칙을 사용하여 동일한 클러스터의 큐 관리자 이름을 지정할 것을 고려하십시오. 채널 이름에 유사한 이름 지정 규칙을 사용하고 채널 특성을 설명하도록 확장하십시오.

클러스터링*:* 토폴로지 설계 고려사항
이 주제는 IBM MQ 클러스터 계획 및 관리를 위한 지침을 제공합니다. 이 정보는 고객의 피드백과 테스트에 기반을 둔 지침입니다.

사용자 애플리케이션과 내부 관리 프로세스가 위치할 곳을 미리 생각해 두면 많은 문제점을 피하고 나중에 최소화할 수 있습니다. 이 주제는 성능을 개선하고 유지보수 태스크를 클러스터 배율로 단순화할 수 있는 설계 의사결정에 대한 정보를 포함합니다.

- 39 페이지의 『클러스터링 인프라의 성능』
- 40 페이지의 『전체 저장소』
- 41 페이지의 『애플리케이션에 전체 저장소의 큐를 사용해야 합니까?』
- 41 페이지의 『채널 정의 관리』
- 41 페이지의 『다중 채널에 대한 워크로드 밸런싱』

클러스터링 인프라의 성능

애플리케이션이 클러스터의 큐 관리자에서 큐를 열려고 시도하면 클러스터에서 큐가 존재하는 위치를 알 수 있도록 큐 관리자는 해당 큐의 전체 저장소에 관심을 등록합니다. 큐 위치 또는 구성에 대한 업데이트는 전체 저장소를 통해 관심 있는 큐 관리자에 자동으로 송신됩니다. 이러한 관심 등록을 내부적으로 구독이라 합니다(이 구독은 IBM MQ의 발행/구독 메시징에 사용되는 IBM MQ 구독과 동일하지 않음).

클러스터에 대한 모든 정보는 모든 전체 저장소를 통과합니다. 따라서 전체 저장소는 관리 메시지 트래픽을 위해 항상 클러스터에 사용됩니다. 이 구독을 관리할 때의 많은 시스템 자원 사용과 구독 및 결과적인 구성 메시지의 전송은 클러스터링 인프라에 상당한 로드를 야기시킬 수 있습니다. 이 로드가 이해되고 가능한 위치에서 최소화되는지 확인할 때 고려할 많은 사항이 있습니다.

- 클러스터 큐를 사용하는 개별 큐 관리자가 더 많을수록 시스템에 더 많은 구독이 있으므로 특히 전체 저장소 큐 관리자에서 변경이 발생하고 관심 있는 구독을 알려야 할 때에는 관리 오버헤드가 더 커집니다. 불필요한 트래픽과 전체 저장소 로드를 최소화하기 위한 한 가지 방법은 유사한 애플리케이션(즉, 동일한 큐에 대해 작업하는 애플리케이션)을 더 작은 수의 큐 관리자에 연결시키는 것입니다.
- 성능에 영향을 미치는 시스템의 구독 수 외에도, 클러스터된 큐 구성의 빈번한 변경과 같은 클러스터된 오브젝트의 구성 변화율은 성능에 영향을 줄 수 있습니다.

- 큐 관리자가 다중 클러스터의 멤버(즉, 중첩 클러스터 시스템의 일부)일 때에는 동일한 큐 관리자가 둘 이상의 클러스터에 대한 전체 저장소일지라도 큐에서 등록한 관심으로 인해 멤버로 속해 있는 각 클러스터에 대한 구독이 발생합니다. 이 배열은 시스템의 로드를 증가시키며 단일 클러스터보다 여러 중첩 클러스터가 필요할지 여부를 고려하는 한 가지 이유입니다.
- 애플리케이션 메시지 트래픽(즉, IBM MQ 애플리케이션이 클러스터 큐에 보내는 메시지)은 목적지 큐 관리자에 도달하기 위해 전체 저장소를 거치지 않습니다. 이 메시지 트래픽은 클러스터에 메시지를 넣는 큐 관리자와 클러스터 큐가 존재하는 큐 관리자 간에 직접 송신됩니다. 따라서 전체 저장소 큐 관리자가 우연히 언급한 두 개의 큐 관리자 중 하나인 경우가 아니면 전체 저장소 큐 관리자에 관하여 애플리케이션 메시지 트래픽을 높은 비율로 유지할 필요가 없습니다. 이러한 이유로, 클러스터링 인프라 로드가 중요한 클러스터의 애플리케이션 메시지 트래픽에는 전체 저장소 큐 관리자를 사용하지 않는 것이 좋습니다.

전체 저장소

저장소는 클러스터의 멤버인 큐 관리자에 대한 정보의 콜렉션입니다. 클러스터에서 모든 큐 관리자에 대한 전체 정보 세트를 호스팅하는 큐 관리자에 전체 저장소가 있습니다. 전체 저장소 및 부분 저장소에 대한 자세한 정보는 클러스터 저장소를 참조하십시오.

신뢰할 수 있고 가능한 한 가용성이 높으며 단일 장애점을 피해야 하는 서버에 전체 저장소를 보유해야 합니다. 클러스터는 두 개의 전체 저장소를 갖도록 설계되어야 합니다. 한 전체 저장소에 장애가 있어도 클러스터가 여전히 작동할 수 있습니다.

클러스터의 큐 관리자가 수행한 클러스터 자원 업데이트의 세부사항(예를 들어, 클러스터된 큐)은 해당 큐 관리자에서 이 클러스터의 많아야 두 개인 전체 저장소로(또는 클러스터에 전체 저장소 큐 관리자가 하나만 있으면 한 전체 저장소로) 송신됩니다. 이 전체 저장소는 정보를 보유하며 관심을 표시하는(즉, 구독하는) 클러스터의 큐 관리자에 이를 전파합니다. 클러스터의 각 멤버에 클러스터 자원의 최신 보기가 있는지 확인하려면 각 큐 관리자가 언제든지 적어도 하나의 전체 저장소 큐 관리자와 통신할 수 있어야 합니다.

큐 관리자가 어떠한 이유로 전체 저장소와 통신할 수 없는 경우 이미 캐싱된 정보 레벨에 따라 일정 기간 클러스터에서 계속하여 작동할 수 있지만 이전에 사용하지 않은 클러스터 자원에 대한 액세스 또는 새 업데이트는 사용할 수 없습니다.

이러한 이유로 항상 두 개의 전체 저장소를 사용 가능하게 유지해야 합니다. 그렇다고 해서 이 배열이 전체 저장소가 없는 잠시 동안 클러스터가 적절하게 작동하기 때문에 극단적인 조치를 취해야 함을 의미하는 것은 아닙니다.

클러스터 정보의 가용성 외에, 두 개의 전체 저장소 큐 관리자를 보유해야 하는 또 다른 이유가 있습니다. 이 이유는 전체 저장소 캐시에 보유된 클러스터 정보가 복구 용도로 두 위치에 존재하게 하는 것입니다. 전체 저장소가 하나만 있고 클러스터에 대한 정보를 잃으면 클러스터를 작동시키기 위해 클러스터 내 모든 큐 관리자에 대한 수동 개입이 필요합니다. 하지만 두 개의 전체 저장소가 있으면 두 개의 전체 저장소에서 항상 정보가 발행되고 구독되므로 최소의 노력으로 실패한 전체 저장소를 복구할 수 있습니다.

- 두 개의 전체 저장소 클러스터 설계에서는 클러스터의 사용자에게 영향을 주지 않고 전체 저장소 큐 관리자에서 유지보수를 수행할 수 있습니다. 클러스터는 한 저장소만 있어도 계속해서 기능하므로 저장소가 작동 중지될 경우 유지보수가 적용되고 한 번에 다시 백업됩니다. 두 번째 전체 저장소가 가동 중단되어도 실행 애플리케이션에 최소 3일 동안은 영향이 미치지 않습니다.
- 지리적 이유로 로컬 전체 저장소를 사용하는 것처럼, 세 번째 저장소를 사용해야 하는 타당한 이유가 없으면 두 개의 저장소 설계를 사용하십시오. 세 개의 전체 저장소를 사용한다는 것은 현재 사용 중인 두 개를 결코 알지 못하며 여러 워크로드 관리 매개변수 간의 상호작용으로 인한 관리 문제가 있을 수 있음을 의미합니다. 전체 저장소를 셋 이상은 보유하지 않는 것이 좋습니다.
- 여전히 가용성을 개선할 필요가 있으면 전체 저장소 큐 관리자를 다중 인스턴스 큐 관리자로서 호스트하거나 플랫폼별 고가용성 지원을 사용하여 가용성을 개선할 것을 고려하십시오.
- 수동으로 정의된 클러스터 송신자 채널과 모든 전체 저장소 큐 관리자가 완전히 상호 연결되어 있어야 합니다. 일부 정당한 사유로 클러스터에 셋 이상의 전체 저장소를 사용할 때에는 특히 주의해야 합니다. 이 경우 하나 이상의 채널이 빈번히 누락될 수 있으며 이는 바로 드러나지 않습니다. 완전히 상호 연결되지 않은 경우에는 자주 발생하는 문제점을 진단하기 어렵습니다. 일부 전체 저장소는 모든 저장소 데이터를 보유하지 않아서 결과적으로 클러스터에 있는 큐 관리자가 연결하는 전체 저장소에 따라 클러스터의 보기가 상이하기 때문에 진단이 어렵습니다.

애플리케이션에 전체 저장소의 큐를 사용해야 합니까?

전체 저장소는 대부분 다른 큐 관리자와 유사하므로 전체 저장소의 애플리케이션 큐를 호스트하고 이 큐 관리자에 직접 애플리케이션을 연결시킬 수 있습니다. 애플리케이션에 전체 저장소의 큐를 사용해야 합니까?

일반적으로 허용되는 대답은 "아니오"입니다. 이 구성이 가능하더라도 많은 고객은 이 큐 관리자를 전체 저장소 클러스터 캐시의 유지보수 전용으로 사용하는 것을 선호합니다. 어느 옵션을 사용할지 결정할 때 고려할 사항은 여기에 설명되어 있지만 궁극적으로는 클러스터 아키텍처가 환경의 특정 요구에 적합해야 합니다.

- 업그레이드: 일반적으로 IBM MQ의 새 릴리스에서 새 클러스터 기능을 사용하려면 이 클러스터의 전체 저장소 큐 관리자를 먼저 업그레이드해야 합니다. 클러스터의 애플리케이션에 새 기능을 사용하려는 경우 같은 위치에 있는 많은 애플리케이션을 테스트하지 않고 전체 저장소(및 부분 저장소의 일부 서브세트)를 업데이트하는 것이 유용할 수 있습니다.
- 유지보수: 이와 유사한 방식으로 전체 저장소에 긴급 유지보수를 적용해야 하는 경우 애플리케이션 없이 **REFRESH** 명령으로 전체 저장소를 재시작하거나 새로 고칠 수 있습니다.
- 성능: 클러스터가 커지고 전체 저장소 클러스터 캐시 유지보수에 대한 요구가 늘어남에 따라 애플리케이션을 별도로 유지하면 시스템 자원에 대한 경합을 통해 애플리케이션 성능에 영향을 미칠 위험성이 감소합니다.
- 하드웨어 요구사항: 일반적으로 전체 저장소가 강력할 필요는 없습니다. 예를 들어, 가용성 기대치가 양호한 단순 UNIX 서버면 충분합니다. 또는 매우 크거나 끊임없이 변경되는 클러스터의 경우 전체 저장소 컴퓨터의 성능을 고려해야 합니다.
- 소프트웨어 요구사항: 요구사항은 대개 전체 저장소에서 애플리케이션 큐를 호스트하도록 선택하는 주된 이유입니다. 작은 클러스터에서 공동 위치는 전체적으로 더 작은 수의 큐 관리자/서버에 대한 요구사항을 의미하는 것일 수 있습니다.

채널 정의 관리

단일 클러스터 내에도 두 개의 큐 관리자 간에 다중 라우트를 제공하는 여러 채널 정의가 존재할 수 있습니다.

단일 클러스터 내에 병렬 채널을 사용하면 때로 이점이 있지만 이 설계 의사결정을 철저히 고려해야 합니다. 이 설계는 복잡도를 더하는 것 외에 채널 이용을 감소시켜서 성능을 저하시킬 수 있습니다. 테스트 시에 일반적으로 많은 메시지를 일정한 비율로 보내서 병렬 채널이 완전히 사용되기 때문에 이러한 상황이 발생합니다. 하지만 메시지 스트림이 일정하지 않은 실세계 조건에서는 워크로드 밸런싱 알고리즘으로 인해 메시지 플로우가 한 채널에서 다른 채널로 전환될 때 성능이 떨어집니다.

큐 관리자가 다중 클러스터의 멤버인 경우 각 클러스터마다 별도의 CLUSRCVR 채널을 정의하지 않고 클러스터 이름 목록과 함께 단일 채널 정의를 사용하는 옵션이 있습니다. 하지만 이 설정은 나중에 관리의 어려움을 유발할 수 있습니다. 예를 들어, TLS를 한 클러스터에 적용하고 두 번째 클러스터에는 적용하지 않을 경우를 고려해 보십시오. 별도의 정의를 작성하는 것이 바람직하며 30 페이지의 『클러스터 이름 지정 규칙』에 제안된 이름 지정 규칙이 이를 지원합니다.

다중 채널에 대한 워크로드 밸런싱

이 정보는 주제에 대한 고급 이해를 돕기 위한 것입니다. 이 주제에 대한 기본 설명은(여기의 정보를 사용하기 전에 이해해야 함) 워크로드 관리에 클러스터 사용, 클러스터의 워크로드 밸런싱, 클러스터 워크로드 관리 알고리즘을 참조하십시오.

클러스터 워크로드 관리 알고리즘은 큰 도구 세트를 제공하지만 도구의 작동 및 상호작용 방식을 완전히 이해하지 않은 채로 도구를 함께 사용해서는 안됩니다. 워크로드 밸런싱 프로세스에 채널이 얼마나 중요한지는 바로 드러나지 않을 수도 있습니다. 워크로드 관리 라운드로빈 알고리즘은 클러스터된 큐를 소유하는 큐 관리자에 대한 다중 클러스터 채널이 해당 큐의 다중 인스턴스로 처리되는 것처럼 작동합니다. 이 프로세스는 다음 예에 보다 자세히 설명되어 있습니다.

1. 클러스터에서 큐를 호스팅하는 두 개의 큐 관리자가 있습니다(QM1 및 QM2).
2. QM1에는 다섯 개의 클러스터 수신자 채널이 있습니다.
3. QM2에는 클러스터 수신자 채널이 하나만 있습니다.
4. QM3의 **MQPUT** 또는 **MQOPEN**이 인스턴스를 선택할 때 알고리즘은 QM2보다 QM1에 메시지를 보낼 가능성이 5배 더 높습니다.

5. 4 단계의 상황은 알고리즘이 QM1에 대한 모든 채널 및 QM2에 대한 단일 채널 전체의 라운드 로빈과 (5+1) 중에서 선택할 수 있는 6가지 옵션을 볼 수 있기 때문에 발생합니다.

감지하기 힘든 또 다른 작동은 로컬 큐 관리자에 우연히 한 인스턴스가 구성되어 있는 클러스터된 큐에 메시지를 넣을 때조차도 IBM MQ가 로컬 클러스터 수신자 채널의 상태를 사용하여 큐의 로컬 인스턴스나 큐의 원격 인스턴스에 메시지를 넣을지 결정하는 경우입니다. 이 시나리오에서,

1. 메시지를 넣을 때 워크로드 관리 알고리즘은 개별 클러스터 큐를 보지 않고 이 목적지에 도달할 수 있는 클러스터 채널을 봅니다.
2. 로컬 목적지에 도달하기 위해 로컬 수신자 채널이 이 목록에 포함되어 있습니다(메시지를 보내는 데 사용되지 않지만).
3. 로컬 수신자 채널이 중지될 때 CLUSRCVR이 중지되지 않으면 워크로드 관리 알고리즘은 기본적으로 대체 인스턴스를 선호합니다. 목적지에 대한 여러 로컬 CLUSRCVR 인스턴스가 있고 최소 하나가 중지되지 않을 경우 로컬 인스턴스를 적합하다고 간주합니다.

클러스터링*:* 다중 클러스터 전송 큐를 사용하여 애플리케이션 격리
클러스터의 큐 관리자 간에 메시지 플로우를 분리시킬 수 있습니다. 다른 클러스터 송신자 채널에서 전송되는 메시지를 상이한 클러스터 전송 큐에 둘 수 있습니다. 단일 클러스터 또는 중첩 클러스터에 이 접근법을 사용할 수 있습니다. 이 주제는 사용할 접근법을 선택하는 데 유용한 예와 몇 가지 우수 사례를 제공합니다.

애플리케이션을 배치할 때에는 다른 애플리케이션과 공유하는 IBM MQ 자원 및 공유하지 않는 자원을 선택합니다. 공유할 수 있는 여러 유형의 자원이 있으며 주된 자원은 서버 자체, 큐 관리자, 채널, 큐입니다. 별도의 큐, 채널, 큐 관리자 또는 심지어 서버를 개별 애플리케이션에 할당하여 공유 자원 수가 작은 애플리케이션을 구성하도록 선택할 수 있습니다. 이러한 경우에는 전체 시스템 구성이 더 커지고 복잡해집니다. IBM MQ 클러스터를 사용하면 더 많은 서버, 큐 관리자, 큐 및 채널을 관리하는 복잡도가 감소되지만 다른 공유 자원인 클러스터 전송 큐 (SYSTEM.CLUSTER.TRANSMIT.QUEUE)도 도입됩니다.

43 페이지의 그림 12은(는) SYSTEM.CLUSTER.TRANSMIT.QUEUE 공유의 중요성을 설명하는 대규모 IBM MQ 배치의 일부입니다. 다이어그램에서 Client App 애플리케이션은 CL1 클러스터의 QM2 큐 관리자에 연결됩니다. Client App의 메시지는 Server App 애플리케이션에서 처리합니다. 메시지는 CLUSTER2의 QM3 큐 관리자에 있는 Q1 클러스터 큐에서 Server App을(를) 통해 검색합니다. 클라이언트 및 서버 애플리케이션이 동일한 클러스터에 있지 않기 때문에 게이트웨이 큐 관리자 QM1이(가) 메시지를 전송합니다.

클러스터 게이트웨이를 구성하는 일반적인 방법은 게이트웨이 큐 관리자를 모든 클러스터의 멤버로 만드는 것입니다. 게이트웨이 큐 관리자에는 모든 클러스터의 클러스터 큐에 대한 클러스터된 알리어스 큐가 정의되어 있습니다. 클러스터된 큐 알리어스는 모든 클러스터에서 사용 가능합니다. 클러스터 큐 알리어스에 넣은 메시지는 게이트웨이 큐 관리자를 통해 올바른 목적지로 라우팅됩니다. 게이트웨이 큐 관리자는 클러스터링된 알리어스 큐에 송신된 메시지를 QM1의 공통 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 넣습니다.

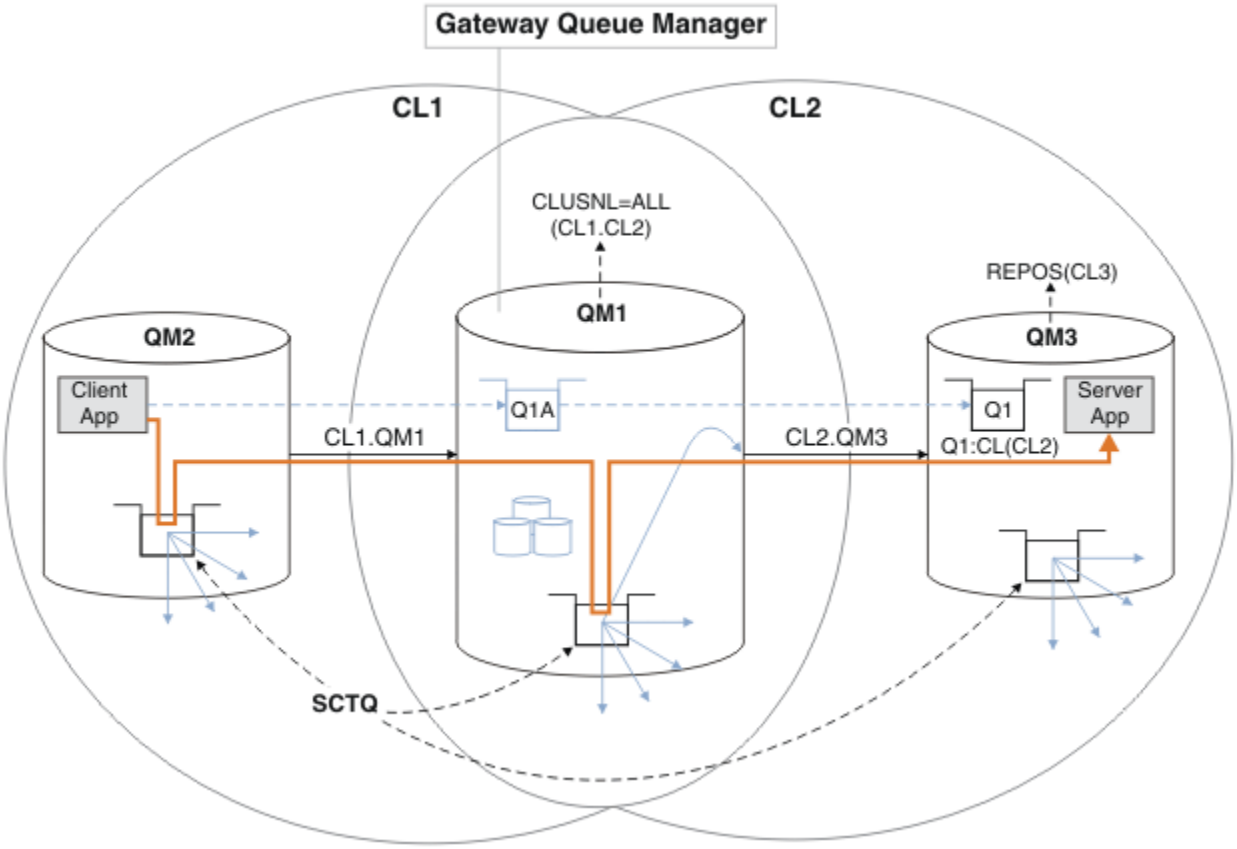
허브 및 스포크 아키텍처에서는 클러스터 간의 모든 메시지가 게이트웨이 큐 관리자를 통과해야 합니다. 결과적으로 모든 메시지는 QM1, SYSTEM.CLUSTER.TRANSMIT.QUEUE의 단일 클러스터 전송 큐를 통해 플로우됩니다.

성능면에서 보면 단일 큐는 문제가 되지 않습니다. 공용 전송 큐는 일반적으로 성능 병목을 나타내지 않습니다. 게이트웨이의 메시지 처리량이 대부분 연결된 채널의 성능에 따라 판별됩니다. 처리량은 대개 큐의 수나 채널을 사용하는 큐의 메시지 수에 영향을 받지 않습니다.

다른 관점에서는, 다중 애플리케이션에 단일 전송 큐를 사용할 경우 문제가 있습니다.

- 한 목적지로 향하는 메시지 플로우를 다른 목적지로 향하는 메시지 플로우와 분리시킬 수 없습니다. 목적지가 다른 큐 관리자의 다른 클러스터에 있어도 전달 전에 메시지의 스토리지를 구분할 수 없습니다.

  한 클러스터 목적지가 사용 불가능하게 되면 이 목적지에 대한 메시지가 단일 전송 큐에 빌드되어 결과적으로 메시지가 이 큐를 가득 채우게 됩니다. 전송 큐는 일단 가득 차면 어느 클러스터 목적지에 대한 전송 큐에도 메시지를 받지 않습니다.
- 다른 클러스터 목적지로의 메시지 전송을 모니터하기 쉽지 않습니다. 모든 메시지가 단일 전송 큐에 있습니다. 전송 큐 용량을 표시해도 메시지가 모든 목적지로 전송되고 있는지 거의 표시되지 않습니다.



참고: 43 페이지의 그림 12의 화살표와 다음 그림은 여러 가지 유형입니다. 단색 화살표는 메시지 플로우를 표시합니다. 단색 화살표의 레이블은 메시지 채널 이름입니다. 회색 실선 화살표는 SYSTEM.CLUSTER.TRANSMIT.QUEUE에서 클러스터 송신자 채널로의 잠재적 메시지 플로우입니다. 검은색 파선은 레이블을 대상에 연결합니다. 회색 파선은 참조입니다(예를 들어, Client App의 MQOPEN 호출에서 클러스터 알리어스 큐 정의의 Q1A까지의 파선).

그림 *12. IBM MQ* 클러스터를 사용하여 허브 및 스포크 아키텍처에 배치되는 클라이언트 서버 애플리케이션

43 페이지의 그림 12에서 Server App의 클라이언트가 Q1A 큐를 엽니다. 메시지는 QM2의 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 배치되고 QM1의 SYSTEM.CLUSTER.TRANSMIT.QUEUE(으)로 전송된 다음 QM3의 Q1(으)로 전송되어 Server App 애플리케이션이 수신합니다.

Client App의 메시지는 QM2 및 QM1의 시스템 클러스터 전송 큐를 통과합니다. 43 페이지의 그림 12의 목적은 메시지가 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 저장되지 않도록 클라이언트 애플리케이션에서 게이트웨이 큐 관리자의 메시지 플로우를 분리하는 것입니다. 클러스터된 다른 모든 큐 관리자에서 플로우를 분리시킬 수 있습니다. 또한 클라이언트로 돌아오는 다른 방향의 플로우도 분리시킬 수 있습니다. 솔루션 설명을 간략하게 보여 주기 위해 설명은 클라이언트 애플리케이션으로부터의 단일 플로우만 고려합니다.

클러스터 게이트웨이 큐 관리자의 클러스터 메시지 트래픽 분리를 위한 솔루션

문제 해결을 위한 한 가지 방법은 클러스터 간 브릿지로 큐 관리자 알리어스 또는 리모트 큐 정의를 사용하는 것입니다. 게이트웨이 큐 관리자의 각 메시지 플로우를 분리시킬 클러스터된 리모트 큐 정의, 전송 큐, 채널을 작성하십시오(게이트웨이 큐 관리자에서 송신된 메시지를 분리할 리모트 큐 정의 추가 참조).

IBM WebSphere® MQ 7.5부터는, 클러스터 큐 관리자가 단일 클러스터 전송 큐로 제한되지 않습니다. 다음 두 가지 선택사항이 있습니다.

1. 추가 클러스터 전송 큐를 수동으로 정의하고 각 전송 큐에서 메시지를 전송하는 클러스터 송신자 채널을 정의하십시오(게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 분리할 클러스터 전송 큐 추가 참조).

2. 큐 관리자가 추가 클러스터 전송 큐를 자동으로 작성하고 관리하게 하십시오. 큐 관리자는 각 클러스터 송신자 채널마다 각기 다른 클러스터 전송 큐를 정의합니다(메시지 트래픽을 격리하기 위해 클러스터 전송 큐를 분리하도록 기본값 변경 참조).

일부 클러스터 송신자 채널의 정의된 클러스터 전송 큐를 나머지를 관리하는 큐 관리자와 수동으로 결합시켜야 합니다. 전송 큐 결합은 게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가에 사용된 접근법입니다. 이 솔루션에서 클러스터 간 대부분의 메시지는 공통 SYSTEM.CLUSTER.TRANSMIT.QUEUE을(를) 사용합니다. 한 중요 애플리케이션은 수동으로 정의된 클러스터 전송 큐를 사용하여 모든 메시지 플로우를 다른 플로우와 분리시킵니다.

게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가의 구성은 제한됩니다. 이 구성은 다른 클러스터 큐와 동일한 클러스터에 있는 동일한 큐 관리자의 클러스터 큐로 향하는 메시지 트래픽을 분리시키지 않습니다. 분산 큐잉의 일부인 리모트 큐 정의를 사용하여 개별 큐로 메시지 트래픽을 분리시킬 수 있습니다. 클러스터의 경우에는 다수의 클러스터 전송 큐를 사용하여 다른 클러스터 송신자 채널로 향하는 메시지 트래픽을 분리시킬 수 있습니다. 동일한 클러스터의 동일한 큐 관리자에 있는 여러 클러스터 큐는 클러스터 송신자 채널을 공유합니다. 이러한 큐에 대한 메시지는 게이트웨이 큐 관리자로부터 전달되기 전에 동일한 전송 큐에 저장됩니다. 게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 및 클러스터 전송 큐 추가의 구성에서는, 또 다른 클러스터를 추가하고 큐 관리자 및 클러스터 큐를 새 클러스터의 멤버로 지정해서 제한사항을 회피합니다. 새 큐 관리자가 클러스터의 유일한 큐 관리자일 수 있습니다. 더 많은 큐 관리자를 클러스터에 추가하고 동일한 클러스터를 사용하여 큐 관리자의 클러스터 큐를 분리시킬 수도 있습니다.

관련 개념

26 페이지의 『액세스 제어 및 다중 클러스터 전송 큐』
애플리케이션이 리모트 클러스터 큐에 메시지를 넣을 때 검사하는 세 가지 모드 중에서 선택하십시오. 모드에서는 클러스터 큐에 대해 원격으로 검사하거나 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 대해 로컬로 검사하거나 클러스터 큐 또는 클러스터 큐 관리자의 로컬 프로파일에 대해 검사합니다.

클러스터 전송 큐 및 클러스터 송신자 채널에 대한 작업

32 페이지의 『중첩 클러스터』
중첩 클러스터는 추가 관리 기능을 제공합니다. 이름 목록을 사용하여 중첩 클러스터를 관리하는 데 필요한 명령의 수를 줄이십시오.

관련 태스크

리모트 클러스터 큐에 메시지를 넣는 권한 부여

게이트웨이 큐 관리자에서 송신된 메시지를 격리하기 위한 리모트 큐 정의 추가

게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가

게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 및 클러스터 전송 큐 추가

메시지 트래픽을 격리하기 위해 클러스터 전송 큐를 분리하도록 기본값 변경

게이트웨이 큐 관리자로 두 개의 중첩 클러스터 작성

클러스터 간 메시지 경로 구성

보안 설정

관련 참조

setmqaut

클러스터링: 클러스터 전송 큐를 구성하는 방법을 계획
클러스터 전송 큐의 선택에 대한 지침입니다. 하나의 공용 기본 큐, 별도의 기본 큐, 또는 수동으로 정의된 큐를 구성할 수 있습니다.

시작하기 전에

47 페이지의 『사용할 클러스터 전송 큐의 유형 선택 방법』의 내용을 검토하십시오.

이 태스크 정보

다음은 클러스터 전송 큐를 선택하도록 큐 관리자를 구성하는 방법을 계획할 때 사용할 몇 가지 선택사항입니다.

1. 클러스터 메시지 전송을 위한 기본 클러스터 전송 큐는 무엇입니까?

a. 공통 클러스터 전송 큐, SYSTEM.CLUSTER.TRANSMIT.QUEUE입니다.

b. 별도의 클러스터 전송 큐. 큐 관리자가 별도의 클러스터 전송 큐를 관리합니다. 모델 큐, SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE에서 영구 동적 큐로 작성합니다. 사용하는 각 클러스터 송신자 채널마다 클러스터 전송 큐를 하나씩 작성합니다.

2. 수동으로 작성하도록 결정한 클러스터 전송 큐의 경우 추가로 다음 두 선택사항이 있습니다.

a. 수동으로 구성하도록 결정한 각 클러스터 송신자 채널마다 별도의 전송 큐를 정의합니다. 이 경우, 전송 큐의 **CLCHNAME** 큐 속성을 클러스터 송신자 채널의 이름으로 설정하십시오. 이 전송 큐에서 메시지를 전송할 클러스터 송신자 채널을 선택하십시오.

b. 클러스터 송신자 채널의 그룹에 대한 메시지 트래픽을 동일한 클러스터 전송 큐로 결합하십시오(45 페이지의 그림 13 참조). 이 경우, 각 공통 전송 큐의 **CLCHNAME** 큐 속성을 일반 클러스터 송신자 채널 이름으로 설정하십시오. 일반 클러스터 송신자 채널 이름은 클러스터 송신자 채널 이름을 그룹화하는 필터입니다. 예를 들어, SALES.*에서는 이름이 SALES.(으)로 시작하는 모든 클러스터 송신자 채널을 그룹화합니다. 필터 문자열의 어디에나 여러 와일드카드 문자를 넣을 수 있습니다. 와일드카드 문자는 별표("*")입니다. 0(영)부터 임의의 수의 문자를 나타냅니다.

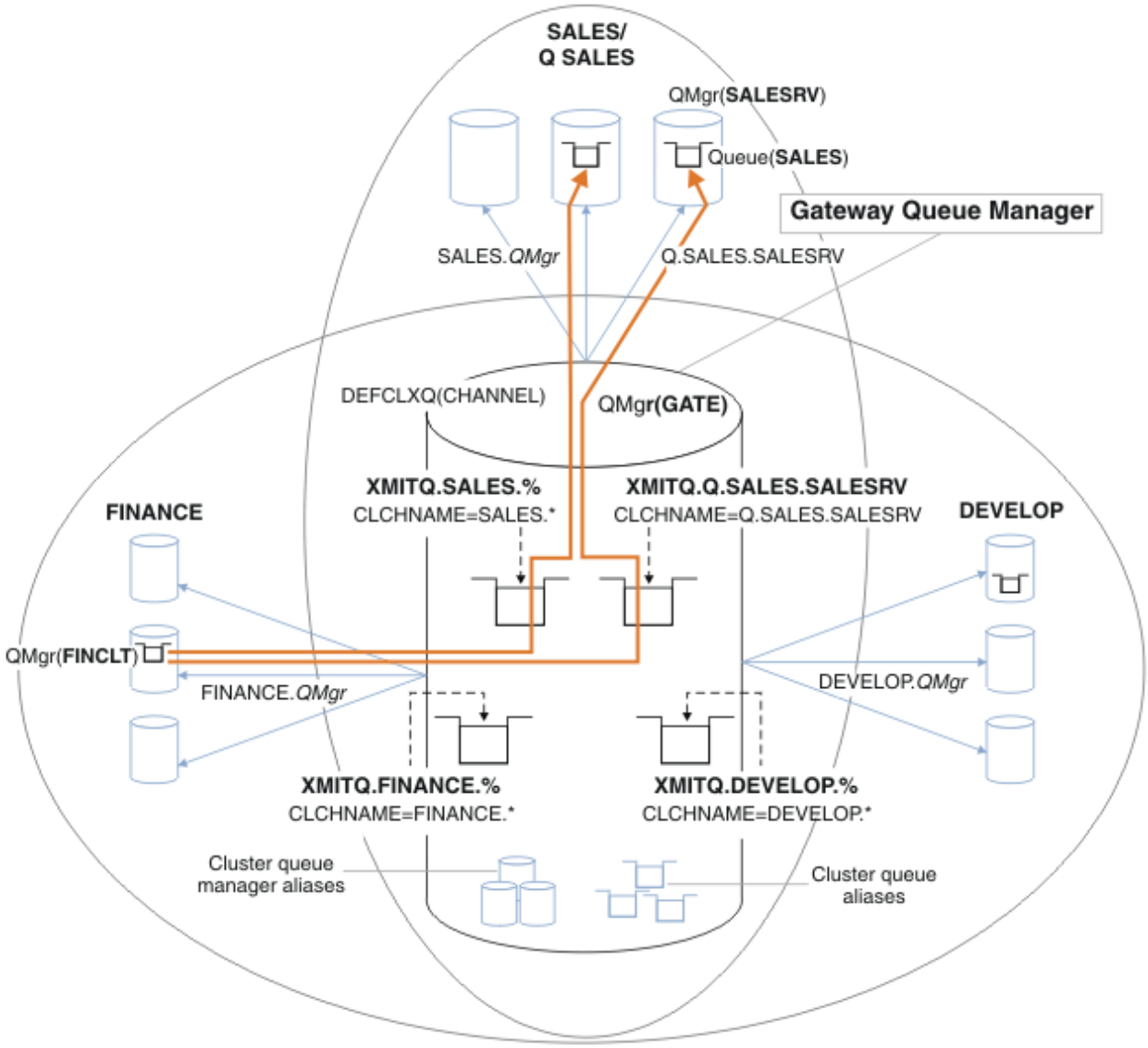


그림 *13.* 다른 부서 *IBM MQ* 클러스터의 특정 전송 큐 예

프로시저

1. 사용할 기본 클러스터 전송 큐의 유형을 선택하십시오.

- 각 클러스터 연결에 대해 단일 클러스터 전송 큐 또는 별도의 큐를 선택하십시오.

기본 설정을 그대로 두거나 **MQSC** 명령을 실행하십시오.

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

2. 다른 플로우와 클러스터 전송 큐를 공유해서는 안되는 메시지 플로우를 분리시키십시오.
   - 48 페이지의 『클러스터링: 다중 클러스터 전송 큐 구성 예』의 내용을 참조하십시오. 이 예에서 분리해야 하는 SALES 큐는 SALESRV에 있는 SALES 클러스터의 멤버입니다. SALES 큐를 분리하려면 새 클러스터 Q.SALES을(를) 작성하고 SALESRV 큐 관리자를 멤버로 설정하고 Q.SALES에 속하는 SALES 큐를 수정하십시오.
   - SALES(으)로 메시지를 전송하는 큐 관리자도 새 클러스터의 멤버여야 합니다. 클러스터된 큐 알리어스 및 게이트웨이 큐 관리자를 사용할 때에는 예에서처럼, 대부분의 경우 게이트웨이 큐 관리자가 새 클러스터 멤버에 수행하는 변경을 제한할 수 있습니다.
   - 하지만 게이트웨이에서 목적지로 향하는 플로우를 분리시키면 소스 큐 관리자에서 게이트웨이로의 플로우가 분리되지 않습니다. 때로는 게이트웨이로 향하는 플로우가 아닌 게이트웨이에서 나오는 플로우만 분리시켜도 충분한 경우가 있습니다. 충분하지 않은 경우에는 소스 큐 관리자를 새 클러스터에 추가하십시오. 메시지가 게이트웨이를 통과하게 하려면 클러스터 알리어스를 새 클러스터로 이동시킨 후 대상 큐 관리자가 아닌 게이트웨이의 클러스터 알리어스로 계속해서 직접 메시지를 보내십시오.

   다음 단계에 따라 메시지 플로우를 분리시키십시오.

   a) 각 대상 큐가 해당 큐 관리자의 특정 클러스터에 있는 유일한 큐가 되도록 플로우 목적지를 구성하십시오.

   b) 체계적인 이름 지정 규칙을 따라 작성한 새 클러스터에 대해 클러스터 송신자 및 클러스터 수신자 채널을 작성하십시오.
      - 38 페이지의 『클러스터링: 중첩 클러스터에 대한 특수 고려사항』의 내용을 참조하십시오.

   c) 대상 큐로 메시지를 보내는 모든 큐 관리자의 격리된 각 목적지마다 클러스터 전송 큐를 정의하십시오.
      - 클러스터 전송 큐의 이름 지정 규칙은 XMITQ. 접두부가 있는 클러스터 채널 이름 속성 CLCHNAME의 값을 사용하는 것입니다.
3. 제어 또는 모니터링 요구사항을 충족하는 클러스터 전송 큐를 작성하십시오.
   - 일반적인 제어 및 모니터링 요구사항은 클러스터당 하나의 전송 큐 또는 큐 관리자당 하나의 전송 큐입니다. 클러스터 채널, *ClusterName. QueueManagerName*의 이름 지정 규칙을 따르는 경우 큐 관리자의 클러스터를 선택하는 일반 채널 이름 또는 큐 관리자가 멤버인 모든 클러스터를 쉽게 작성할 수 있습니다. 48 페이지의 『클러스터링: 다중 클러스터 전송 큐 구성 예』의 내용을 참조하십시오.
   - 퍼센트 부호로 별표 기호를 바꿔서 일반 채널 이름을 제공하도록 클러스터 전송 큐의 이름 지정 규칙을 확장하십시오. 예:

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

관련 개념

클러스터 전송 큐 및 클러스터 송신자 채널에 대한 작업

26 페이지의 『액세스 제어 및 다중 클러스터 전송 큐』
애플리케이션이 리모트 클러스터 큐에 메시지를 넣을 때 검사하는 세 가지 모드 중에서 선택하십시오. 모드에서는 클러스터 큐에 대해 원격으로 검사하거나 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 대해 로컬로 검사하거나 클러스터 큐 또는 클러스터 큐 관리자의 로컬 프로파일에 대해 검사합니다.

32 페이지의 『중첩 클러스터』
중첩 클러스터는 추가 관리 기능을 제공합니다. 이름 목록을 사용하여 중첩 클러스터를 관리하는 데 필요한 명령의 수를 줄이십시오.

관련 태스크

게이트웨이 큐 관리자에서 송신된 메시지를 격리하기 위한 리모트 큐 정의 추가

게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가

게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 및 클러스터 전송 큐 추가

메시지 트래픽을 격리하기 위해 클러스터 전송 큐를 분리하도록 기본값 변경
게이트웨이 큐 관리자로 두 개의 중첩 클러스터 작성
클러스터 간 메시지 경로 구성

사용할 클러스터 전송 큐의 유형 선택 방법
다른 클러스터 전송 큐 구성 옵션 간에 선택 방법.

클러스터-송신자 채널과 연관된 클러스터 전송 큐를 선택할 수 있습니다.

1. 모든 클러스터-송신자 채널을 단일 기본 클러스터 전송 큐 SYSTEM.CLUSTER.TRANSMIT.QUEUE와 연관시킬 수 있습니다. 이 옵션은 기본값입니다.
2. 모든 클러스터 송신자 채널을 개별 클러스터 전송 큐와 자동으로 연관시키도록 설정할 수 있습니다. 큐는 모델 큐 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE의 큐 관리자가 작성하고 이름은 SYSTEM.CLUSTER.TRANSMIT. *ChannelName*입니다. 큐 관리자 속성 **DEFCLXQ**이(가) CHANNEL로 설정된 경우 채널은 고유하게 이름 지정된 클러스터 전송 큐를 사용합니다.
3. 단일 클러스터 전송 큐에서 제공할 특정 클러스터 송신자 채널을 설정할 수 있습니다. 전송 큐를 작성하고 **CLCHNAME** 속성을 클러스터 송신자 채널의 이름으로 설정하여 이 옵션을 선택하십시오.
4. 단일 클러스터 전송 큐에서 제공할 클러스터 송신자 채널 그룹을 선택할 수 있습니다. 전송 큐를 작성하고 **CLCHNAME** 속성을 일반 채널 이름(예: *ClusterName*.*)으로 설정하여 이 옵션을 선택하십시오. 38 페이지의 『클러스터링: 중첩 클러스터에 대한 특수 고려사항』에서 이름 지정 규칙에 따라 클러스터 채널의 이름을 지정한 경우, 이 이름은 클러스터 *ClusterName*의 큐 관리자에 연결된 모든 클러스터 채널을 선택합니다.

일부 클러스터 송신자 채널에 대한 기본 클러스터 전송 큐 옵션 중 하나와 임의의 수의 특정 및 일반 클러스터 전송 큐 구성을 결합할 수 있습니다.

우수 사례

대부분의 경우 기존 IBM MQ 설치에는 기본 구성이 최상의 선택입니다. 클러스터 큐 관리자는 클러스터 메시지를 단일 클러스터 전송 큐 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 저장합니다. 상이한 큐 관리자와 상이한 클러스터에 대한 메시지를 별도의 전송 큐에 저장하거나 자신의 전송 큐를 정의하도록 기본값을 변경하는 선택사항이 있습니다.

대부분의 경우에는 새 IBM MQ 설치에도 기본 구성이 최상의 선택입니다. 기본 구성에서 각 클러스터 송신자 채널마다 전송 큐가 하나씩 있는 대체 구성으로 전환하는 프로세스는 자동입니다. 재전환도 자동입니다. 어느 것을 선택하든 중요하지 않으며 되돌릴 수 있습니다.

다른 구성을 선택하는 이유는 기능성이나 성능보다는 제어 및 관리와 더 관련이 있습니다. 몇 가지 예외를 제외하면, 다수의 전송 큐를 구성해도 큐 관리자 작동에 이점이 없습니다. 큐가 더 많아져서 단일 전송 큐를 참조하는, 이미 설정한 모니터링 및 관리 프로시저를 수정해야 합니다. 이는 다른 선택사항에 대한 강력한 제어 또는 관리 사유가 없으면 모든 것을 감안할 때 기본 구성을 그대로 사용하는 것이 최상의 선택이기 때문입니다.

두 예외 모두 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 저장된 메시지 수가 증가하는 경우 발생하는 상황과 관련이 있습니다. 한 목적지의 메시지를 다른 목적지의 메시지와 분리시키기 위한 모든 단계를 수행한 경우에는 한 목적지에 대한 채널 및 전달 문제가 다른 목적지로의 전달에 영향을 미치면 안됩니다. 그러나 한 대상에 충분히 빠르게 메시지를 전달하지 않아 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 저장된 메시지 수가 증가할 수 있습니다. 한 대상에 대한 SYSTEM.CLUSTER.TRANSMIT.QUEUE의 메시지 수는 다른 대상으로의 메시지 전달에 영향을 줄 수 있습니다.

단일 전송 큐가 가득 차서 발생하는 문제를 방지하려면 구성에 충분한 용량을 빌드하십시오. 그렇게 하면 한 목적지가 실패하고 메시지 백로그가 빌드되기 시작할 경우 문제점을 수정할 시간이 있습니다.

메시지가 허브 큐 관리자(예: 클러스터 게이트웨이)를 통해 라우팅되는 경우, 공통 전송 큐, SYSTEM.CLUSTER.TRANSMIT.QUEUE을(를) 공유합니다. 게이트웨이 큐 관리자의 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 저장된 메시지 수가 최대 용량에 도달하면 깊이가 줄어들 때까지 큐 관리자가 전송 큐에 대한 새 메시지를 거부하기 시작합니다. 게이트웨이를 통해 라우팅된 모든 목적지에 대한 메시지에 정체의 영향이 미칩니다. 메시지는 게이트웨이에 메시지를 보내는 다른 큐 관리자의 전송 큐를 백업합니다. 큐 관리자 오류 로그에 기록된 메시지, 메시지 처리량 감소, 메시지를 송신하고 대상에 메시지가 도착하는 시간 사이의 연장된 경과 시간 등으로 문제점이 드러납니다.

단일 전송 큐의 정체 효과는 큐가 가득 차기 전에도 명백해질 수 있습니다. 일부는 큰 비지속 메시지이고 일부는 작은 메시지가 있는 혼합 메시지 트래픽의 경우 전송 큐가 가득 차면 작은 메시지를 전달하는 시간이 늘어납니다. 이 지연은 일반적으로 디스크에 쓰지 않는 큰 비지속 메시지를 디스크에 쓰기 때문입니다. 시간이 관건인 메시지 플로우의 경우 클러스터 전송 큐를 다른 혼합 메시지 플로우과 공유하면 다른 메시지 플로우와 분리시키기 위해 특수 메시지 경로를 구성할 가치가 있을 수 있습니다(게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 및 클러스터 전송 큐 추가 참조).

별도의 클러스터 전송 큐를 구성하는 다른 이유는 제어 요구사항을 충족시키거나 상이한 클러스터 목적지로 송신되는 메시지 모니터링을 단순화하기 위해서입니다. 예를 들어, 한 목적지에 대한 메시지가 다른 목적지에 대한 메시지와 전송 큐를 결코 공유하지 않음을 증명해야 할 수 있습니다.

기본 클러스터 전송 큐를 제어하는 큐 관리자 속성 **DEFCLXQ**을(를) 변경하여 모든 클러스터 송신자 채널에 사용할 다른 클러스터 전송 큐를 작성하십시오. 여러 목적지가 클러스터 송신자 채널을 공유할 수 있으므로 클러스터가 이 목표를 완전히 충족시키도록 계획해야 합니다. 게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 및 클러스터 전송 큐 추가 메소드를 모든 클러스터 큐에 체계적으로 적용하십시오. 목표로 한 결과는 클러스터 목적지가 클러스터 송신자 채널을 다른 클러스터 목적지와 공유하지 않는 것입니다. 결과적으로 클러스터 목적지에 대한 메시지가 다른 목적지에 대한 메시지와 클러스터 전송 큐를 공유하지 않습니다.

일부 특정 메시지 플로우에 대한 별도의 클러스터 전송 큐를 작성하면 목적지로 향하는 메시지 플로우를 쉽게 모니터할 수 있습니다. 새 클러스터 전송 큐를 사용하려면 큐를 정의하고 클러스터 송신자 채널과 연관시키고 나서 채널을 중지한 후 시작하십시오. 변경사항이 영구적일 필요는 없습니다. 잠시 동안 메시지 플로우를 분리시켜서 전송 큐를 모니터한 후 기본 전송 큐를 다시 사용하도록 되돌릴 수 있습니다.

관련 태스크

클러스터링: 다중 클러스터 전송 큐 구성 예
이 태스크에서는 세 개의 중첩 클러스터에 대한 다중 클러스터 전송 큐를 계획하는 단계를 적용합니다. 요구사항은 한 클러스터 큐로 가는 메시지 플로우를 다른 모든 메시지 플로우와 분리시키고 여러 다른 클러스터에 대한 메시지를 상이한 클러스터 전송 큐에 저장해야 하는 점입니다.

클러스터링: 클러스터 전송 큐 전환
기존 프로덕션 큐 관리자의 클러스터 전송 큐 변경사항을 적용할 방법을 계획합니다.

클러스터링: 다중 클러스터 전송 큐 구성 예
이 태스크에서는 세 개의 중첩 클러스터에 대한 다중 클러스터 전송 큐를 계획하는 단계를 적용합니다. 요구사항은 한 클러스터 큐로 가는 메시지 플로우를 다른 모든 메시지 플로우와 분리시키고 여러 다른 클러스터에 대한 메시지를 상이한 클러스터 전송 큐에 저장해야 하는 점입니다.

이 태스크 정보

이 태스크의 단계는 44 페이지의 『클러스터링: 클러스터 전송 큐를 구성하는 방법을 계획』 의 프로시저를 적용할 방법 및 49 페이지의 그림 14에 표시된 구성에 이르는 방법을 표시합니다. 이는 별도의 클러스터 전송 큐가 구성된 게이트웨이 큐 관리자가 있는 세 개의 중첩 클러스터 예입니다. 클러스터를 정의할 MQSC 명령은 51 페이지의 『예 클러스터 작성』 에 설명되어 있습니다.

예를 들어, 두 개의 요구사항이 있습니다. 하나는 게이트웨이 큐 관리자에서 판매를 기록하는 판매 애플리케이션으로 향하는 메시지 플로우를 분리시키는 것입니다. 두 번째는 임의의 시점에 다른 부서 영역으로의 송신을 대기 중인 메시지 수를 조회하는 것입니다. SALES, FINANCE 및 DEVELOP 클러스터가 이미 정의되어 있습니다. 클러스터 메시지는 현재 SYSTEM.CLUSTER.TRANSMIT.QUEUE에서 전달됩니다.

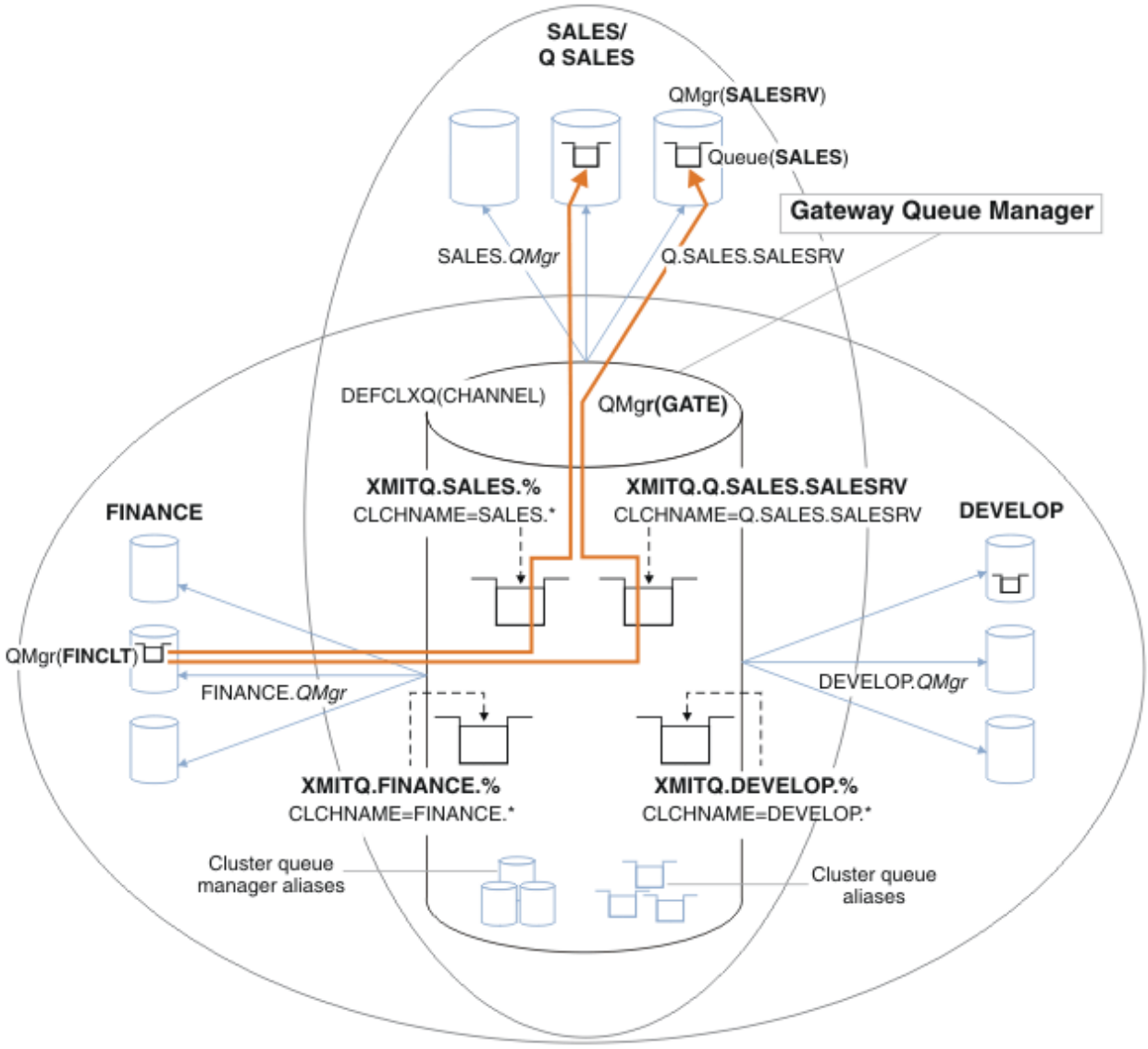


그림 *14.* 다른 부서 *IBM MQ* 클러스터의 특정 전송 큐 예

클러스터를 수정하는 단계는 다음과 같습니다. 정의에 대해서는 새 클러스터에서 판매 큐를 분리하고 게이트웨이 클러스터 전송 큐를 분리하기 위한 변경사항을 참조하십시오.

프로시저

1. 첫 번째 구성 단계는 "사용할 기본 클러스터 전송 큐의 유형을 선택하십시오"입니다.

   결정 사항은 GATE 큐 관리자에서 다음 **MQSC** 명령을 실행하여 별도의 기본 클러스터 전송 큐를 작성하는 것입니다.

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

   클러스터 전송 큐를 수동으로 정의하려고 하기 때문에 이 기본값을 선택할 이유가 많지 않습니다. 선택사항은 약한 진단 값이 있습니다. 수동 정의를 잘못 수행하고 메시지가 아래의 기본 클러스터 전송 큐로 플로우되는 경우 이는 영구 동적 클러스터 전송 큐의 작성으로 나타납니다.
2. 두 번째 구성 단계는 "다른 플로우와 클러스터 전송 큐를 공유해서는 안되는 메시지 플로우를 분리시키십시오"입니다.

   이 경우 SALESRV의 SALES 큐에서 메시지를 수신하는 매출 애플리케이션은 분리해야 합니다. 게이트웨이 큐 관리자에서 나오는 메시지만 격리시키면 됩니다. 세 가지 하위 단계는 다음과 같습니다.

a) " 각 대상 큐가 해당 큐 관리자의 특정 클러스터에 있는 유일한 큐가 되도록 플로우 목적지를 구성하십시오 ".

예제에서는 큐 관리자 SALESRV을(를) 영업 부서의 새 클러스터에 추가해야 합니다. 분리해야 하는 큐가 거의 없는 경우, SALES 큐에 대한 특정 클러스터를 작성하도록 결정할 수 있습니다. 클러스터 이름에 사용 가능한 이름 지정 규칙은 해당 클러스터, Q. *QueueName*의 이름을 지정하는 것입니다(예: Q.SALES). 많은 수의 큐를 격리시켜야 하는 경우에 보다 실제적일 수 있는 대체 방법은 필요한 시기 및 위치에서 격리된 큐의 클러스터를 작성하는 것입니다. 클러스터 이름은 QUEUES. *n*일 수 있습니다.

이 예에서는 새 클러스터를 Q.SALES(이)라고 합니다. 새 클러스터를 추가하려면 새 클러스터에서 판매 큐를 분리하고 게이트웨이 클러스터 전송 큐를 분리하기 위한 변경사항의 정의를 참조하십시오. 정의 변경 요약은 다음과 같습니다.

i) Q.SALES을(를) 저장소 큐 관리자의 클러스터 이름 목록에 추가하십시오. 이름 목록은 큐 관리자 **REPOSNL** 매개변수에서 참조됩니다.

ii) 게이트웨이 큐 관리자의 클러스터 이름 목록에 Q.SALES을(를) 추가하십시오. 이름 목록은 게이트웨이 큐 관리자의 모든 클러스터 큐 알리어스 및 클러스터 큐 관리자 알리어스 정의에 참조됩니다.

iii) 멤버로 속해 있는 두 클러스터 모두의 큐 관리자 SALESRV에 이름 목록을 작성하고 SALES 큐의 클러스터 멤버를 변경하십시오.

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

SALES 큐는 전이의 경우에만 두 클러스터 모두의 멤버입니다. 새 구성이 실행 중이면 SALES 클러스터에서 SALES 큐를 제거합니다. 54 페이지의 그림 15을(를) 참조하십시오.

b) " 체계적인 이름 지정 규칙을 따라 작성한 새 클러스터에 대해 클러스터 송신자 및 클러스터 수신자 채널을 작성하십시오 ".

i) 클러스터 수신자 채널 Q.SALES. *RepositoryQMgr*을(를) 각 저장소 큐 관리자에 추가

ii) 클러스터 전송자 채널 Q.SALES. *OtherRepositoryQMgr*을(를) 각 저장소 큐 관리자에 추가하여 다른 저장소 관리자에 연결하십시오. 이 채널을 시작하십시오.

iii) 클러스터 수신자 채널 Q.SALES.SALESRV 및 Q.SALES.GATE을(를) 실행 중인 저장소 큐 관리자 중 하나에 추가하십시오.

iv) 클러스터 송신자 채널 Q.SALES.SALESRV 및 Q.SALES.GATE을(를) SALESRV 및 GATE 큐 관리자에 추가하십시오. 클러스터 송신자 채널을 클러스터 수신자 채널이 작성되는 저장소 큐 관리자에 연결하십시오.

c) " 대상 큐로 메시지를 보내는 모든 큐 관리자의 격리된 각 목적지마다 클러스터 전송 큐를 정의하십시오 ".

게이트웨이 큐 관리자에서 Q.SALES.SALESRV 클러스터 송신자 채널에 대한 클러스터 전송 큐 XMITQ.Q.SALES.SALESRV을(를) 정의하십시오.

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

3. 세 번째 구성 단계는 "제어 또는 모니터링 요구사항을 충족하는 클러스터 전송 큐를 작성하십시오"입니다.

게이트웨이 큐 관리자에서 클러스터 전송 큐를 정의하십시오.

```
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
```

다음에 수행할 작업

게이트웨이 큐 관리자의 새 구성으로 전환하십시오.

새 채널을 시작하고 현재 상이한 전송 큐와 연관된 채널을 다시 시작하면 전환이 트리거됩니다. 또는 게이트웨이 큐 관리자를 중지한 후 시작할 수 있습니다.

1. 게이트웨이 큐 관리자에서 다음 채널을 중지하십시오.

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

2. 게이트웨이 큐 관리자에서 다음 채널을 시작하십시오.

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

스위치가 완료되면 SALES 클러스터에서 SALES 큐를 제거하십시오. 54 페이지의 그림 15의 내용을 참조하십시오.

관련 개념

사용할 클러스터 전송 큐의 유형 선택 방법
다른 클러스터 전송 큐 구성 옵션 간에 선택 방법.

관련 태스크

클러스터링: 클러스터 전송 큐 전환
기존 프로덕션 큐 관리자의 클러스터 전송 큐 변경사항을 적용할 방법을 계획합니다.

예 클러스터 작성
예제 클러스터를 작성하고 게이트웨이 큐 관리자에서 SALES 큐 및 개별 메시지를 분리하기 위해 수정하는 데 필요한 정의 및 지시사항입니다.

이 태스크 정보

FINANCE, SALES및 Q.SALES 클러스터를 작성하기 위한 전체 **MQSC** 명령은 기본 클러스터에 대한 정의, 새 클러스터에서 판매 큐를 분리하고 게이트웨이 클러스터 전송 큐를 분리하기 위한 변경사항및 판매 클러스터에서 큐 관리자 SALESRV의 판매 큐 제거에서 제공됩니다. DEVELOP 클러스터는 정의를 더 짧게 유지하기 위해 정의에서 생략됩니다.

프로시저

1. SALES과(와) FINANCE 클러스터 및 게이트웨이 큐 관리자를 작성하십시오.
   a) 큐 관리자를 작성하십시오.

   51 페이지의 표 4의 각 큐 관리자 이름에 대해 `crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE` *QmgrName* 명령을 실행하십시오.

표 *4.* 큐 관리자 이름 및 포트 번호

설명	큐 관리자 이름	포트 번호
재무 저장소	FINR1	1414
재무 저장소	FINR2	1415
재무 클라이언트	FINCLT	1418
판매 저장소	SALER1	1416
판매 저장소	SALER2	1417
판매 서버	SALESRV	1419
게이트웨이	GATE	1420

   b) 모든 큐 관리자를 시작하십시오.

51 페이지의 표 4의 각 큐 관리자 이름에 대해 `strmqm` *QmgrName* 명령을 실행하십시오.

c) 각 큐 관리자에 대한 정의를 작성하십시오.

`runmqsc` *QmgrName* < *filename* 명령을 실행하십시오. 여기서 파일은 기본 클러스터에 대한 정의에 나열되며 파일 이름은 큐 관리자 이름과 일치합니다.

기본 클러스터에 대한 정의

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. 샘플 요청 프로그램을 실행하여 구성을 테스트하십시오.
   a) SALESRV 큐 관리자에서 트리거 모니터 프로그램 시작

      Windows에서 명령 창을 열고 `runmqtrm -m SALESRV` 명령 실행

   b) 샘플 요청 프로그램을 실행하고 요청을 송신하십시오.

      Windows에서 명령 창을 열고 `amqsreq A.SALES FINCLT` 명령 실행

      요청 메시지가 다시 에코되고 15초 후에 샘플 프로그램이 완료됩니다.
3. 정의를 작성하여 Q.SALES 클러스터에서 SALES 큐를 분리하고 게이트웨이 큐 관리자에서 SALES 및 FINANCE 클러스터에 대한 클러스터 메시지를 구분하십시오.

   `runmqsc` *QmgrName* < *filename* 명령을 실행하십시오. 여기서 파일은 다음 목록에 나열되며 파일 이름은 큐 관리자 이름과 거의 일치합니다.

   새 클러스터의 판매 큐를 격리시키고 게이트웨이 클러스터 전송 큐를 분리하기 위한 변경사항
   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. SALES 클러스터에서 SALES 큐를 제거하십시오.

   54 페이지의 그림 15의 **MQSC** 명령을 실행하십시오.

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

   그림 *15.* 판매 클러스터에서 큐 관리자 *SALESRV*의 판매 큐 제거

5. 새 전송 큐로 채널을 전환하십시오.

   요구사항은 GATE 큐 관리자가 사용 중인 모든 채널을 중지하고 시작하는 것입니다. 최소의 명령 수로 이를 수행하려면 큐 관리자를 중지한 후 시작하십시오.

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

다음에 수행할 작업

1. 샘플 요청 프로그램을 재실행하여 새 구성 작업을 확인하십시오(53 페이지의 『2』 단계 참조).
2. GATE 큐 관리자의 모든 클러스터 전송 큐를 통해 플로우되는 메시지를 모니터링하십시오.
   a. 각 클러스터 전송 큐 정의를 변경하여 큐 모니터링을 켜십시오.

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

   b. 큐 관리자 통계 모니터링이 OFF인지 확인하여 출력을 최소화하고 모니터링 간격을 낮은 값으로 설정하여 여러 테스트를 편리하게 수행할 수 있습니다.

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

   c. GATE 큐 관리자를 재시작하십시오.
   d. 샘플 요청 프로그램을 몇 번 실행하여 SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV 및 SYSTEM.CLUSTER.TRANSMIT.QUEUE을(를) 통해 동일한 수의 메시지가 플로우되는지 확인하십시오. 요청은 SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV을(를) 통해 플로우되고 SYSTEM.CLUSTER.TRANSMIT.QUEUE을(를) 통해 응답합니다.

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

   e. 두 간격의 결과는 다음과 같습니다.

```
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.
```

하나의 요청 및 응답 메시지가 첫 번째 간격에 송신되고 두 개의 요청 및 응답 메시지가 두 번째 간격에 송신되었습니다. 요청 메시지는 SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV에 있고 응답 메시지는 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 있음을 추론할 수 있습니다.

클러스터링: 클러스터 전송 큐 전환
기존 프로덕션 큐 관리자의 클러스터 전송 큐 변경사항을 적용할 방법을 계획합니다.

시작하기 전에

전환 프로세스가 새 전송 큐에 전송해야 하는 메시지 수를 줄이면 전환이 보다 빨리 완료됩니다. 더 진행하기 전에 전송 큐를 비우려 시도하는 이유에 대해서는 클러스터 송신자 채널을 다른 전송 큐로 전환하는 프로세스의 작동 방식을 읽어보십시오.

이 태스크 정보

클러스터 전송 큐의 변경사항을 적용할 두 가지 방법이 있습니다.

1. 큐 관리자가 자동으로 변경을 수행할 수 있게 하십시오. 기본값입니다. 클러스터 송신자 채널이 다음 번에 시작될 때 큐 관리자가 보류 중인 전송 큐 변경사항이 있는 클러스터 송신자 채널을 전환합니다.
2. 수동으로 변경을 수행하십시오. 클러스터 송신자 채널이 중지되었을 때 이 채널을 변경할 수 있습니다. 클러스터 송신자 채널이 시작되기 전에 한 클러스터 전송 큐에서 다른 클러스터 전송 큐로 이를 전환할 수 있습니다.

둘 중 어느 옵션을 선택할지 결정할 때 어떤 요인을 고려하며 전환을 어떻게 관리합니까?

프로시저

- 옵션 1: 큐 관리자가 자동으로 변경을 수행할 수 있게 하십시오(57 페이지의 『활성 클러스터 송신자 채널을 다른 클러스터 전송 큐 세트로 전환』 참조).

  큐 관리자가 전환하게 하려는 경우 이 옵션을 선택하십시오.

  이 옵션을 설명하는 대체 방법은 큐 관리자에게 채널 중지를 강제 실행하지 않은 채로 클러스터 송신자 채널을 전환하라고 알리는 것입니다. 채널이 중지되도록 강제 실행한 후 채널을 시작해서 즉시 전환이 발생하게 하는 옵션이 있습니다. 옵션 2와 다르게, 전환은 채널이 시작될 때 시작되고 채널이 실행 중인 동안에 실행됩니다. 옵션 2에서는 채널이 중지될 때 전환이 발생합니다.

  전환이 자동으로 발생하도록 이 옵션을 선택하는 경우 클러스터 송신자 채널이 시작될 때 전환 프로세스가 시작합니다. 채널이 중지되지 않은 경우 처리할 메시지가 있으면 채널이 비활성화된 후에 프로세스가 시작합니다. 채널이 중지된 경우 START CHANNEL 명령을 사용하여 시작하십시오.

  채널이 제공하는 전송 큐의 클러스터 송신자 채널에 메시지가 남아 있지 않으면 그 즉시 전환 프로세스가 완료됩니다. 그러한 경우 클러스터 송신자 채널의 새로 도착한 메시지가 새 전송 큐에 직접 저장됩니다. 그 때까지는, 메시지가 이전 전송 큐에 저장되고 전환 프로세스는 이전 전송 큐에서 새 전송 큐로 메시지를 전송합니다. 클러스터 송신자 채널은 전체 전환 프로세스 중 새 클러스터 전송 큐에서 메시지를 전달합니다.

  전환 프로세스가 완료되는 시기는 시스템의 상태에 따라 다릅니다. 유지보수 창에서 변경 중인 경우 전환 프로세스가 제 시간에 완료될지 미리 가늠해보십시오. 제 시간에 완료될지 여부는 이전 전송 큐에서 전송을 대기 중인 메시지 수가 0이 되는지 여부에 따라 다릅니다.

  첫 번째 메소드의 장점은 자동이라는 것입니다. 단점은 구성 변경을 수행하는 시간이 유지보수 창으로 제한된 경우 유지보수 창에서 전환 프로세스를 완료하도록 시스템을 제어할 수 있는 확신이 있어야 한다는 점입니다. 확실하지 않다면 옵션 2가 더 나은 선택일 수 있습니다.
- 옵션 2: 수동으로 변경을 수행하십시오(58 페이지의 『중지된 클러스터 송신자 채널을 다른 클러스터 전송 큐로 전환』 참조).

  전체 전환 프로세스를 수동으로 제어하려는 경우나 비활성 또는 중지된 채널을 전환하려는 경우 이 옵션을 선택하십시오. 몇 개의 클러스터 송신자 채널을 전환 중이며 유지보수 창에서 전환을 수행하려는 경우에 좋은 방법입니다.

이 옵션의 대체 설명은 클러스터 송신자 채널이 중지된 동안에 클러스터 송신자 채널을 전환하는 것입니다.

이 옵션을 선택하면 전환의 발생 시기를 완전히 제어합니다.
유지보수 창에서 정해진 시간 안에 전환 프로세스를 완료한다고 확신할 수 있습니다. 전환 발생 시기는 한 전송 큐에서 다른 전송 큐로 전송해야 하는 메시지 수에 따라 다릅니다. 메시지가 계속 도착하는 경우에는 프로세스가 모든 메시지를 전송하기까지 시간이 걸릴 수 있습니다.
이전 전송 큐에서 메시지를 전송하지 않고 채널을 전환하는 옵션이 있습니다. 이는 "즉석" 전환입니다.
클러스터 송신자 채널을 다시 시작하면 최근에 지정한 전송 큐의 메시지를 채널이 처리하기 시작합니다.

두 번째 방법의 장점은 전환 프로세스를 제어한다는 점입니다. 단점은 전환할 클러스터 송신자 채널을 식별하고, 필요한 명령을 실행하며, 클러스터 송신자 채널이 중지되지 못하게 할 수 있는 인다우트(in-doubt) 채널을 해결해야 하는 점입니다.

관련 개념
사용할 클러스터 전송 큐의 유형 선택 방법
다른 클러스터 전송 큐 구성 옵션 간에 선택 방법.

클러스터 송신자 채널을 다른 전송 큐로 전환하는 프로세스의 작동 방식
관련 태스크
클러스터링: 다중 클러스터 전송 큐 구성 예
이 태스크에서는 세 개의 중첩 클러스터에 대한 다중 클러스터 전송 큐를 계획하는 단계를 적용합니다. 요구사항은 한 클러스터 큐로 가는 메시지 플로우를 다른 모든 메시지 플로우와 분리시키고 여러 다른 클러스터에 대한 메시지를 상이한 클러스터 전송 큐에 저장해야 하는 점입니다.

활성 클러스터 송신자 채널을 다른 클러스터 전송 큐 세트로 전환
이 태스크는 활성 클러스터 송신자 채널을 전환하기 위한 세 가지 옵션을 제공합니다. 한 옵션은 큐 관리자가 자동으로 전환하게 하는 방법으로, 실행 중인 애플리케이션에 영향을 주지 않습니다. 다른 옵션은 수동으로 채널을 중지한 후 시작하거나 큐 관리자를 다시 시작하는 방법입니다.

시작하기 전에

클러스터 전송 큐 구성을 변경하십시오. **DEFCLXQ** 큐 관리자 속성을 변경하거나 전송 큐의 **CLCHNAME** 속성을 추가 또는 수정할 수 있습니다.

전환 프로세스가 새 전송 큐에 전송해야 하는 메시지 수를 줄이면 전환이 보다 빨리 완료됩니다. 더 진행하기 전에 전송 큐를 비우려 시도하는 이유에 대해서는 클러스터 송신자 채널을 다른 전송 큐로 전환하는 프로세스의 작동 방식을 읽어보십시오.

이 태스크 정보

이 태스크의 단계를 클러스터 전송 큐 구성 변경을 수행하기 위한 계획을 세우는 기반으로 삼으십시오.

프로시저

1. 옵션: 현재 채널 상태를 기록하십시오.

   클러스터 전송 큐를 서비스 중인 현재 및 저장된 채널의 상태를 기록하십시오. 다음 명령은 시스템 클러스터 전송 큐와 연관된 상태를 표시합니다. 정의한 클러스터 전송 큐와 연관된 상태를 표시하려면 명령을 직접 추가하십시오. XMITQ. *ChannelName*와 같은 규칙을 사용하여 해당 전송 큐에 대한 채널 상태를 쉽게 표시할 수 있도록 정의하는 클러스터 전송 큐의 이름을 지정하십시오.

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

2. 전송 큐를 전환하십시오.

   - 아무 것도 하지 마십시오. 큐 관리자가 중지 또는 비활성화된 후 다시 시작할 때 클러스터 송신자 채널을 전환합니다.

     큐 관리자 구성 변경에 대한 관심 또는 규칙이 없으면 이 옵션을 선택하십시오. 실행 중인 애플리케이션은 변경의 영향을 받지 않습니다.

- 큐 관리자를 재시작하십시오. 모든 클러스터 송신자 채널은 중지되고 요구가 있으면 자동으로 다시 시작됩니다.

  모든 변경사항을 즉시 시작하려면 이 옵션을 선택하십시오. 큐 관리자가 시스템 종료된 후 재시작할 때 실행 애플리케이션을 인터럽트합니다.
- 개별 클러스터-송신자 채널을 중지한 후 재시작하십시오.

  몇 개의 채널을 즉시 변경하려면 이 옵션을 선택하십시오. 메시지 채널을 중지한 후 다시 시작하는 사이에 실행 중인 애플리케이션은 메시지 전송 중 잠시 지연됩니다. 클러스터 송신자 채널은 중지된 시간을 제외하고, 실행 상태를 유지합니다. 전환 프로세스 중 메시지는 이전 전송 큐로 전달되고, 전환 프로세스를 통해 새 전송 큐에 전송되며, 클러스터 송신자 채널을 통해 새 전송 큐에서 전달됩니다.

3. 옵션: 전환하는 채널을 모니터하십시오.

   전환 중 채널 상태 및 전송 큐 용량을 표시하십시오. 다음 예는 시스템 클러스터 전송 큐의 상태를 표시합니다.

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

4. 옵션: 큐 관리자 오류 로그에 기록되는 AMQ7341 메시지인 채널 *ChannelName*에 대한 전송 큐가 큐 *QueueName*에서 *QueueName*으로 전환되었습니다. 를 모니터링하십시오.

중지된 클러스터 송신자 채널을 다른 클러스터 전송 큐로 전환
수동으로 변경하도록 선택하는 경우, 클러스터 송신자 채널이 중지될 때 이 채널을 변경하고 클러스터 송신자 채널이 시작되기 전에 한 클러스터 전송 큐에서 다른 클러스터 전송 큐로 이를 전환할 수 있습니다.

시작하기 전에

몇 가지 구성 변경을 수행했으며 이제 영향이 미치는 클러스터 송신자 채널을 시작하지 않은 채로 변경사항을 적용시키려 합니다. 또는 필요한 구성 변경을 태스크의 단계 중 하나로 수행합니다.

전환 프로세스가 새 전송 큐에 전송해야 하는 메시지 수를 줄이면 전환이 보다 빨리 완료됩니다. 더 진행하기 전에 전송 큐를 비우려 시도하는 이유에 대해서는 클러스터 송신자 채널을 다른 전송 큐로 전환하는 프로세스의 작동 방식을 읽어보십시오.

이 태스크 정보

이 태스크는 비활성 또는 중지된 클러스터 송신자 채널에 제공된 전송 큐를 전환합니다. 클러스터 송신자 채널이 중지되어서 전송 큐를 부시 전환하려 하기 때문에 이 태스크를 수행할 수 있습니다. 예를 들어, 어떠한 이유로 클러스터 송신자 채널이 시작되지 않거나 일부 다른 구성 문제가 있는 경우입니다. 문제를 해결하려면 클러스터 송신자 채널을 작성하고 이전 클러스터 송신자 채널의 전송 큐를 새 클러스터 송신자 채널의 전송 큐와 연관시키도록 결정합니다.

더 가능성 있는 시나리오는 클러스터 전송 큐의 재구성이 수행되는 시기를 제어하려는 경우입니다. 재구성을 완전히 제어하려면 채널을 중지하고 구성을 변경한 후 전송 큐를 전환시킵니다.

프로시저

1. 전환하려는 채널을 중지하십시오.

   a) 전환하려는 비활성 또는 실행 중인 채널을 중지하십시오. 비활성 클러스터 송신자 채널을 중지하면 구성 변경을 수행하는 동안에 이 채널이 시작되지 않습니다.

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

2. 옵션: 구성 변경을 수행하십시오.

   예를 들어, 48 페이지의 『클러스터링: 다중 클러스터 전송 큐 구성 예』 을(를) 참조하십시오.
3. 클러스터 송신자 채널을 새 클러스터 전송 큐로 전환하십시오.

Multi 멀티플랫폼에서 다음 명령을 실행하십시오.

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

z/OS z/OS에서는 CSQUTIL 명령의 SWITCH 함수를 사용하여 메시지를 전환하거나 발생하는 사항을 모니터하십시오. 다음 명령을 사용하십시오.

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

자세한 정보는 SWITCH 함수를 참조하십시오.

runswchl 또는 CSQUTIL SWITCH 명령은 이전 전송 큐의 메시지를 새 전송 큐로 전송합니다. 이 채널에 대한 이전 전송 큐의 메시지 수가 0이 되면 전환이 완료됩니다. 명령은 동기식입니다. 명령은 전환 프로세스 중 창에 진행 메시지를 씁니다.

전송 단계 중 클러스터 송신자 채널로 예정된 기존 및 새 메시지는 새 전송 큐에 순서대로 전송됩니다.

클러스터 송신자 채널이 중지되었기 때문에 새 전송 큐에 메시지가 빌드됩니다. 중지된 클러스터 송신자 채널을 57 페이지의 『활성 클러스터 송신자 채널을 다른 클러스터 전송 큐 세트로 전환』의 57 페이지의 『2』 단계와 대조시키십시오. 이 단계에서는 클러스터 송신자 채널이 실행 중이므로 새 전송 큐에 메시지가 반드시 빌드되지는 않습니다.

4. 옵션: 전환하는 채널을 모니터하십시오.

   다른 명령 창에서 전환 중의 전송 큐 용량을 표시하십시오. 다음 예는 시스템 클러스터 전송 큐의 상태를 표시합니다.

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

5. 옵션: 큐 관리자 오류 로그에 기록되는 AMQ7341 메시지인 채널 *ChannelName*에 대한 전송 큐가 큐 *QueueName*에서 *QueueName*으로 전환되었습니다.를 모니터링하십시오.
6. 중지한 클러스터 송신자 채널을 재시작하십시오.

   채널이 중지되어 STOPPED 상태에 있으므로 자동으로 시작되지 않습니다.

   ```
   START CHANNEL(ChannelName)
   ```

관련 참조

runswchl

RESOLVE CHANNEL

STOP CHANNEL

클러스터링: 마이그레이션 및 수정 우수 사례

이 주제는 IBM MQ 클러스터 계획 및 관리를 위한 지침을 제공합니다. 이 정보는 고객의 피드백과 테스트에 기반을 둔 지침입니다.

1. 59 페이지의 『클러스터의 오브젝트 이동』(IBM MQ의 새 버전이나 수정팩을 설치하지 않은 채, 클러스터 내에서 오브젝트를 이동시키는 우수 사례).
2. 61 페이지의 『업그레이드 및 유지보수 설치』(유지보수 또는 업그레이드를 적용하고 새 아키텍처를 테스트하는 동안 작업 중인 클러스터 아키텍처를 계속 가동하고 실행하는 우수 사례).

클러스터의 오브젝트 이동

애플리케이션 및 큐

하나의 큐 관리자에서 호스트되는 큐 인스턴스를 다른 큐 관리자로 이동해야 할 때에는 워크로드 밸런싱 매개변수에 대해 작업하여 원활하게 전이할 수 있습니다.

새로 호스트할 큐의 인스턴스를 작성하되, 클러스터 워크로드 밸런싱 설정을 사용하여 애플리케이션이 전환할 준비가 될 때까지 원래 인스턴스로 계속 메시지를 보내십시오. 이를 위해 다음 단계를 따르십시오.

1. 기존 큐의 **CLWLRANK** 특성을 높은 값(예: 5)으로 설정하십시오.
2. 큐의 새 인스턴스를 작성하고 해당 **CLWLRANK** 특성을 0으로 설정하십시오.
3. 큐의 새 인스턴스에 대해 이용 애플리케이션을 배치하고 시작하는 것과 같은 새 시스템의 추가 구성을 완료하십시오.
4. 새 큐 인스턴스의 **CLWLRANK** 특성을 원래 인스턴스보다 높게 설정하십시오(예: 9).
5. 원래 큐 인스턴스가 시스템의 큐에 대기된 메시지를 처리하도록 한 후 큐를 삭제하십시오.

전체 큐 관리자 이동

큐 관리자가 동일한 호스트에 있지만 IP 주소가 변경되는 경우 프로세스는 다음과 같습니다.

- 제대로 사용할 경우 DNS가 프로세스를 단순화하는 데 유용할 수 있습니다. 연결 이름(CONNAME) 채널 속성을 DNS를 사용하는 방법에 대한 정보는 ALTER CHANNEL을 참조하십시오.
- 전체 저장소를 이동하는 경우 변경을 수행하기 전에 최소 하나의 다른 전체 저장소가 순조롭게(예를 들어, 채널 상태에 대한 문제 없이) 실행 중인지 확인하십시오.
- 트래픽이 빌드되지 않도록 SUSPEND QMGR 명령을 사용하여 큐 관리자를 일시중단하십시오.
- 컴퓨터의 IP 주소를 수정하십시오. CLUSRCVR 채널 정의에 CONNAME 필드의 IP 주소가 사용되는 경우에는 이 IP 주소 항목을 수정하십시오. 어디에서나 이 업데이트를 사용할 수 있도록 DNS 캐시를 비워야 할 수 있습니다.
- 큐 관리자가 전체 저장소에 다시 연결할 때 채널 자동 정의가 자동으로 자체 분석됩니다.
- 큐 관리자가 전체 저장소를 호스트했으며 IP 주소가 변경되면 수동으로 정의된 CLUSSDR 채널을 새 위치로 가리키기 위해 일부분이 가능한 빨리 전환되는지 확인하는 것이 중요합니다. 이 전환이 이행될 때까지는 이 큐 관리자가 나머지(변경되지 않은) 전체 저장소에만 접속할 수 있으며 올바르지 않은 채널 정의에 관한 경고 메시지가 표시될 수 있습니다.
- RESUME QMGR 명령을 사용하여 큐 관리자를 재개하십시오.

큐 관리자를 새 호스트로 이동시켜야 하는 경우 큐 관리자 데이터를 복사하고 백업에서 복원할 수 있습니다. 하지만 다른 옵션이 있을 경우 이 프로세스는 권장되지 않습니다. 이전 절에 설명된 대로 새 시스템에 큐 관리자를 작성하고 큐와 애플리케이션을 복제하는 것이 더 나을 수도 있습니다. 이렇게 하면 롤오버/롤백 메커니즘이 순조롭게 작동합니다.

백업을 사용하여 전체 큐 관리자를 이동시키기로 판별된 경우 다음 우수 사례를 따르십시오.

- 운영 체제 환경에 적합한 시스템 복구에 일반적으로 사용하는 프로세스를 적용하여, 백업에서 큐 관리자를 복원하는 것으로 전체 프로세스를 처리하십시오.
- 마이그레이션 후 **REFRESH CLUSTER** 명령을 사용하여 로컬에 보유한 모든 정보를(인다우트(in-doubt) 상태인 자동 정의된 채널을 포함하여) 제거하고 재빌드되도록 강제 실행하십시오.

  참고: 대형 클러스터의 경우, **REFRESH CLUSTER** 명령을 사용하면 진행 중에 클러스터에 혼란을 줄 수 있으며, 클러스터 오브젝트가 모든 관심 있는 큐 관리자에 자동으로 상태 업데이트를 보낸 이후 27일 간격으로 다시 수행됩니다. 대형 클러스터를 새로 고치면 클러스터의 성능 및 가용성에 영향을 줄 수 있음을 참조하십시오.

큐 관리자를 작성하고 클러스터의 기존 큐 관리자에서 설정을 복제할 때에는(이 주제에서 이전에 설명한 대로) 두 개의 상이한 큐 관리자를 실제로 동일한 큐 관리자인 것처럼 처리하지 마십시오. 특히, 새 큐 관리자에 동일한 큐 관리자 이름 및 IP 주소를 제공하지 마십시오. 대체 큐 관리자를 '드롭 인'하려는 시도가 IBM MQ 클러스터 문제의 원인인 경우가 빈번합니다. 캐시는 **QMID** 속성을 포함한 업데이트를 수신할 것으로 예상하며 상태가 손상될 수 있습니다.

두 개의 다른 큐 관리자가 실수로 같은 이름으로 작성된 경우, RESET CLUSTER **QMID** 명령을 사용하여 클러스터에서 올바르지 않은 항목을 꺼내는 것이 좋습니다.

업그레이드 및 유지보수 설치

빅뱅 시나리오(예를 들어, 모든 클러스터 및 큐 관리자 활동을 중지하고 모든 큐 관리자에 모든 업그레이드 및 유지보수를 적용한 후 모든 것을 동시에 시작함)를 피하십시오. 클러스터는 큐 관리자의 여러 버전이 공존하는 상황에서도 여전히 작동하도록 설계되어 있으므로 계획이 잘 짜여진 단계별 유지보수 접근법을 권장합니다.

백업을 계획하십시오.

- 백업을 했습니까?
- 새 클러스터 기능을 바로 사용하지 마십시오. 모든 큐 관리자가 새 레벨로 업그레이드되었음이 확실하고 어느 큐 관리자도 다시 롤백하지 않을 것이란 확신이 있을 때까지 대기하십시오. 일부 큐 관리자가 여전히 이전 레벨인 클러스터에서 새 클러스터 기능을 사용하면 정의되지 않은 작동이 발생할 수 있습니다.

  저장소는 수신하는 레코드를 자신의 버전에 저장합니다. 수신하는 레코드가 나중 버전인 경우 그 레코드가 저장될 때 해당 나중 버전 속성은 제거됩니다. IBM MQ 9.4 큐 관리자에 대한 정보를 수신하는 IBM MQ 9.3 큐 관리자는 IBM MQ 9.3 정보만 저장합니다. IBM MQ 9.3 레코드를 수신하는 IBM MQ 9.4 저장소는 이후 버전에 도입된 속성의 기본값을 저장합니다. 기본값은 수신되는 레코드에 포함되지 않는 속성의 값을 정의합니다.

전체 저장소를 먼저 마이그레이션하십시오. 전체 저장소는 이해하지 않는 정보를 전달할 수 있지만 지속할 수 없으므로 절대적으로 필요한 경우가 아니면 이는 권장하는 접근법이 아닙니다. 자세한 정보는 큐 관리자 클러스터 마이그레이션을 참조하십시오.

클러스터링*: REFRESH CLUSTER* 사용 우수 사례
REFRESH CLUSTER 명령을 사용하여 로컬에 보유된 클러스터에 대한 모든 정보를 제거하고 클러스터의 전체 저장소에서 이 정보를 다시 빌드합니다. 예외 상황을 제외하고는 이 명령을 사용해서는 안됩니다. 사용해야 하는 경우 사용 방법에 대한 특수 고려사항이 있습니다. 이 정보는 고객의 피드백과 테스트에 기반을 둔 지침입니다.

필요한 경우 REFRESH CLUSTER만 실행

IBM MQ 클러스터 기술은 클러스터된 큐의 변경과 같은 클러스터 구성 변경을 해당 정보를 알아야 하는 클러스터의 멤버에게 자동으로 알릴 수 있습니다. 이 정보를 전파하기 위해 추가로 관리 단계를 수행할 필요가 없습니다.

애플리케이션이 처음으로 열기를 시도할 때 클러스터의 다른 큐 관리자가 클러스터된 큐를 알지 못하는 경우처럼, 이러한 정보가 필요로 하는 클러스터의 큐 관리자에 도달하지 않는 경우 이는 클러스터 인프라에 문제가 있음을 암시합니다. 예를 들어, 큐 관리자와 전체 저장소 큐 관리자 간에 채널을 시작하지 못할 수 있습니다. 따라서 불일치가 관찰되는 상황을 조사해야 합니다. 가능하면 **REFRESH CLUSTER** 명령을 사용하지 않고 상황을 해결하십시오.

제품 문서에 설명된 드문 경우 또는 IBM 지원에서 요청할 때에는, **REFRESH CLUSTER** 명령을 사용하여 로컬에 보유된 클러스터에 대한 모든 정보를 제거하고 클러스터의 전체 저장소에서 이 정보를 다시 빌드할 수 있습니다.

대형 클러스터를 새로 고치면 클러스터의 성능 및 가용성에 영향을 줄 수 있음

REFRESH CLUSTER 명령은 진행 중인 클러스터에 지장을 줄 수 있습니다. 예를 들어, 전체 저장소가 큐 관리자 클러스터 자원의 재전파를 처리할 때 전체 저장소에 대한 작업을 급격히 증가시킵니다. 대형 클러스터(즉, 수백 개의 큐 관리자가 있는)를 새로 고치는 경우 가능하면 일상적인 작업에 명령을 사용해서는 안되며 대체 메소드를 사용하여 특정 불일치를 정정하십시오. 예를 들어, 클러스터 큐가 클러스터에서 제대로 전파되지 않으면 클러스터된 큐 정의 업데이트(정의의 설명 변경과 같은)의 초기 조사 기술을 통해 클러스터에 큐 구성을 재전파합니다. 이 프로세스는 문제점을 식별하고 잠재적으로 일시적 불일치를 해결하는 데 도움이 될 수 있습니다.

대체 메소드를 사용할 수 없으며 대형 클러스터에서 **REFRESH CLUSTER**를 실행해야 하는 경우에는 최대 사용 시간에 또는 유지보수 창에서 실행해야 사용자 워크로드에 대한 영향을 피할 수 있습니다. 단일 배치에서도 대형 클러스터의 새로 고치기를 피해야 하며 대신에 61 페이지의 『클러스터 오브젝트가 자동 업데이트를 보낼 때 성능 및 가용성 문제 피하기』에 설명된 대로 활동의 시차를 두십시오.

클러스터 오브젝트가 자동 업데이트를 보낼 때 성능 및 가용성 문제 피하기

큐 관리자에 새 클러스터 오브젝트가 정의되고 나면 이 오브젝트에 대한 업데이트가 정의한 순간부터 27일마다 생성되어 클러스터의 모든 전체 저장소 및 관심이 있는 큐 관리자에게 송신됩니다. **REFRESH CLUSTER** 명령을

큐 관리자에게 발행하면 지정된 클러스터에 로컬로 정의된 모든 오브젝트에서 이 자동 업데이트를 위한 시계가 재설정됩니다.

단일 배치에서 또는 구성 백업으로부터 시스템을 재작성하는 것과 같은 상황에서 대형 클러스터(즉, 수백 개의 큐 관리자)를 새로 고치면 27일 후에 모든 큐 관리자가 동시에 전체 오브젝트 정의를 전체 저장소에 다시 광고합니다. 이로 인해 시스템이 다시 상당히 느려지거나 모든 업데이트가 완료될 때까지 사용 불가능하게 될 수도 있습니다. 따라서 대형 클러스터에서 다수의 큐 관리자를 새로 고치거나 재작성해야 할 때에는 후속 자동 업데이트가 정기적으로 시스템 성능에 영향을 주지 않도록 몇 시간 또는 며칠 간의 활동의 시차를 두어야 합니다.

시스템 클러스터 실행 기록 큐

REFRESH CLUSTER가 수행되면 큐 관리자는 새로 고치기 이전의 클러스터 상태 스냅샷을 작성하고 큐 관리자에 정의된 경우 SYSTEM.CLUSTER.HISTORY.QUEUE(SCHQ)에 스냅샷을 저장합니다. 이 스냅샷은 추후에 시스템에 관한 문제가 발생할 경우 IBM 서비스 용도로만 사용됩니다.

SCHQ는 기본적으로 시동 시 분산 큐 관리자에 정의됩니다. z/OS 마이그레이션의 경우 SCHQ를 수동으로 정의해야 합니다.

SCHQ에 대한 메시지는 3개월 후에 만기됩니다.

관련 개념

93 페이지의 『발행/구독 클러스터에 대한 REFRESH CLUSTER 고려사항』

REFRESH CLUSTER 명령을 발행하면 큐 관리자가 클러스터 토픽 및 연관된 프록시 구독을 포함하여, 로컬에 보유된 클러스터에 대한 정보를 일시적으로 제거합니다.

관련 참조

REFRESH CLUSTER를 실행할 때 표시되는 애플리케이션 문제

MQSC 명령 참조: REFRESH CLUSTER

클러스터링: 가용성, 다중 인스턴스, 재해 복구

이 주제는 IBM MQ 클러스터 계획 및 관리를 위한 지침을 제공합니다. 이 정보는 고객의 피드백과 테스트에 기반을 둔 지침입니다.

IBM MQ 클러스터링 자체는 고가용성 솔루션이 아니지만 일부 상황에서 IBM MQ를 사용하여(예를 들어, 큐의 여러 인스턴스를 각기 다른 큐 관리자에 배치하여) 서비스 가용성을 개선할 수 있습니다. 이 절은 IBM MQ 인프라를 이러한 아키텍처에 사용할 수 있도록 가능한 고가용 상태가 되게 하기 위한 지침을 제공합니다.

참고: IBM MQ에 기타 고가용성 및 재해 복구 솔루션을 사용할 수 있습니다. 고가용성, 복구 및 다시 시작 구성을 참조하십시오.

클러스터 자원의 가용성

두 개의 전체 저장소를 유지보수하도록 일반적으로 권장하는 이유는 한 전체 저장소가 유실되어도 클러스터의 원활한 실행에 문제가 되지 않기 때문입니다. 둘 다 사용할 수 없게 되는 경우에도 부분 저장소에 기존 정보를 보유할 60일의 유예 기간이 있지만 이 경우에는 새 자원이나 이전에 액세스하지 않은 자원(예: 큐)을 사용할 수 없습니다.

클러스터를 사용한 애플리케이션 가용성 개선

클러스터는 큐와 애플리케이션의 여러 인스턴스를 사용하여 고가용성 애플리케이션(예를 들어, 요청/응답 유형 서버 애플리케이션)을 설계하는 데 도움을 줄 수 있습니다. 큐 관리자나 채널이 사용 불가능한 경우가 아니면 필요에 따라 우선순위 속성이 '라이브' 애플리케이션에 우선권을 부여할 수 있습니다. 이는 문제가 발생할 때 새 메시지 처리를 계속하도록 빠르게 전환시키는 데 있어서 강력한 요인이 됩니다.

하지만 클러스터의 특정 큐 관리자에 전달된 메시지는 해당 큐 인스턴스에만 보유되고 이 큐 관리자가 복구될 때까지는 처리할 수 없습니다. 이러한 이유로 데이터 고가용성을 위해서는 다중 인스턴스 큐 관리자와 같은 다른 기술을 고려해 볼 수 있습니다.

다중 인스턴스 큐 관리자

소프트웨어 고가용성(다중 인스턴스)은 기존 메시지를 사용 가능하게 유지하기 위한 기본 제공 오퍼링입니다. 자세한 정보는 고가용성 구성의 IBM MQ 사용, 다중 인스턴스 큐 관리자 작성 및 다음 절을 참조하십시오. 클러스터의 모든 큐 관리자가 최소 IBM WebSphere MQ 7.0.1에서 실행 중이면 이 기술을 사용하여 클러스터의 큐 관리자를 고가용성 상태가 되게 할 수 있습니다. 이전 레벨인 클러스터의 큐 관리자가 있는 경우 보조 IP로 장애 복구하면 다중 인스턴스 큐 관리자와의 연결이 유실될 수 있습니다.

이 주제에서 이전에 논의한 바와 같이, 두 개의 전체 저장소가 구성되어 있으면 그 자체로 거의 고가용성 상태가 됩니다. 필요한 경우 IBM MQ 소프트웨어 고가용성/다중 인스턴스 큐 관리자를 전체 저장소에 사용할 수 있습니다. 이 방법을 사용해야 하는 강력한 이유는 없으며 사실, 일시적 가동 중단의 경우 이 방법은 장애 복구 중에 추가 성능 비용을 유발할 수 있습니다. 예를 들어, 단일 채널 가동 중단이 발생할 경우 반드시 장애 복구할 필요는 없지만 부분 저장소가 클러스터 자원을 조회할 수 없는 상태가 되므로, 두 개의 부분 저장소를 실행하는 대신 소프트웨어 HA를 사용하는 것은 권장되지 않습니다.

재해 복구

큐 관리자의 데이터를 저장하는 디스크가 손상되었을 때의 복구처럼 재해 복구는 잘 수행하기 어려운 작업입니다. IBM MQ가 이를 도울 수 있지만 자동으로 수행할 수는 없습니다. IBM MQ의 유일하게 '진정한' 재해 복구 옵션은(운영 체제 또는 기타 기본 복제 기술을 제외하고) 백업에서 복원하는 것입니다. 다음 상황에서 고려할 클러스터별 몇 가지 사항이 있습니다.

- 재해 복구 시나리오를 테스트할 때 주의하십시오. 예를 들어, 백업 큐 관리자의 조작을 테스트하는 경우 동일한 네트워크에서 온라인으로 불러올 때 주의하십시오. 우발적으로 라이브 클러스터를 조인하여 라이브 클러스터 큐 관리자와 동일하게 이름 지정된 큐를 호스트해서 메시지 '도용'을 시작할 수 있기 때문입니다.
- 재해 복구 테스트 시 실행 중인 라이브 클러스터를 간섭하면 안됩니다. 간섭을 피하기 위한 기법은 다음과 같습니다.
  - 완전한 네트워크 분리 또는 방화벽 레벨의 분리.
  - z/OS 채널 시작 또는 z/OS **chinit** 주소 공간을 시작하지 않음.
  - 실제 재해 복구 시나리오가 발생할 때까지 또는 발생하지 않으면 재해 복구 시스템에 라이브 TLS 인증서를 발행하지 않습니다.
- 클러스터의 큐 관리자 백업을 복원하는 경우 백업이 나머지 클러스터와 비동기화될 수 있습니다. **REFRESH CLUSTER** 명령은 업데이트를 해결하고 클러스터와 동기화하지만 **REFRESH CLUSTER** 명령을 최후의 수단으로 사용해서는 안됩니다. 61 페이지의 『클러스터링: REFRESH CLUSTER 사용 우수 사례』의 내용을 참조하십시오. 명령을 사용하기 전에 간단한 단계가 누락된 것이 아닌지 확인하려면 내부 프로세스 문서 및 IBM MQ 문서를 검토하십시오.
- 복구를 위해서는 애플리케이션이 재실행과 데이터 손실을 처리해야 합니다. 큐를 알려진 상태로 지울지 또는 재생을 관리할 충분한 정보가 어딘가에 있는지 결정해야 합니다.

분산 발행/구독 네트워크 계획

한 큐 관리자에 작성된 구독이 네트워크의 다른 큐 관리자에 연결된 애플리케이션이 발행한 일치하는 메시지를 수신할 큐 관리자 네트워크를 작성할 수 있습니다. 적합한 토폴로지를 선택하려면 수동 제어, 네트워크 크기, 변경 빈도, 가용성, 확장성에 대한 요구사항을 고려해야 합니다.

시작하기 전에

이 태스크는 사용자가 분산 발행/구독 네트워크의 개념과 작동 방식을 이해하고 있다고 가정합니다. 기술 개요는 분산 발행/구독 네트워크를 참조하십시오.

이 태스크 정보

발행/구독 네트워크에 적합한 세 가지 기본 토폴로지가 있습니다.

- 직접 라우트 클러스터
- 토픽 호스트 라우트 클러스터
- 계층 구조

처음 두 개의 토폴로지는 시작점이 IBM MQ 클러스터 구성입니다. 세 번째 토폴로지는 클러스터를 포함하거나 제외하고 작성할 수 있습니다. 기본 큐 관리자 네트워크 계획에 대한 정보는 18 페이지의 『분산 큐와 클러스터 계획』의 내용을 참조하십시오.

직접 라우트 클러스터는 클러스터가 아직 존재하지 않을 때 구성할 가장 단순한 토폴로지입니다. 큐 관리자에 정의하는 토픽이 클러스터의 모든 큐 관리자에 자동으로 사용 가능하며, 발행 애플리케이션이 연결하는 모든 큐 관리자로부터 일치하는 구독이 존재하는 각 큐 관리자로 발행이 직접 라우팅됩니다. 이러한 구성의 단순성은 클러스터의 모든 큐 관리자 간에 상위 레벨의 정보 및 연결 공유를 유지보수하는 IBM MQ에 의존합니다. 작고 단순한 네트워크(즉, 큐 관리자의 수가 적고 발행자 및 구독자 세트가 상당히 정적임)에서 이는 허용 가능합니다. 하지만

크거나 보다 동적인 환경에서 사용할 경우 오버헤드가 매우 높을 수 있습니다. 68 페이지의 『발행/구독 클러스터의 직접 라우팅』의 내용을 참조하십시오.

토픽 호스트 라우트 클러스터는 클러스터의 큐 관리자에 정의하는 토픽을 클러스터의 모든 큐 관리자에 자동으로 사용 가능하게 해서 직접 라우트 클러스터와 동일한 혜택을 제공합니다. 하지만 토픽 호스트 라우트 클러스터의 경우 이 토픽에 대한 모든 정보 및 발행물이 해당 토픽 호스트 큐 관리자를 통과하기 때문에 각 토픽을 호스트하는 큐 관리자를 주의해서 선택할 필요가 있습니다. 이는 시스템이 모든 큐 관리자 간의 채널 및 정보 플로우를 유지보수할 필요가 없음을 의미합니다. 하지만 이는 구독자에게 발행물을 더 이상 직접 보낼 수 없고 토픽 호스트 큐 관리자를 통해 라우팅할 수 있음을 의미하기도 합니다. 이러한 이유로 특히 토픽을 호스트하는 큐 관리자에 시스템의 추가 로드가 있을 수 있으므로 토폴로지 계획 시 주의해야 합니다. 이 토폴로지는 많은 큐 관리자가 있거나 발행자 및 구독자의 동적 세트(즉, 발행자나 구독자가 자주 추가 또는 제거됨)를 호스트하는 네트워크에 특히 효과적입니다. 추가 토픽 호스트를 정의하여 라우트 가용성을 개선하고 발행 워크로드를 수평적으로 확장할 수 있습니다. 72 페이지의 『발행/구독 클러스터의 토픽 호스트 라우팅』의 내용을 참조하십시오.

계층은 수동 구성 설정이 가장 요구되며 수정하기에 가장 어려운 토폴로지입니다. 계층의 각 큐 관리자와 직접 관계 사이의 관계를 수동으로 구성해야 합니다. 관계를 구성하고 나면 계층 내 다른 큐 관리자의 구독으로 발행물(이전 두 토폴로지에 대한)이 라우팅됩니다. 계층 관계를 사용하여 발행물이 라우팅됩니다. 이를 통해 여러 다른 요구사항을 충족시킬 매우 구체적 토폴로지를 구성할 수 있지만 발행물이 구독에 도달하기까지 중간 큐 관리자의 많은 "홉(Hop)"이 초래될 수 있습니다. 발행에 대한 계층의 라우트가 항상 하나 뿐이므로 모든 큐 관리자의 가용성이 중요합니다. 계층 구조는 일반적으로 단일 클러스터를 구성할 수 없는 경우(예를 들어, 여러 조직에 걸쳐 있을 때)에만 선호됩니다. 93 페이지의 『발행/구독 계층의 라우팅』의 내용을 참조하십시오.

필요에 따라 위의 세 가지 토폴로지를 결합하여 특정 토폴로지 요구사항을 해결할 수 있습니다. 예를 보려면 다중 클러스터의 토픽 공간 결합을 참조하십시오.

분산 발행/구독 네트워크에 적합한 토폴로지를 선택하려면 다음의 광범위한 질문을 고려해야 합니다.

- 네트워크는 얼마나 큽니까?
- 구성에 대한 수동 제어가 어느 정도 필요합니까?
- 토픽 및 구독 관점에서 그리고 큐 관리자 관점에서 시스템은 얼마나 동적입니까?
- 가용성 및 확장성 요구사항은 무엇입니까?
- 모든 큐 관리자를 서로 직접 연결할 수 있습니까?

프로시저

- 필요한 네트워크의 크기를 추정하십시오.
  a) 필요한 토픽 수를 추정하십시오.
  b) 예상하는 발행자와 구독자 수를 추정하십시오.
  c) 발행/구독 활동에 관련될 큐 관리자 수를 추정하십시오.

  81 페이지의 『발행/구독 클러스터링: 우수 사례』에서 특히 다음 섹션도 참조하십시오.

  - 시스템 크기 조절 방법
  - 발행/구독 활동에 관련된 클러스터 큐 관리자의 수를 제한하는 이유
  - 클러스터링할 토픽의 결정 방법

  네트워크에 많은 큐 관리자가 있고 네트워크에서 많은 발행자와 구독자를 핸들링할 경우 토픽 호스트 라우트 클러스터 또는 계층를 사용해야 합니다. 직접 라우트 클러스터는 수동 구성이 거의 필요하지 않으며 작거나 정적 네트워크에 좋은 솔루션일 수 있습니다.
- 각 토픽, 발행자 또는 구독자를 호스트하는 큐 관리자에 대한 수동 제어가 어느 정도 필요한지 고려하십시오.
  a) 일부 큐 관리자가 다른 큐 관리자보다 덜 기능적인지 고려하십시오.
  b) 일부 큐 관리자에 대한 통신 링크가 다른 링크에 비해 더 취약한지 고려하십시오.
  c) 토픽에 많은 발행물이 있고 구독자는 거의 없을 것이라 예상하는 경우를 식별하십시오.
  d) 토픽에 많은 구독자가 있고 발행자는 거의 없을 것이라 예상하는 경우를 식별하십시오.

  모든 토폴로지에서 발행물은 다른 큐 관리자의 구독에 전달됩니다. 직접 라우트 클러스터에서는 발행물을 구독에 전달하는 경로가 가장 짧습니다. 토픽 호스트 라우트 클러스터 또는 계층에서는 발행물의 라우트를 사용

자가 제어합니다. 큐 관리자의 기능이 각기 다르거나 가용성 및 연결성의 레벨이 다양한 경우 특정 워크로드를 특정 큐 관리자에 지정할 수 있습니다. 토픽 호스트 라우트 클러스터 또는 계층를 사용하여 이를 수행할 수 있습니다.

모든 토폴로지에서, 최소 오버헤드와 최대 성능이 가능할 때마다 구독과 동일한 큐 관리자에 발행 애플리케이션을 함께 두십시오. 토픽 호스트 라우트 클러스터의 경우 토픽을 호스트하는 큐 관리자에 발행자나 구독자를 둘 것을 고려하십시오. 그러면 구독자에 발행물을 전달할 큐 관리자 간에 여분의 "홉(Hop)"이 제거됩니다. 이 접근법은 한 토픽에 많은 발행자가 있고 구독자는 거의 없거나 많은 구독자가 있고 발행자는 거의 없는 경우에 특히 효과적입니다. 예를 보려면 집중식 발행자 또는 구독자를 사용한 토픽 호스트 라우팅을 참조하십시오.

81 페이지의 『발행/구독 클러스터링: 우수 사례』 에서 특히 다음 섹션도 참조하십시오.

- 클러스터링할 토픽의 결정 방법
- 발행자 및 구독 위치

- 네트워크 활동이 얼마나 동적인지 고려하십시오.

  a) 구독자가 다른 토픽에 추가 및 제거될 빈도를 추정하십시오.

  큐 관리자로부터 구독이 추가 또는 제거되고 특정 토픽 문자열의 첫 번째 또는 마지막 구독이 될 때마다 이 정보는 토폴로지의 다른 큐 관리자에게 통신됩니다. 직접 라우트 클러스터 및 계층에서는, 토픽에 발행자가 있는지 여부와 상관 없이 이 구독 정보가 토폴로지의 모든 큐 관리자에 전파됩니다. 토폴로지가 많은 큐 관리자로 이루어진 경우 이는 상당한 성능 오버헤드일 수 있습니다. 토픽 호스트 라우트 클러스터에서는, 구독의 토픽 문자열로 맵핑된 클러스터된 토픽을 호스트하는 큐 관리자에만 이 정보가 전파됩니다.

  81 페이지의 『발행/구독 클러스터링: 우수 사례』의 구독 변경 및 동적 토픽 문자열 절도 참조하십시오.

  참고: 많은 고유 토픽 문자열 세트가 항상 빠르게 변경되는 매우 동적인 시스템에서는 "모든 위치에서 발행" 모드로 모델을 전환하는 것이 최상일 수 있습니다. 발행/구독 네트워크에서의 구독 성능을 참조하십시오.

  b) 토폴로지의 큐 관리자가 얼마나 동적인지 고려하십시오.

  계층는 토폴로지의 큐 관리자 변경을 계층에서 수동으로 삽입하거나 제거해야 하며, 계층의 상위 레벨에서 큐 관리자를 변경할 때에는 주의해야 합니다. 또한 계층의 큐 관리자는 보통 수동으로 구성된 채널 연결을 사용합니다. 계층에서 큐 관리자가 추가 및 제거될 때 채널을 추가하고 제거하면서 이 연결을 유지보수해야 합니다.

  발행/구독 클러스터에서, 큐 관리자는 처음으로 클러스터를 결합할 때 필요한 다른 큐 관리자에 자동으로 연결되고 토픽 및 구독을 자동으로 인식하게 됩니다.

- 라우트 가용성 및 발행 트래픽 확장성 요구사항을 고려하십시오.

  a) 큐 관리자가 사용 불가능할 때에도 발행 큐 관리자로부터 구독 큐 관리자까지의 사용 가능한 루트가 항상 필요한지 여부를 결정하십시오.

  b) 네트워크의 필요한 확장성을 고려하십시오. 발행 트래픽 레벨이 단일 큐 관리자 또는 채널로 라우트하기에 너무 높은지 여부 및 발행 트래픽 레벨을 단일 토픽 분기에서 핸들링해야 하거나 여러 토픽 분기에 분산시킬 수 있는지 여부를 결정하십시오.

  c) 메시지 순서화를 유지보수할 필요가 있는지 고려하십시오.

  직접 라우트 클러스터는 발행 큐 관리자에 직접 메시지를 보내기 때문에 라우트에 따른 중간 큐 관리자의 가용성을 고려할 필요가 없습니다. 이와 마찬가지로, 중간 큐 관리자로의 확장도 고려 대상이 아닙니다. 하지만 이전에 언급한 대로, 클러스터의 모든 큐 관리자 간에 자동으로 유지보수되는 채널 및 정보 플로우의 오버헤드는 특히 크거나 동적인 환경에서 성능에 상당한 영향을 미칠 수 있습니다.

  토픽 호스트 라우트 클러스터는 개별 토픽에 대해 조정할 수 있습니다. 발행 워크로드가 상당한 토픽 트리의 각 분기가 각기 다른 큐 관리자에 정의되는지 및 각 큐 관리자가 토픽 트리의 해당 분기에 대한 예상 워크로드에 충분히 사용 가능하고 성능이 우수한지 확인할 수 있습니다. 다수의 큐 관리자에 각 토픽을 정의하여 가용성 및 수평적 확장을 추가로 개선할 수도 있습니다. 그러면 시스템이 사용 불가능한 토픽 호스트 큐 관리자 주위를 라우트해서 큐 관리자에 발행 트래픽의 워크로드 밸런스를 조절할 수 있습니다. 하지만 여러 큐 관리자에 주어진 토픽을 정의할 때에는 다음 제한조건도 따릅니다.

  - 발행물에 대한 메시지 순서화가 유실됩니다.

- 보유된 발행물을 사용할 수 없습니다. 91 페이지의 『발행/구독 클러스터에서 보유된 발행에 대한 디자인 고려사항』의 내용을 참조하십시오.

  다중 라우트를 통한 계층에서는 라우팅의 확장성 또는 고가용성을 구성할 수 없습니다.

  81 페이지의 『발행/구독 클러스터링: 우수 사례』의 발행 트래픽절도 참조하십시오.
- 이러한 계산을 기반으로, 제공된 링크를 사용하여 토픽 호스트 라우트 클러스터, 직접 라우트 클러스터, 계층 또는 이 토폴로지의 혼합을 사용할지 결정하십시오.

다음에 수행할 작업

이제 분산 발행/구독 네트워크를 구성할 준비가 되었습니다.

관련 태스크

큐 관리자 클러스터 구성

분산 큐잉 구성

발행/구독 클러스터 구성

큐 관리자를 발행/구독 계층에 연결

발행/구독 클러스터 디자인

두 가지 기본 발행/구독 클러스터 토폴로지인 직접 라우팅 및 토픽 호스트 라우팅이 있습니다. 각각의 혜택이 다릅니다. 발행/구독 클러스터를 설계할 때에는 예상하는 네트워크 요구사항에 가장 잘 맞는 토폴로지를 선택하십시오.

두 개의 발행/구독 클러스터 토폴로지에 대한 개요는 발행/구독 클러스터를 참조하십시오. 네트워크 요구사항을 평가하는 데 도움을 받으려면 63 페이지의 『분산 발행/구독 네트워크 계획』 및 81 페이지의 『발행/구독 클러스터링: 우수 사례』의 내용을 참조하십시오.

일반적으로 두 클러스터 토폴로지 모두 다음 이점을 제공합니다.

- 포인트-투-포인트 클러스터 토폴로지 최상위의 단순 구성.
- 클러스터에 조인하고 떠나는 큐 관리자의 자동 핸들링.
- 여분의 큐 관리자 추가 및 이들 사이에 추가 구독과 발행자 분배를 통한 추가 구독 및 발행자에 대한 간편한 스케일링.

하지만 두 토폴로지는 요구사항이 보다 구체적이어서 혜택의 차이가 있습니다.

직접 라우트 발행/구독 클러스터

직접 라우팅을 사용하면 클러스터의 모든 큐 관리자가 연결된 애플리케이션으로부터 클러스터의 일치하는 구독이 있는 다른 큐 관리자에게 직접 발행물을 보냅니다.

직접 라우트 발행/구독 클러스터는 다음과 같은 이점을 제공합니다.

- 동일한 클러스터의 특정 큐 관리자에 대한 구독에 예정된 메시지는 해당 큐 관리자에게 직접 전송되므로 중간 큐 관리자를 통과할 필요가 없습니다. 이러한 점은 토픽 호스트 라우트 토폴로지 또는 계층 구조 토폴로지와 비교하여 성능을 개선할 수 있습니다.
- 모든 큐 관리자가 서로 직접 연결되어 있기 때문에 이 토폴로지의 라우팅 인프라에는 단일 장애점이 없습니다. 한 큐 관리자가 사용 불가능한 경우 클러스터의 다른 큐 관리자에 대한 구독이 사용 가능한 큐 관리자에서 여전히 메시지를 수신할 수 있습니다.
- 이는 특히 기존 클러스터에서 매우 간단하게 구성할 수 있습니다.

직접 라우트 발행/구독 클러스터 사용 시에 고려할 사항:

- 클러스터의 모든 큐 관리자가 클러스터의 다른 모든 큐 관리자를 인지합니다.
- 클러스터에서 클러스터된 토픽에 대한 하나 이상의 구독을 호스트하는 큐 관리자는 클러스터의 다른 모든 큐 관리자가 클러스터된 토픽에 대한 메시지를 발행하고 있지 않아도 이 큐 관리자에 대한 클러스터 송신자 채널을 자동으로 작성합니다.
- 클러스터된 토픽 하의 토픽 문자열에 대한 큐 관리자의 첫 번째 구독으로 인해 클러스터의 다른 모든 큐 관리자에게 메시지가 송신됩니다. 마찬가지로, 삭제할 토픽 문자열에 대한 마지막 구독도 메시지를 발생시킵니다. 클

러스터된 토픽 하에 사용 중인 개별 토픽 문자열이 많을수록 구독의 변경률이 더 높아지고 큐 관리자 간 통신이 더 많이 발생합니다.

- 큐 관리자가 해당 토픽을 발행 및 구독하지 않을 때에도 클러스터의 모든 큐 관리자가 알림을 받는 구독 토픽 문자열 정보를 유지보수합니다.

위와 같은 이유로, 직접 라우트 토픽이 정의된 클러스터의 모든 큐 관리자는 추가 오버헤드를 유발합니다. 클러스터에 큐 관리자가 더 많을수록 오버헤드가 더 커집니다. 이와 마찬가지로, 더 많은 토픽 문자열을 구독하고 구독 변경률이 더 클수록 오버헤드가 더 커집니다. 이로 인해 동적 또는 큰 직접 라우트 발행/구독 클러스터의 작은 시스템에서 실행하는 큐 관리자에 너무 많은 로드가 초래될 수 있습니다. 추가 정보는 직접 라우트 발행/구독 성능을 참조하십시오.

클러스터가 직접 라우트 클러스터되는 발행/구독의 오버헤드를 수용할 수 없음이 파악되면 대신에 토픽 호스트 라우트 발행/구독을 사용할 수 있습니다. 또는 극단적인 상황에서, 클러스터의 모든 큐 관리자에서 큐 관리자 속성 **PSCLUS**를 DISABLED로 설정하여 클러스터된 발행/구독 기능을 완전히 사용 불가능하게 할 수 있습니다. 90 페이지의 『클러스터된 발행/구독 금지』의 내용을 참조하십시오. 그러면 클러스터된 토픽이 작성되지 않아서 클러스터된 발행/구독과 연관된 오버헤드가 네트워크에 초래되지 않습니다.

토픽 호스트 라우트 발행/구독 클러스터

토픽 호스트 라우팅의 경우 클러스터된 토픽이 관리적으로 정의된 큐 관리자가 발행물에 대한 라우터가 됩니다. 클러스터에서 비호스팅 큐 관리자의 발행물은 호스팅 큐 관리자를 통해 일치하는 구독이 있는 클러스터의 큐 관리자로 라우팅됩니다.

토픽 호스트 라우트 발행/구독 클러스터는 직접 라우트 발행/구독 클러스터와 비교하여 다음과 같은 추가 혜택을 제공합니다.

- 토픽 호스트 라우트 토픽이 정의된 큐 관리자만이 클러스터의 다른 모든 큐 관리자를 인식합니다.
- 토픽 호스트 큐 관리자만이 클러스터의 다른 모든 큐 관리자에 연결할 수 있어야 하며 일반적으로 구독이 있는 큐 관리자에만 연결합니다. 따라서 큐 관리자 간에 실행 채널이 상당히 적습니다.
- 클러스터된 토픽에 대한 하나 이상의 구독을 호스트하는 클러스터 큐 관리자는 구독의 토픽 문자열로 맵핑할 클러스터 토픽을 호스트하는 큐 관리자에만 자동으로 클러스터 송신자 채널을 작성합니다.
- 클러스터된 토픽 하의 토픽 문자열에 대한 큐 관리자의 첫 번째 구독으로 인해 클러스터에서 클러스터된 토픽을 호스트하는 큐 관리자에게 메시지가 송신됩니다. 마찬가지로, 삭제할 토픽 문자열에 대한 마지막 구독도 메시지를 발생시킵니다. 클러스터된 토픽 하에 사용 중인 개별 토픽 문자열이 많을수록 구독의 변경률이 더 높아지고 큐 관리자 간 통신이 더 많이 발생하지만 이는 구독 호스트와 토픽 호스트 사이에서만입니다.
- 실제 구성에 대한 추가 제어. 직접 라우팅에서는 모든 큐 관리자가 발행/구독 클러스터에 참여해야 하므로 오버헤드가 증가합니다. 토픽 호스트 라우팅에서는 토픽 호스트 큐 관리자만이 다른 큐 관리자와 해당 구독을 인식합니다. 따라서 토픽 호스트 큐 관리자를 명시적으로 선택한 후 이 큐 관리자가 적절한 장치에서 실행 중인지 확인하고 다른 큐 관리자에 대해서는 기능이 적은 시스템을 사용할 수 있습니다.

토픽 호스트 라우트 발행/구독 클러스터 사용 시에 고려할 사항:

- 발행자나 구독자가 토픽 호스트 큐 관리자에 있지 않을 경우 발행 큐 관리자와 구독 큐 관리자 간에 추가 "홉"이 발생합니다. 추가 "홉"으로 인한 대기 시간은 토픽 호스트 라우팅이 직접 라우팅보다 덜 효율적이라는 의미일 수 있습니다.
- 대형 클러스터에는 토픽 호스트 라우팅이 직접 라우팅의 경우에 발생 가능한 상당한 성능 및 스케일링 문제를 완화시킵니다.
- 모든 토픽을 단일 큐 관리자에 또는 매우 작은 수의 큐 관리자에 정의하도록 선택할 수 있습니다. 선택 시 토픽 호스트 큐 관리자가 연결성이 우수한 강력한 시스템에서 호스트되는지 확인하십시오.
- 둘 이상의 큐 관리자에 동일한 토픽을 정의할 수 있습니다. 이 방식은 토픽의 가용성을 개선하며 IBM MQ가 토픽의 모든 호스트에서 토픽에 대한 발행물의 워크로드 밸런스를 조절하므로 확장성도 개선합니다. 하지만 둘 이상의 큐 관리자에 동일한 토픽을 정의하면 해당 토픽에 대한 메시지 순서가 손실됩니다.
- 상이한 큐 관리자에서 여러 다른 토픽을 호스트하여 메시지 순서를 유지한 채로 확장성을 개선할 수 있습니다.

관련 태스크

시나리오: 발행/구독 클러스터 작성

발행/구독 클러스터 구성

분산 발행/구독 네트워크 성능 조정
분산 발행/구독 문제점 해결

발행/구독 클러스터의 직접 라우팅

발행 큐 관리자의 발행물은 클러스터의 일치하는 구독이 있는 다른 큐 관리자로 직접 라우팅됩니다.

발행/구독 계층의 큐 관리자와 클러스터 간에 메시지가 라우팅되는 방식에 대한 소개는 분산 발행/구독 네트워크를 참조하십시오.

직접 라우트 발행/구독 클러스터는 다음과 같이 작동합니다.

- 모든 큐 관리자는 자동으로 다른 모든 큐 관리자를 알고 있습니다.
- 클러스터된 토픽에 대한 구독이 있는 모든 큐 관리자는 클러스터의 다른 모든 큐 관리자에 대한 채널을 작성하고 이 큐 관리자에 구독을 알립니다.
- 애플리케이션이 발행한 메시지는 연결된 큐 관리자에서 일치하는 구독이 있는 각 큐 관리자로 직접 라우팅됩니다.

다음 다이어그램은 발행/구독 또는 포인트-투-포인트 활동에 현재 사용되지 않는 큐 관리자 클러스터를 보여줍니다. 클러스터에 있는 모든 큐 관리자는 전체 저장소 큐 관리자로만 연결됨을 유의하십시오.

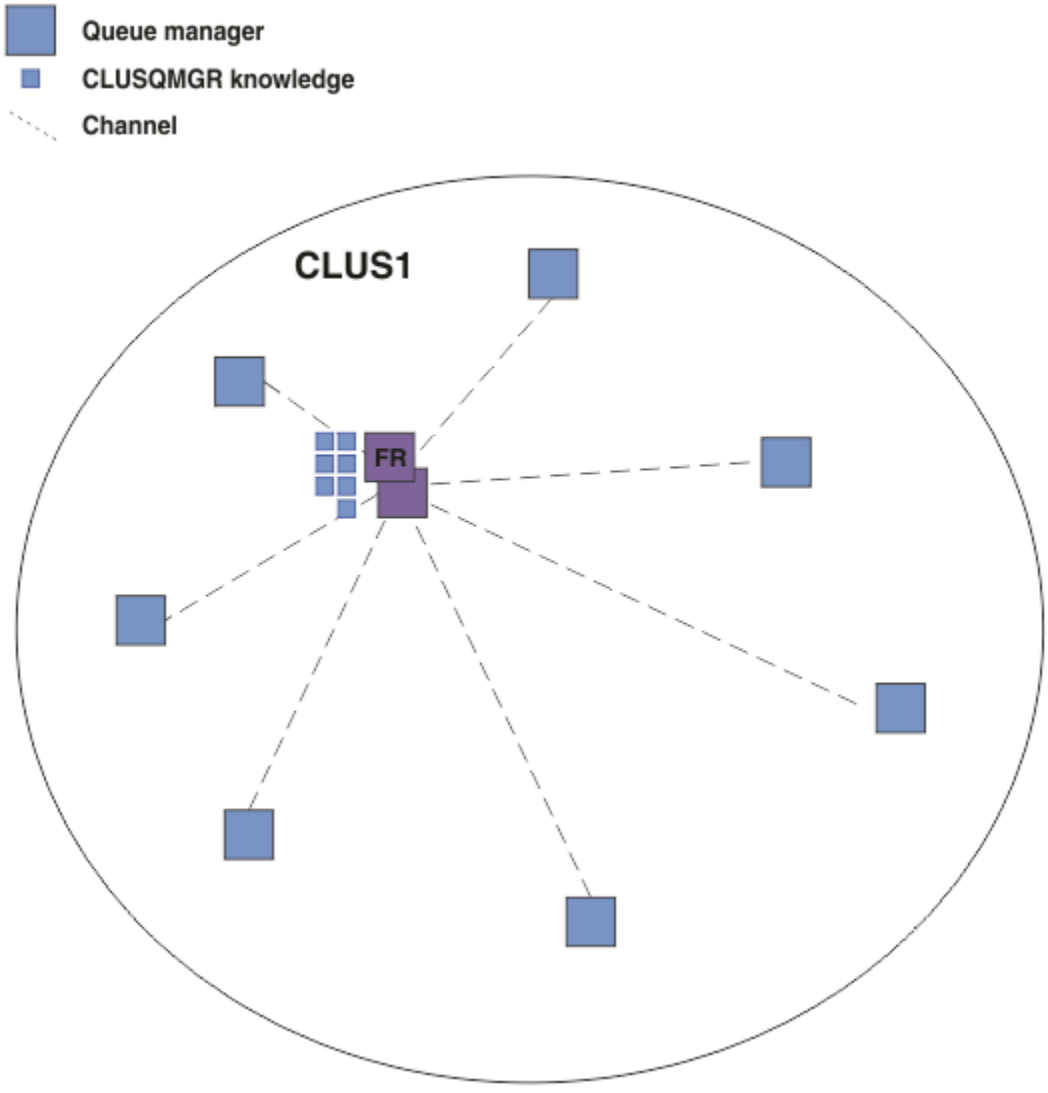


그림 *16.* 큐 관리자 클러스터

직접 라우트 클러스터의 큐 관리자 간에 플로우되는 발행물의 경우 발행/구독 클러스터 구성에 설명된 대로 토픽 트리의 분기를 클러스터링하고 직접 라우팅(기본값)을 지정합니다.

직접 라우트 발행/구독 클러스터에서는 클러스터의 큐 관리자에 대한 토픽 오브젝트를 정의합니다. 이를 수행할 때 오브젝트 정보 및 클러스터의 다른 모든 큐 관리자 정보가 전체 저장소 큐 관리자를 통해 클러스터의 모든 큐 관리자에 자동으로 푸시됩니다. 큐 관리자가 토픽을 참조하기 전에 이 푸시가 발생합니다.

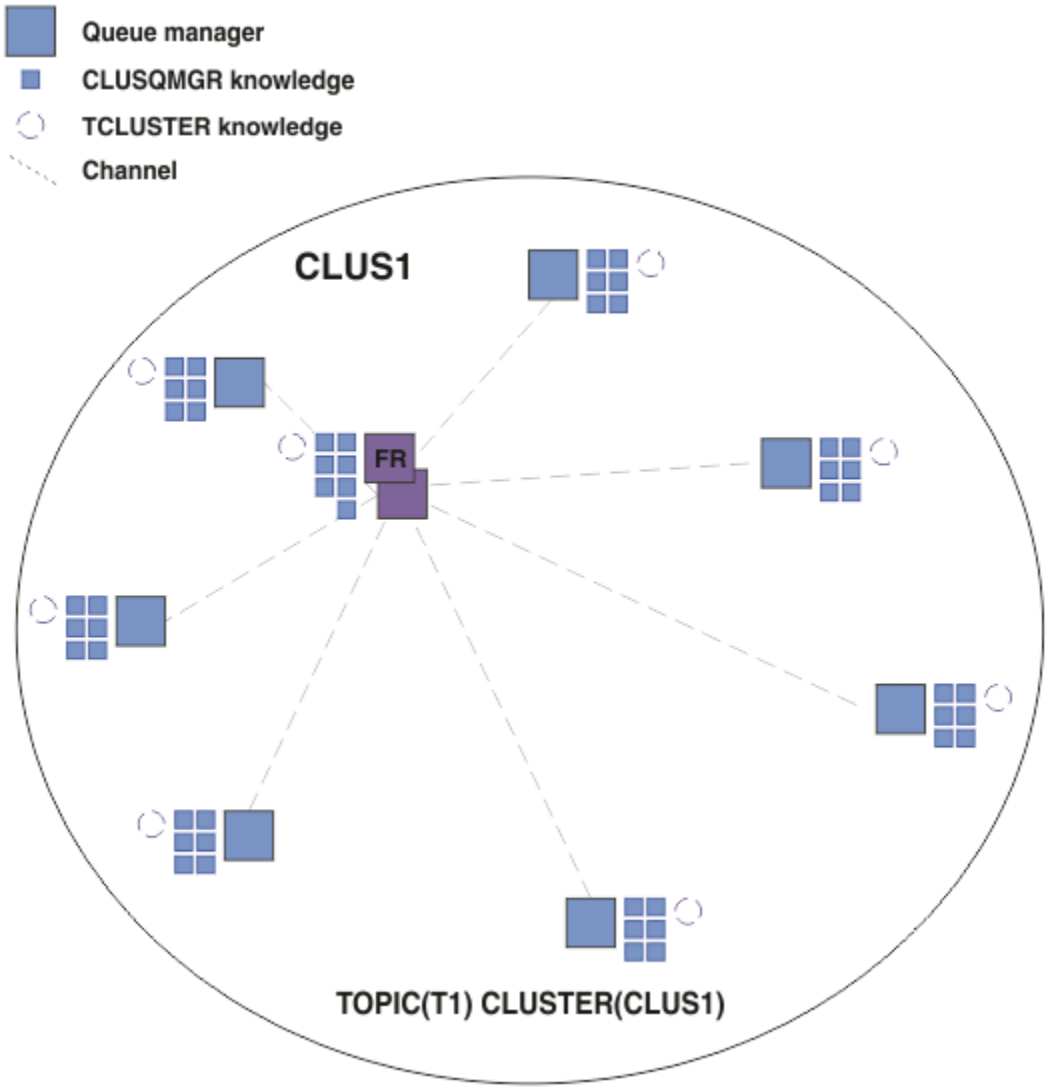


그림 *17.* 직접 라우트 발행/구독 클러스터

구독이 작성될 때 구독을 호스트하는 큐 관리자는 클러스터의 모든 큐 관리자에 대한 채널을 설정하고 구독 세부사항을 보냅니다. 이 분산 구독 지식은 각 큐 관리자에 프록시 구독으로 표시됩니다. 해당 프록시 구독의 토픽 문자열과 일치하는 클러스터의 큐 관리자에서 발행물이 생성되면 발행자 큐 관리자에서 구독을 호스트하는 각 큐 관리자로 클러스터 채널이 설정되고 각 큐 관리자에게 메시지가 송신됩니다.

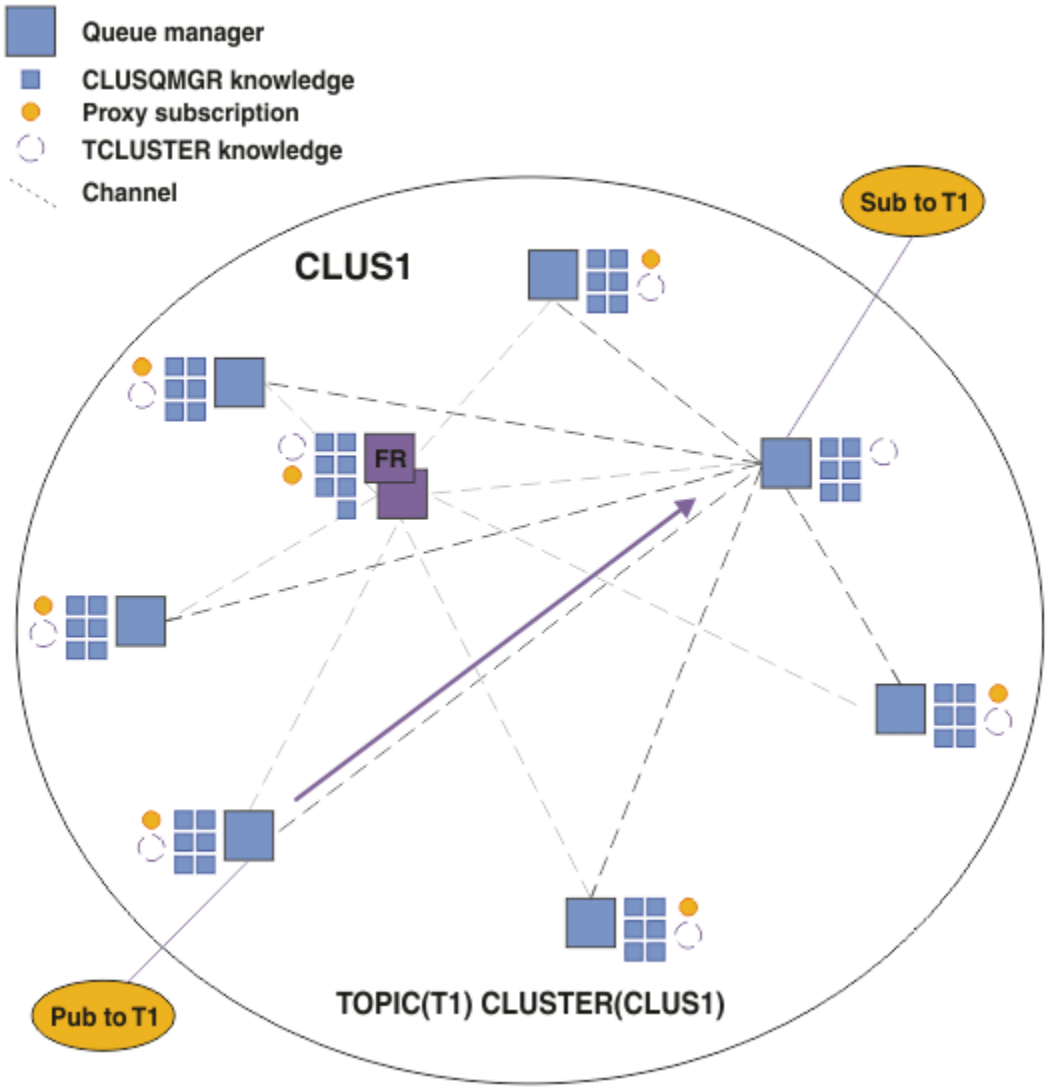


그림 *18.* 클러스터된 토픽에 대한 발행자 및 구독자가 있는 직접 라우트 발행/구독 클러스터

큐 관리자를 호스트하는 구독에 발행물을 직접 라우팅하면 구성이 단순화되고 구독에 발행물을 전달하는 과정의 대기 시간이 최소화됩니다.

하지만 구독 및 발행자의 위치에 따라 클러스터가 다른 모든 큐 관리자에 대한 직접 연결이 있는 모든 큐 관리자와 빠르게 완전히 상호 연결될 수 있습니다. 이는 환경에서 허용 가능 또는 불가능할 수 있습니다. 이와 마찬가지로, 구독 중인 토픽 문자열 세트가 자주 변경되는 경우에는 모든 큐 관리자 간에 정보를 전파하는 오버헤드도 상당해질 수 있습니다. 직접 라우트 발행/구독 클러스터의 모든 큐 관리자는 이러한 오버헤드에 잘 대처할 수 있어야 합니다.

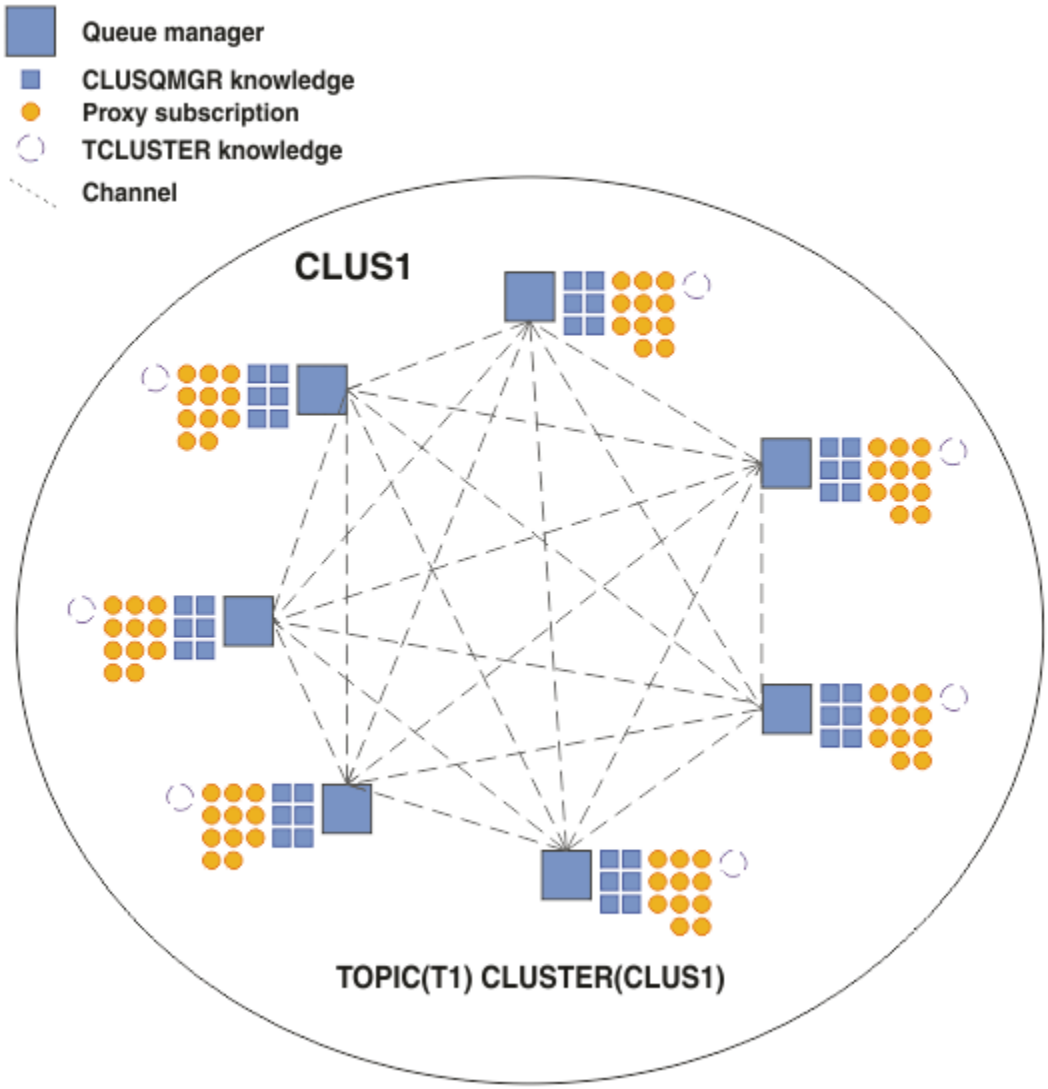


그림 *19.* 완전히 상호 연결된 직접 라우트 발행/구독 클러스터

요약 및 추가 고려사항

직접 라우트 발행/구독 클러스터는 작성 또는 관리를 위한 수동 개입이 거의 필요하지 않으며 발행자와 구독자 간에 직접 라우팅을 제공합니다. 특정 구성의 경우, 특히 큐 관리자가 거의 없는 클러스터나 높은 큐 관리자 연결성이 허용 가능하고 구독이 거의 변경되지 않는 상황에서는 일반적으로 가장 적합한 토폴로지입니다. 하지만 시스템에 대한 일정한 제한조건도 있습니다.

- 각 큐 관리자에 대한 로드는 클러스터의 전체 큐 관리자 수에 비례합니다. 따라서 보다 큰 클러스터에서는 개별 큐 관리자와 시스템이 전체적으로 성능 문제를 경험할 수 있습니다.
- 기본적으로 구독하는 클러스터된 모든 토픽 문자열은 클러스터 전반에 전파되고 발행물은 연관된 토픽에 대한 구독이 있는 리모트 큐 관리자에만 전파됩니다. 따라서 구독 세트의 급격한 변경은 제한 요인이 될 수 있습니다. 이 기본 작동을 변경하고 그 대신 전체 큐 관리자에게 모든 발행물이 전파되도록 할 수 있으며 이 경우 프록시 구독이 필요하지 않습니다. 이는 구독 알림 트래픽을 줄여주지만 발행 트래픽과 각 큐 관리자가 설정하는 채널 수는 늘어날 수 있습니다. 발행/구독 네트워크에서의 구독 성능을 참조하십시오.

  참고: 계층에도 유사한 제한사항이 적용됩니다.
- 발행/구독 큐 관리자의 상호 연결된 속성으로 인해 프록시 구독이 네트워크의 모든 노드에 전파되는 데 시간이 걸립니다. 원격 발행물은 반드시 즉시 구독되는 것은 아니므로 새 토픽 문자열에 대한 구독 후에 이전 발행물이 송신되지 않을 수도 있습니다. 전체 큐 관리자에게 모든 발행물을 전파해서(이 경우 프록시 구독이 필요하지 않음) 구독 지연으로 인한 문제를 피할 수 있습니다. 발행/구독 네트워크에서의 구독 성능을 참조하십시오.

참고: 계층에도 이 제한사항이 적용됩니다.

직접 라우팅을 사용하기 전에 72 페이지의 『발행/구독 클러스터의 토픽 호스트 라우팅』 및 93 페이지의 『발행/구독 계층의 라우팅』에 설명된 대체 접근법을 탐색하십시오.

발행/구독 클러스터의 토픽 호스트 라우팅

클러스터에서 비호스팅 큐 관리자의 발행물은 호스팅 큐 관리자를 통해 일치하는 구독이 있는 클러스터의 큐 관리자로 라우팅됩니다.

발행/구독 계층의 큐 관리자와 클러스터 간에 메시지가 라우팅되는 방식에 대한 소개는 분산 발행/구독 네트워크를 참조하십시오.

토픽 호스트 라우팅의 작동 및 혜택을 이해하려면 먼저 68 페이지의 『발행/구독 클러스터의 직접 라우팅』을 이해하는 것이 가장 좋습니다.

토픽 호스트 라우트 발행/구독 클러스터는 다음과 같이 작동합니다.

- 클러스터되어 관리된 토픽 오브젝트는 클러스터의 개별 큐 관리자에 수동으로 정의됩니다. 이를 토픽 호스트 큐 관리자라 부릅니다.
- 클러스터 큐 관리자에서 구독이 이루어지면 구독 호스트 큐 관리자로부터 토픽 호스트 큐 관리자까지 채널이 작성되고 토픽을 호스트하는 큐 관리자에만 프록시 구독이 작성됩니다.
- 애플리케이션이 토픽에 정보를 발행할 때 연결된 큐 관리자는 항상 토픽을 호스트하는 하나의 큐 관리자에게 발행물을 전달하고, 이 큐 관리자는 클러스터에서 토픽에 대한 일치하는 구독이 있는 모든 큐 관리자에게 이 발행물을 전달합니다.

이 프로세스는 다음 예에 보다 자세히 설명되어 있습니다.

단일 토픽 호스트를 사용한 토픽 호스트 라우팅

토픽 호스트 라우트 클러스터의 큐 관리자 간에 플로우되는 발행물의 경우 발행/구독 클러스터 구성에 설명된 대로 토픽 트리의 분기를 클러스터링하고 토픽 호스트 라우팅을 지정합니다.

클러스터의 여러 큐 관리자에 토픽 호스트 라우트 토픽 오브젝트를 정의하는 많은 이유가 있습니다. 하지만 단순함을 위해 단일 토픽 호스트로 시작합니다.

다음 다이어그램은 발행/구독 또는 포인트-투-포인트 활동에 현재 사용되지 않는 큐 관리자 클러스터를 보여줍니다. 클러스터에 있는 모든 큐 관리자는 전체 저장소 큐 관리자로만 연결됨을 유의하십시오.

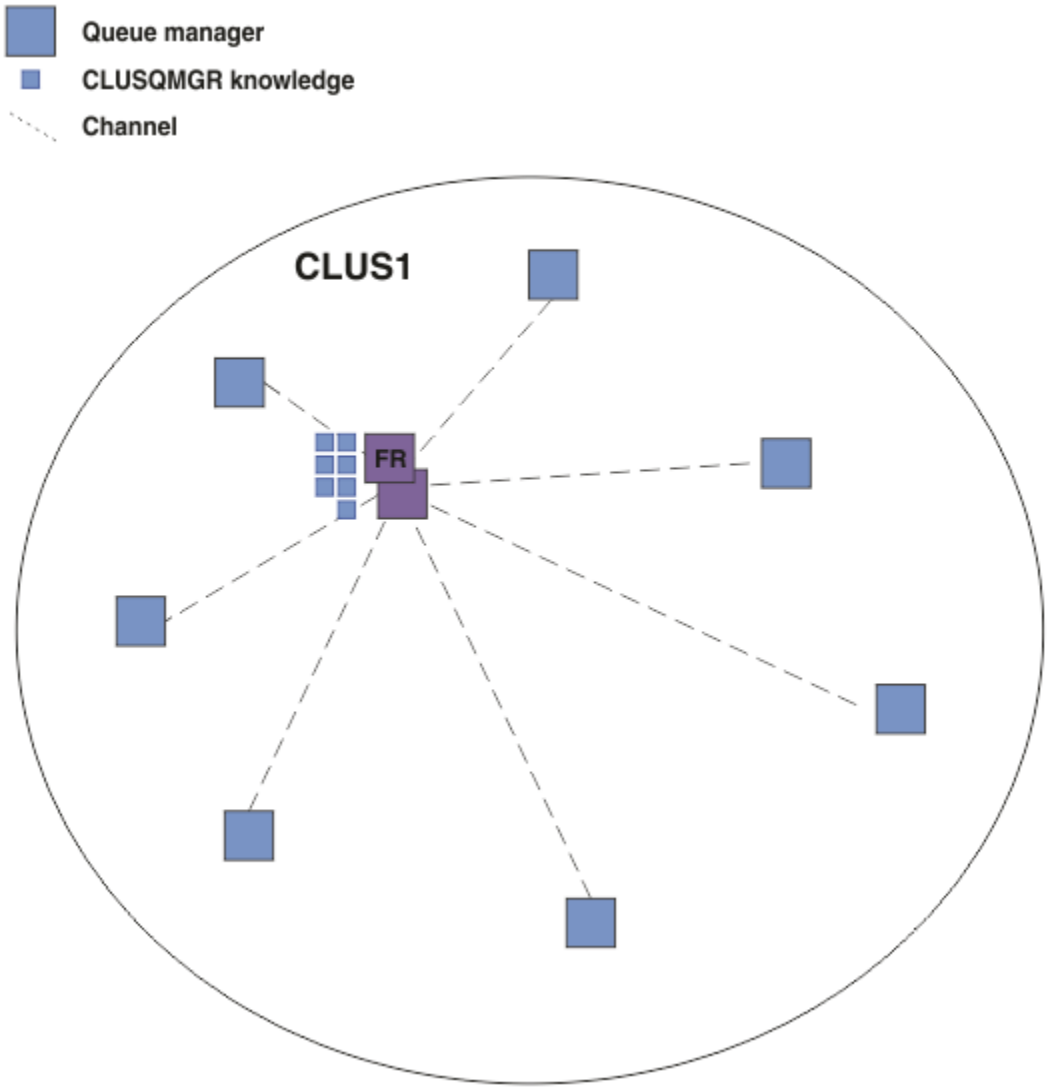


그림 *20.* 큐 관리자 클러스터

토픽 호스트 라우트 발행/구독 클러스터에서는 클러스터의 특정 큐 관리자에 대한 토픽 오브젝트를 정의합니다. 그러면 발행/구독 트래픽이 이 큐 관리자를 통해 플로우되어 이 큐 관리자는 클러스터의 중요 큐 관리자가 되고 워크로드가 증가하게 됩니다. 이러한 이유로 클러스터의 또 다른 큐 관리자를 사용하지 않고 전체 저장소 큐 관리자를 사용하는 방법은 권장하지 않습니다. 호스트 큐 관리자에 토픽 오브젝트를 정의하면 전체 저장소 큐 관리자를 통해 클러스터의 다른 모든 큐 관리자에 오브젝트 및 호스트 정보가 자동으로 푸시됩니다. 직접 라우팅과 달리, 각 큐 관리자에게는 클러스터의 다른 모든 큐 관리자에 대해 알리지 않습니다.

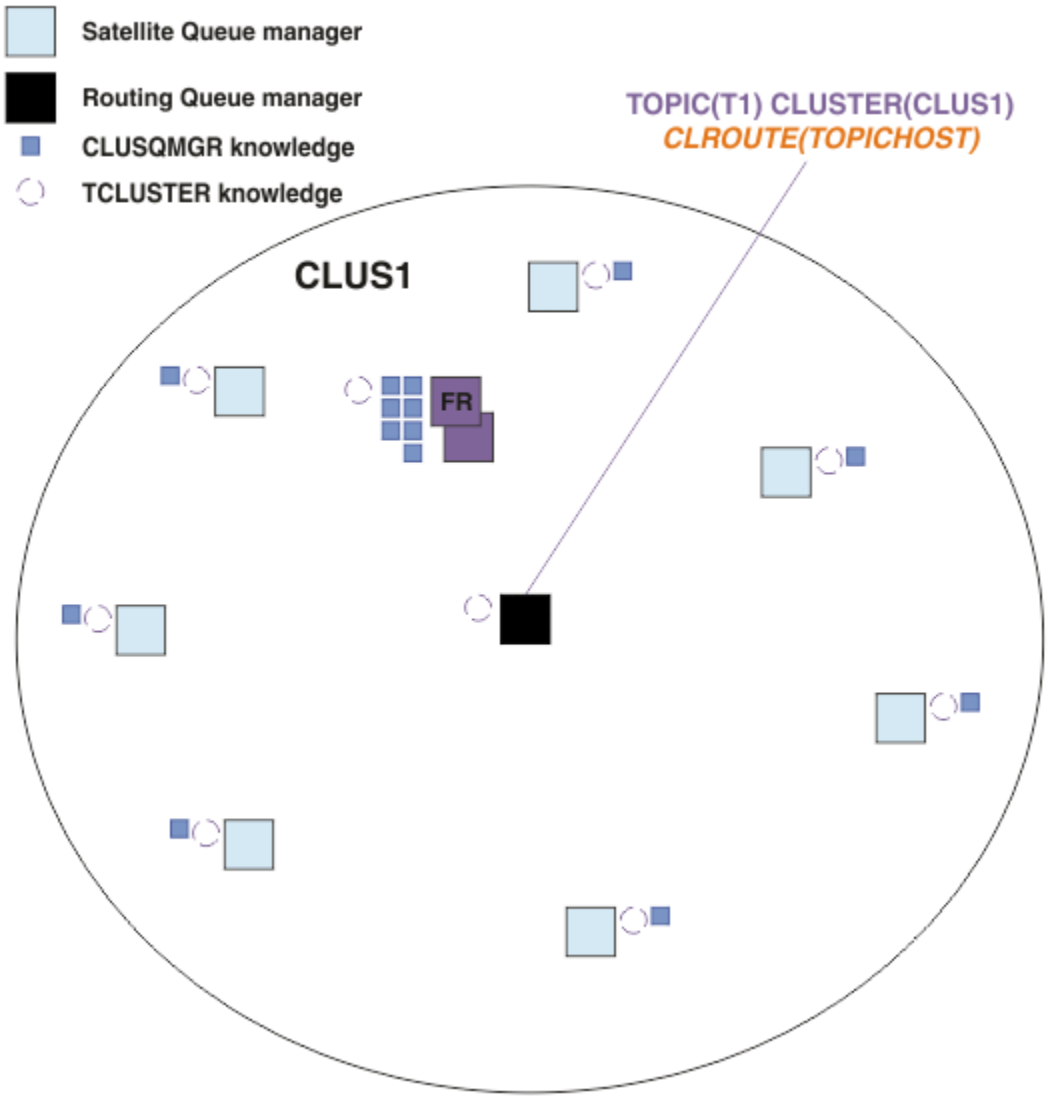


그림 *21.* 하나의 토픽 호스트에 정의된 하나의 토픽이 있는 토픽 호스트 라우트 발행/구독 클러스터

큐 관리자에 구독이 작성될 때 구독하는 큐 관리자와 토픽 호스트 큐 관리자 간에 채널이 작성됩니다. 구독하는 큐 관리자는 토픽 호스트 큐 관리자에만 연결하여 구독 세부사항을 보냅니다(프록시 구독 양식으로). 토픽 호스트 큐 관리자는 클러스터의 추가 큐 관리자에게 이 구독 정보를 전달하지 않습니다.

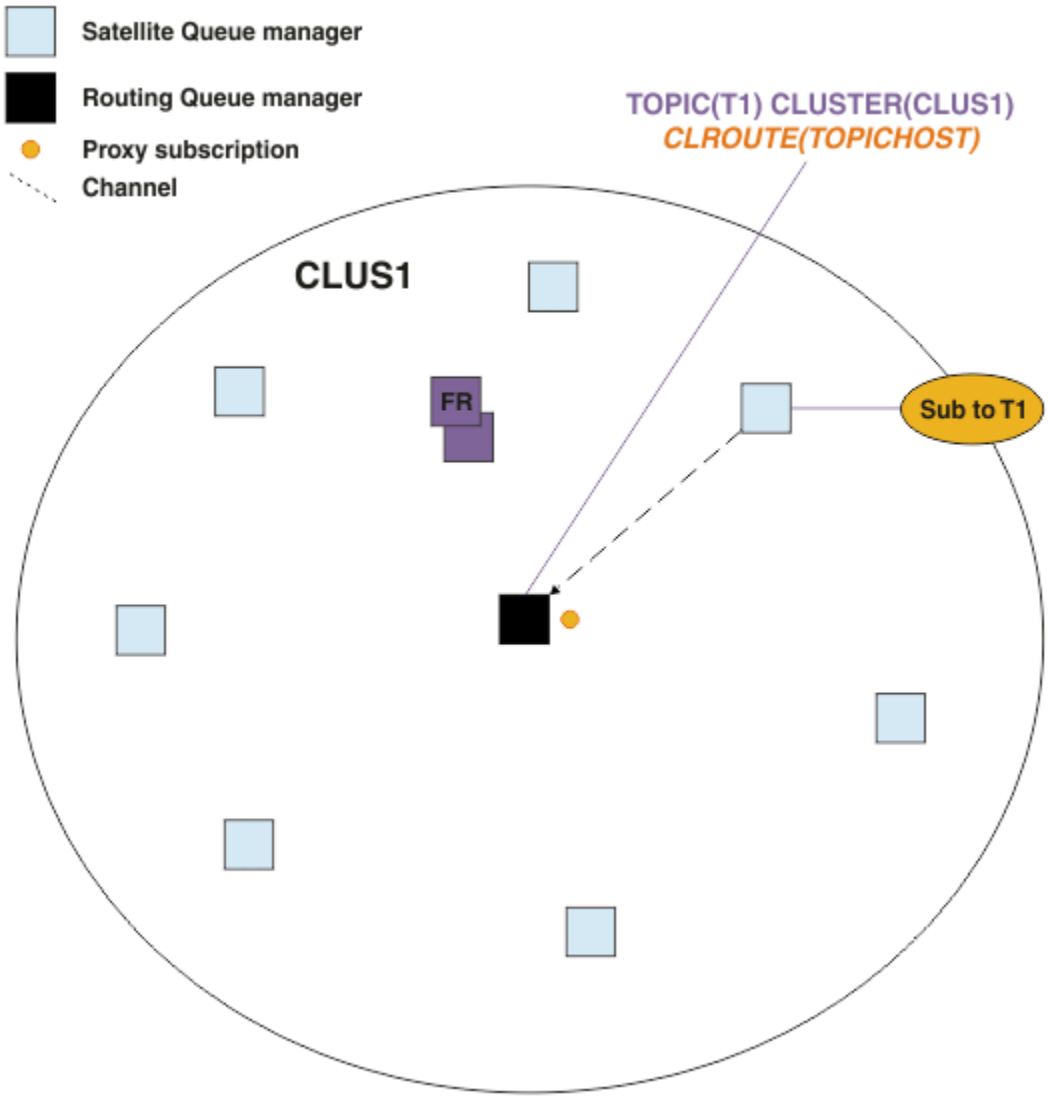


그림 *22.* 하나의 토픽 호스트에 정의된 하나의 토픽 및 하나의 구독자가 있는 토픽 호스트 라우트 발행/구독 클러스터

발행 애플리케이션이 또 다른 큐 관리자에 연결하고 메시지가 발행될 때 발행 큐 관리자와 토픽 호스트 큐 관리자 간에 채널이 작성되고 이 큐 관리자에게 메시지가 전달됩니다. 발행 큐 관리자는 클러스터의 다른 큐 관리자에 대한 구독을 알지 못하므로 클러스터의 이 토픽에 대한 구독이 없어도 토픽 호스트 큐 관리자에게 메시지가 전달됩니다. 발행 큐 관리자는 토픽 호스트 큐 관리자에만 연결합니다. 토픽 호스트를 통해 구독하는 큐 관리자로(존재하는 경우) 발행물이 라우팅됩니다.

발행자와 동일한 큐 관리자에 있는 구독은 먼저 토픽 호스트 큐 관리자에게 메시지를 보내지 않고 직접 충족됩니다.

각 토픽 호스트 큐 관리자가 이행하는 중요 역할로 인해 토픽 호스팅의 로드, 가용성, 연결 요구사항을 핸들링할 수 있는 큐 관리자를 선택해야 합니다.

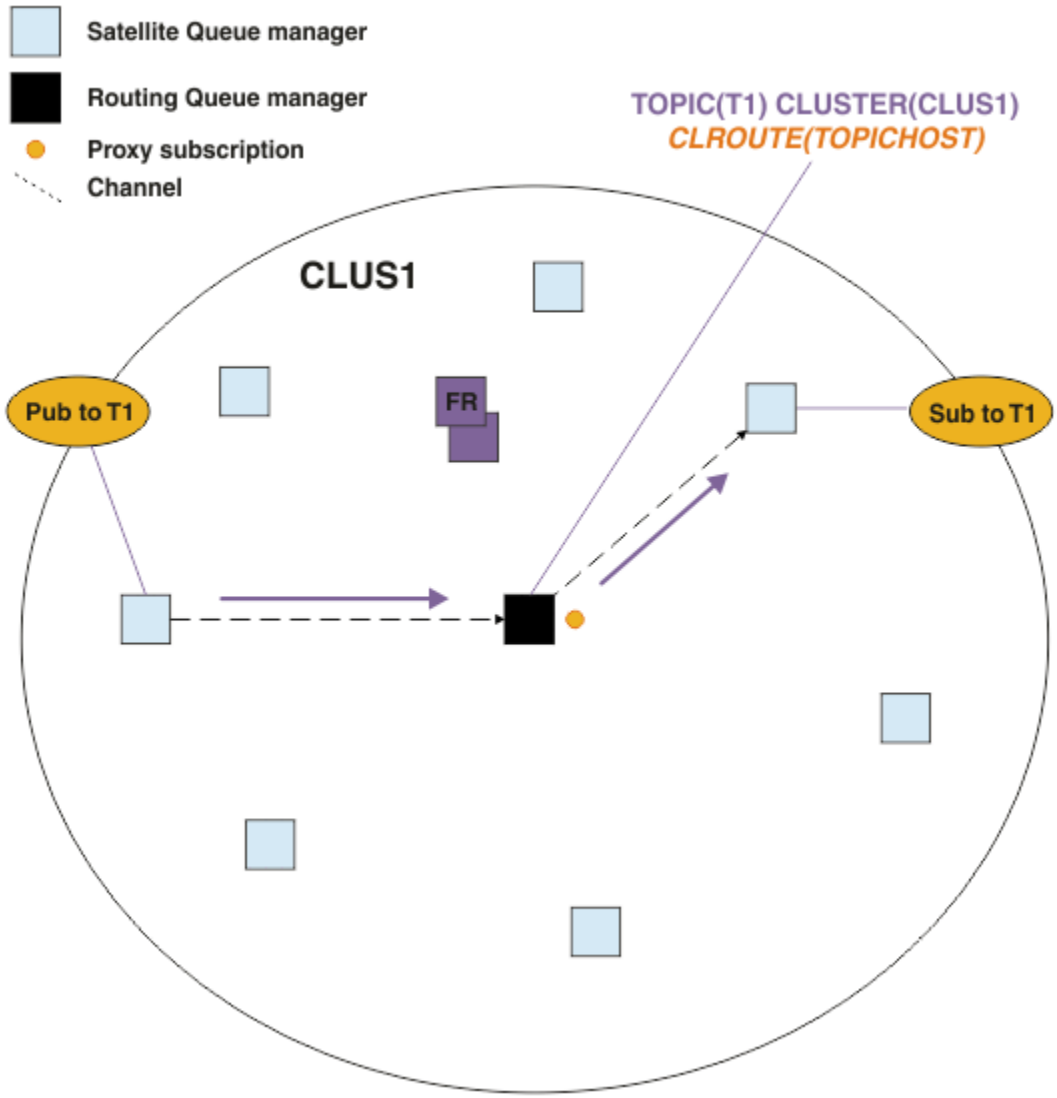


그림 *23.* 한 토픽, 한 구독자, 한 발행자가 있는 토픽 호스트 라우트 발행/구독 클러스터

다중 큐 관리자에 토픽 트리 분할

라우팅된 토픽 호스팅 큐 관리자는 구독 정보 및 관리된 토픽 오브젝트가 구성되는 토픽 트리의 분기에 관련된 발행 메시지에 대해 유일하게 책임이 있습니다. 클러스터의 상이한 발행/구독 애플리케이션에 상이한 토픽이 사용되는 경우 토픽 트리의 상이한 클러스터된 분기를 호스트할 상이한 큐 관리자를 구성할 수 있습니다. 그러면 발행 트래픽, 구독 정보, 클러스터의 각 토픽 호스트 큐 관리자의 채널이 감소하여 스케일링이 허용됩니다. 토픽 트리에서 볼륨이 뚜렷하게 높은 분기에 이 메소드를 사용해야 합니다.

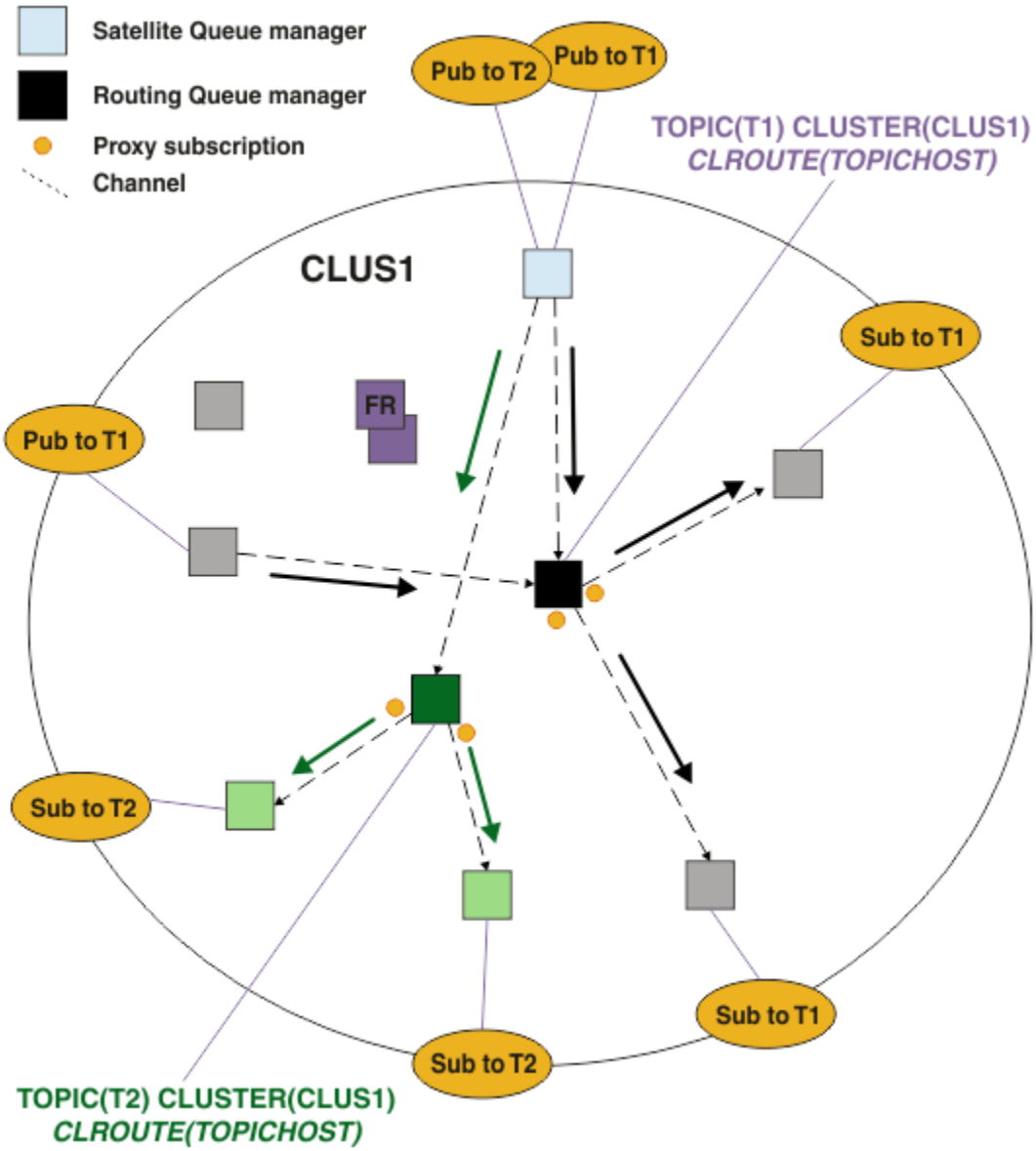


그림 *24.* 하나의 토픽 호스트에 정의된 두 개의 토픽이 있는 토픽 호스트 라우트 발행/구독 클러스터

예를 들어, 토픽 트리에 설명된 토픽을 사용하여 T1 토픽이 토픽 문자열 /USA/Alabama(으)로 구성되고 T2 토픽이 토픽 문자열 /USA/Alaska(으)로 구성된 경우 /USA/Alabama/Mobile(으)로 발행된 메시지는 T1을 호스팅하는 큐 관리자를 통해 라우팅되고 /USA/Alaska/Juneau(으)로 발행된 메시지는 T2를 호스팅하는 큐 관리자를 통해 라우팅됩니다.

참고: 토픽 트리에 클러스터된 지점보다 더 높은 와일드카드를 사용하여 단일 구독 범위가 토픽 트리의 클러스터된 여러 분기에 걸치게 할 수 없습니다. 와일드카드 구독을 참조하십시오.

단일 토픽 호스트에 다중 토픽 호스트를 사용한 토픽 호스트 라우팅

단일 큐 관리자가 토픽 라우팅을 책임지고 이 큐 관리자가 사용 불가능하거나 워크로드를 핸들링할 수 없게 되는 경우 발행물이 구독으로 즉시 플로우되지 않습니다.

하나의 큐 관리자에만 토픽을 정의할 때 탄력성, 확장성, 워크로드 밸런싱이 현재보다 더 필요하면 둘 이상의 큐 관리자에 토픽을 정의할 수 있습니다. 발행된 각 개별 메시지가 단일 토픽 호스트를 통해 라우팅됩니다. 일치하는 토픽 호스트 정의가 여러 개 있으면 토픽 호스트 중 하나가 선택됩니다. 선택은 클러스터된 큐에 대한 방식과 동일하게 이루어집니다. 그러면 사용 불가능한 토픽 호스트를 피해, 사용 가능한 토픽 호스트로 메시지가 라우팅되

고 다수의 토픽 호스트 큐 관리자와 채널 사이에 메시지 로드의 워크로드 밸런싱이 조절됩니다. 하지만 클러스터의 동일한 토픽에 다수의 토픽 호스트를 사용할 때에는 여러 메시지의 순서화가 유지되지 않습니다.

다음 다이어그램은 두 개의 큐 관리자에 동일한 토픽이 정의된 토픽 호스트 라우트 클러스터를 보여줍니다. 이 예에서, 구독하는 큐 관리자는 두 토픽 호스트 큐 관리자 모두에게 구독하는 토픽에 대한 정보를 프록시 구독의 형태로 보냅니다.

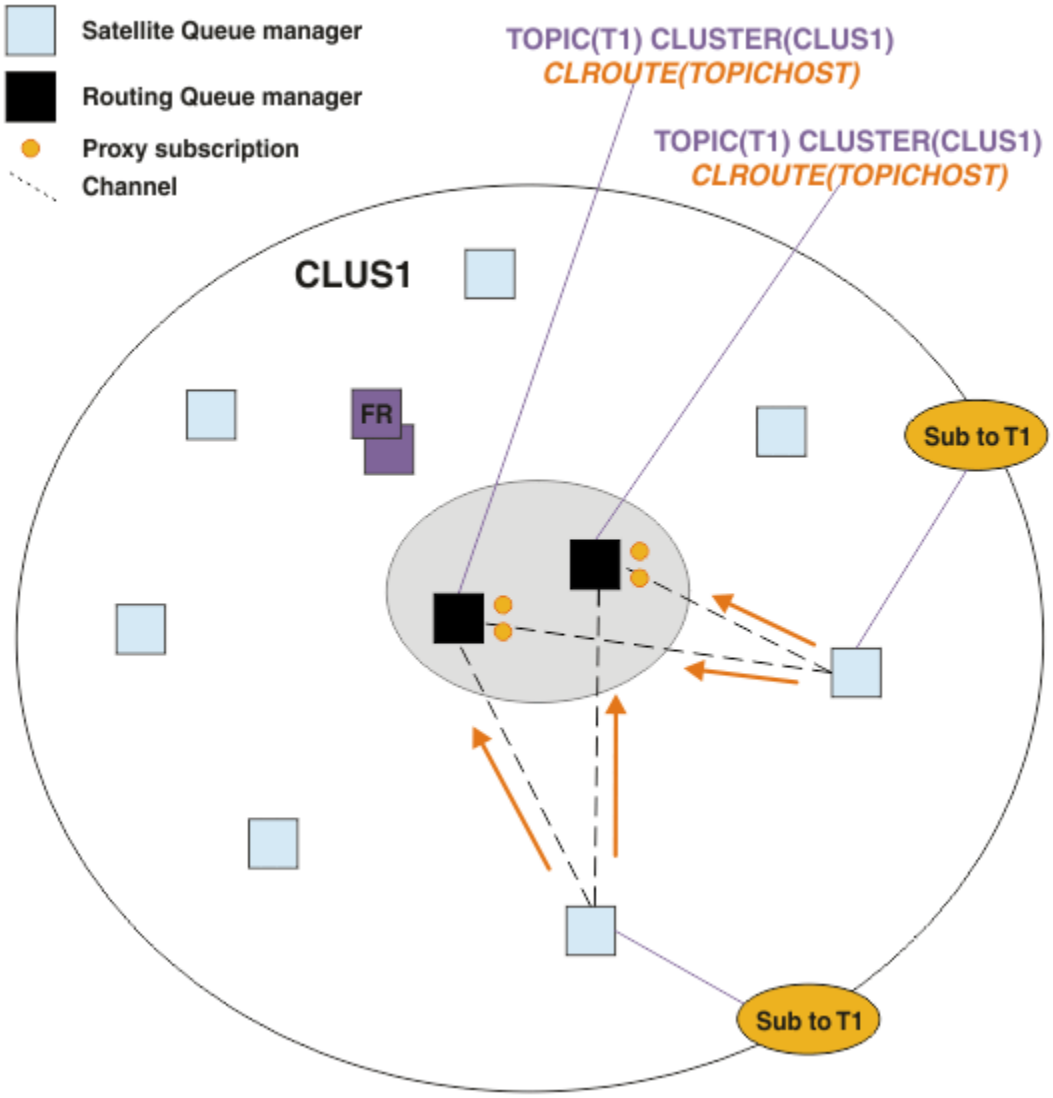


그림 *25.* 다중 토픽 호스트 발행/구독 클러스터에 프록시 구독 작성

호스트하지 않는 큐 관리자로부터 발행물이 작성된 경우 큐 관리자는 토픽 호스트 큐 관리자 중 하나에 해당 토픽에 대한 발행물의 사본을 보냅니다. 시스템은 클러스터 워크로드 관리 알고리즘의 기본 작동에 기초하여 호스트를 선택합니다. 일반 시스템에서 이는 각 토픽 호스트 큐 관리자 전반의 라운드로빈 분배에 근접합니다. 동일한 발행 애플리케이션의 메시지 간에 연관관계가 없어서 이는 `NOTFIXED` 클러스터 바인드 유형을 사용하는 경우와 동일합니다.

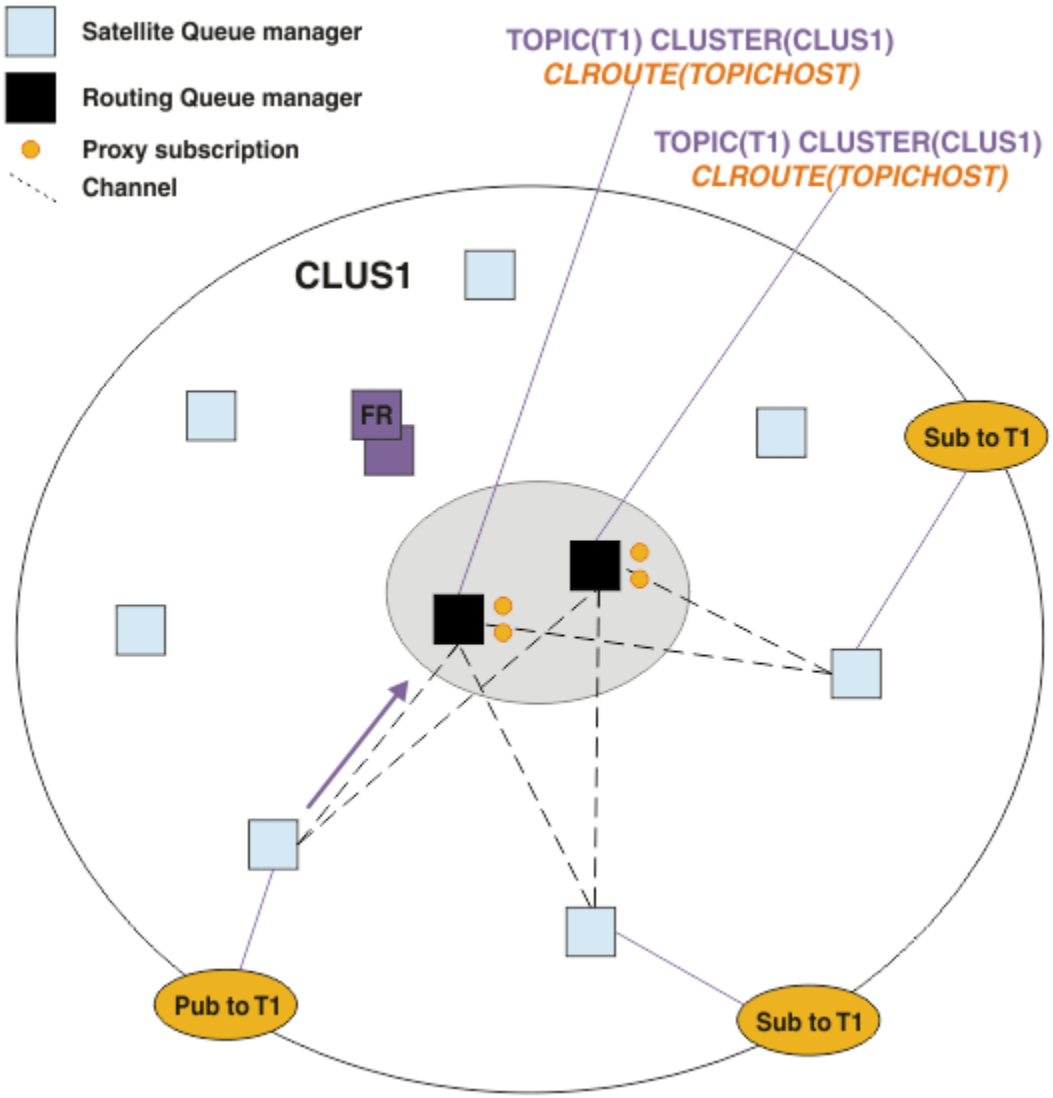


그림 *26.* 다중 토픽 호스트 발행/구독 클러스터의 발행물 수신

선택된 토픽 호스트 큐 관리자에 대한 인바운드 발행물은은 일치하는 프록시 구독을 등록한 모든 큐 관리자에게 전달됩니다.

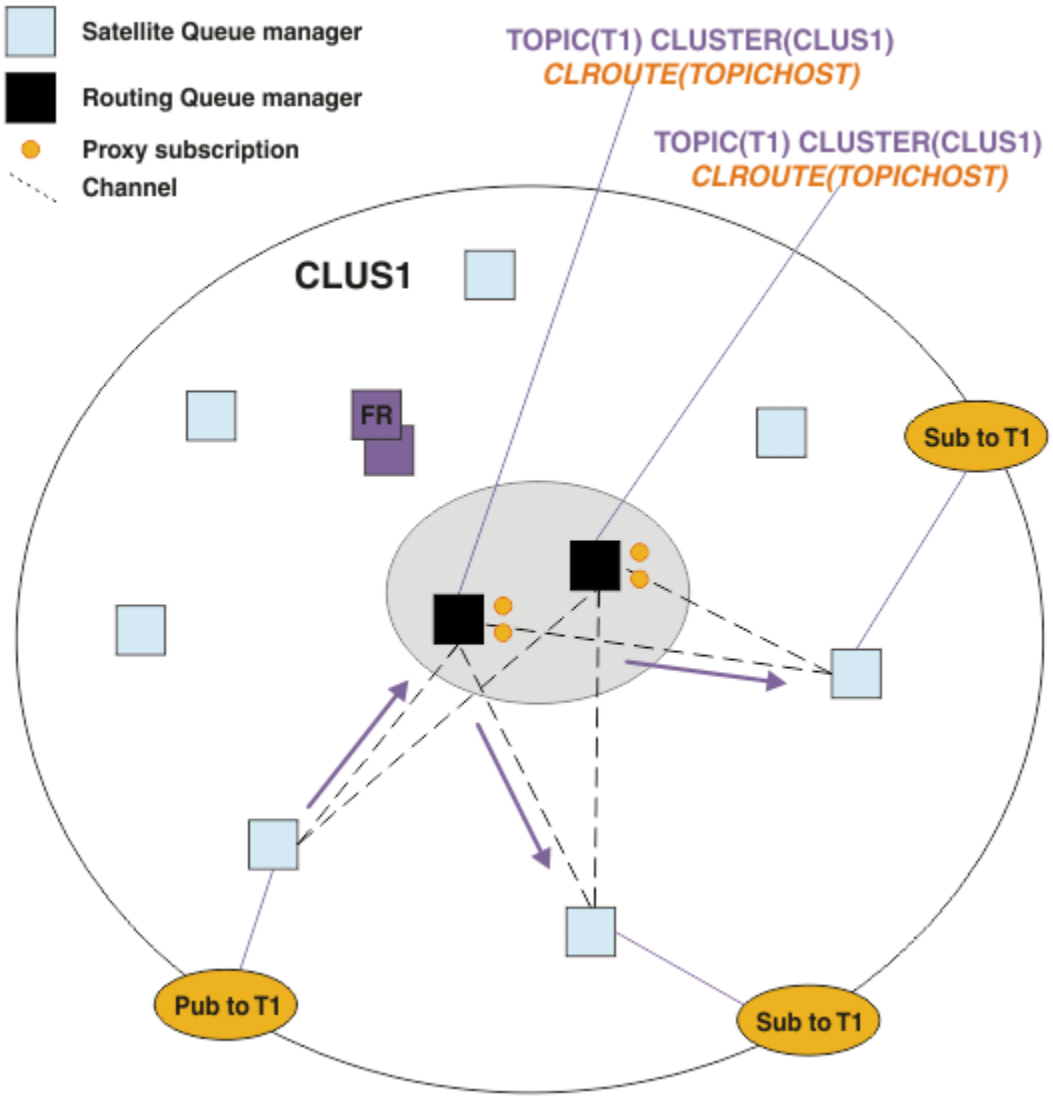


그림 27. 다중 토픽 호스트 발행/구독 클러스터의 구독자로 발행물 라우팅

토픽 호스트 큐 관리자에 로컬화된 구독 및 발행자

위의 예는 관리되는 라우트 토픽 오브젝트를 호스트하지 않는 큐 관리자에서 발행자와 구독자 간의 라우팅을 보여줍니다. 이 토폴로지에서는 메시지가 구독에 도달하기까지 여러 홉이 필요합니다.

추가 홉을 원하지 않으면 키 발행자를 토픽 호스트 큐 관리자에 연결하는 것이 적합할 수 있습니다. 하지만 토픽에 대한 여러 토픽 호스트가 있고 발행자는 하나 뿐인 경우에는 발행자가 연결된 토픽 호스트 큐 관리자를 통해 모든 발행 트래픽이 라우팅됩니다.

이와 마찬가지로, 키 구독이 있는 경우 토픽 호스트 큐 관리자에 있을 수 있습니다. 하지만 라우팅된 토픽의 여러 호스트가 있으면 극히 일부분의 발행물만 추가 홉을 피하고 나머지는 먼저 기타 토픽 호스트 큐 관리자를 통해 라우팅됩니다.

이와 같은 토폴로지는 집중식 발행자 또는 구독자를 사용한 토픽 호스트 라우팅에서 추가로 설명됩니다.

참고: 발행자 또는 구독이 라우팅된 토픽 호스트와 함께 위치할 때 라우팅된 토픽 구성을 변경하는 경우 특수 계획이 필요합니다. 예를 보려면 토픽 호스트 라우트 클러스터에 여분의 토픽 호스트 추가를 참조하십시오.

요약 및 추가 고려사항

토픽 호스트 라우트 발행/구독 클러스터는 어느 큐 관리자가 각 토픽을 호스트하고 이 큐 관리자가 토픽 트리의 해당 분기에 대한 라우팅 큐 관리자가 되는지 면밀히 제어합니다. 또한 구독 또는 발행자가 없는 큐 관리자는 토

픽 호스트 큐 관리자와 연결할 필요가 없고, 구독이 있는 큐 관리자는 토픽을 호스트하지 않는 큐 관리자에 연결할 필요가 없습니다. 이 구성은 클러스터의 큐 관리자 간 연결 수와 큐 관리자 간에 전달되는 정보의 양을 상당히 감소시킬 수 있습니다. 큐 관리자의 한 서브세트만이 발행/구독 작업을 수행 중인 대형 클러스터에 이 구성이 특히 유효합니다. 이 구성은 클러스터의 개별 큐 관리자에 대한 로드도 일부 제어하므로 (예를 들어,) 보다 강력하고 보다 탄력적인 시스템에서 매우 활성화된 토픽을 호스트하도록 선택할 수 있습니다. 특정 구성의 경우 - 특히, 대형 클러스터 - 일반적으로 직접 라우팅보다 더 적합한 토폴로지입니다.

그러나 토픽 호스트 라우팅을 사용하는 경우에도 특정 제한조건이 시스템에 적용됩니다.

- 직접 라우팅보다 시스템 구성과 유지보수에 더 많은 계획이 필요합니다. 토픽 트리에서 클러스터할 지점 및 클러스터의 토픽 정의 위치를 결정해야 합니다.
- 직접 라우트 토픽과 마찬가지로, 새 토픽 호스트 라우트 토픽을 정의할 경우 정보가 전체 저장소 큐 관리자로 푸시되어 거기에서 클러스터의 모든 멤버에게 전달됩니다. 이 이벤트는 채널이 아직 시작되지 않은 경우 전체 저장소에서 클러스터의 각 멤버로 시작되도록 합니다.
- 클러스터에 구독이 없는 경우에도 발행물이 항상 비호스트 큐 관리자에서 호스트 큐 관리자로 송신됩니다. 따라서 일반적으로 구독이 있을 것으로 예상되는 경우 또는 글로벌 연결 및 지식의 오버헤드가 추가 발행 트래픽의 위험보다 클 경우 라우트 토픽을 사용해야 합니다.

  참고: 이전에 설명한 대로, 발행자를 토픽 호스트에 로컬화하면 이 위험이 감소할 수 있습니다.
- 비호스트 큐 관리자에서 발행되는 메시지는 구독을 호스팅하는 큐 관리자로 전달되지 않습니다. 이러한 메시지는 항상 토픽 호스트 큐 관리자를 통해 라우트됩니다. 이 접근방식을 사용할 경우 클러스터에 대한 총 오버헤드와 메시지 대기 시간이 증가하고 성능이 저하될 수 있습니다.

  참고: 이전에 설명한 대로, 구독 또는 발행자를 토픽 호스트에 로컬화하면 이 위험이 감소할 수 있습니다.
- 하나의 토픽 호스트 큐 관리자를 사용할 경우 토픽에 발행되는 모든 메시지에 대해 단일 실패 지점이 제공됩니다. 이 단일 실패 지점은 토픽 호스트를 여러 개 정의하여 제거할 수 있습니다. 그러나 호스트가 여러 개 있으면 구독을 통해 수신되는 발행 메시지의 순서에 영향을 미칩니다.
- 토픽 호스트 큐 관리자가 여러 큐 관리자의 발행 트래픽을 처리해야 하기 때문에 토픽 호스트 큐 관리자에서 추가 메시지 로드가 발생합니다. 이 로드를 줄이려면 하나의 토픽에 대해 여러 개의 토픽 호스트를 사용하거나(메시지 순서가 유지되지 않는 경우), 토픽 트리의 여러 분기에 대한 라우트 토픽을 호스팅할 여러 큐 관리자를 사용하십시오.

토픽 호스트 라우팅을 사용하기 전에 68 페이지의 『발행/구독 클러스터의 직접 라우팅』 및 93 페이지의 『발행/구독 계층의 라우팅』에 설명된 대체 접근법을 탐색하십시오.

발행/구독 클러스터링: 우수 사례

클러스터된 토픽을 사용하면 큐 관리자 간의 발행/구독 도메인 확장이 단순해지지만 그 역학 및 영향을 완전히 이해하지 못하면 문제가 초래될 수 있습니다. 정보 공유 및 발행 라우팅에 대한 두 개의 모델이 있습니다. 개별 비즈니스 요구에 가장 잘 맞는 모델을 구현하면 선택한 클러스터에서 최상으로 수행됩니다.

다음 절의 우수 사례 정보는 두루 적용되는 솔루션을 제공하기보다는 공통 문제점을 해결하는 일반적인 접근법을 공유합니다. 사용자가 IBM MQ 클러스터 및 발행/구독 메시징을 기본적으로 이해하고 있으며 분산 발행/구독 네트워크 및 66 페이지의 『발행/구독 클러스터 디자인』의 정보에 익숙하다고 가정합니다.

포인트-투-포인트 메시징을 위해 클러스터를 사용하는 경우 클러스터의 각 큐 관리자는 필요할 때 필요한 것만 알려주는 방식으로 작동합니다. 즉, 큐 관리자에 연결하는 애플리케이션이 사용을 요청할 때 클러스터의 다른 큐 관리자 및 클러스터된 큐와 같은 기타 클러스터 자원에 대해서만 알아냅니다. 클러스터에 발행/구독 메시징을 추가하면 클러스터 큐 관리자 간의 정보 및 연결 공유 레벨이 증가합니다. 발행/구독 클러스터에 대한 우수 사례를 따르려면 이러한 작동 변경의 영향을 완전히 이해해야 합니다.

최상의 아키텍처를 빌드하기 위해, 정확한 필요에 기반을 둔 발행/구독 클러스터의 정보 공유 및 발행 라우팅에 대한 두 개의 모델, 직접 라우팅 및 토픽 호스트 라우팅이 있습니다. 올바른 선택을 하려면 각 모델이 충족시키는 상이한 요구사항과 두 모델을 모두 이해해야 합니다. 이 요구사항은 63 페이지의 『분산 발행/구독 네트워크 계획』과 함께 다음 절에서 논의됩니다.

- 82 페이지의 『발행/구독 활동에 관련된 클러스터 큐 관리자의 수를 제한하는 이유』
- 82 페이지의 『클러스터링할 토픽 결정 방법』
- 83 페이지의 『시스템 크기 조절 방법』
- 83 페이지의 『발행자 및 구독 위치』

- 84 페이지의 『발행 트래픽』
- 84 페이지의 『구독 변경 및 동적 토픽 문자열』

발행/구독 활동에 관련된 클러스터 큐 관리자의 수를 제한하는 이유

클러스터에서 발행/구독 메시징을 사용할 때에는 용량 및 성능 고려사항이 있습니다. 따라서 큐 관리자 전반의 발행/구독 활동에 대한 필요를 주의깊게 고려하고 이를 필요로 하는 큐 관리자의 수만으로 이를 제한하는 것이 우수 사례입니다. 토픽을 발행 및 구독해야 하는 큐 관리자의 최소 세트가 식별되고 나면 이러한 큐 관리자를 클러스터의 멤버로 만들 수 있습니다. 이 클러스터에는 이러한 큐 관리자만 포함되고 다른 큐 관리자는 포함되지 않습니다.

이 접근법은 포인트-투-포인트 메시징에 이미 잘 기능하는 설정된 클러스터가 있는 경우에 특히 유용합니다. 기존의 대형 클러스터를 발행/구독 클러스터로 전환할 때에는 현재 클러스터를 사용하지 않고 초기에 발행/구독 작업을 위한 별도의 클러스터를 작성하는 것이 더 나은 사례입니다. 하나 이상의 포인트-투-포인트 클러스터에 이미 있는 기존 큐 관리자 서브세트를 사용하고 이 서브세트를 새 발행/구독 클러스터의 멤버로 만들 수 있습니다. 하지만 새 클러스터의 전체 저장소 큐 관리자는 다른 클러스터의 멤버가 되면 안됩니다. 기존 클러스터 전체 저장소에서 추가 로드를 분리시키기 때문입니다.

새 클러스터를 작성할 수 없으며 기존 대형 클러스터를 발행/구독 클러스터로 전환해야 하는 경우 직접 라우트 모델을 사용하지 마십시오. 토픽 호스트 라우트 모델은 일반적으로 발행/구독 정보 공유 및 발행/구독 작업을 활발하게 수행 중인 큐 관리자 세트에 대한 연결을 제한하고 토픽을 호스트하는 큐 관리자에 집중하기 때문에 보통, 대형 클러스터에서는 토픽 호스트 라우트 모델이 더 잘 수행합니다. 이에 대한 예외는 토픽 정의를 호스트하는 큐 관리자에서 구독 정보의 수동 새로 고치기가 호출되어 토픽 호스트 큐 관리자가 클러스터의 모든 큐 관리자에 연결하는 경우입니다. 프록시 구독 재동기화를 참조하십시오.

클러스터를 크기나 현재 로드로 인해 발행/구독에 사용할 수 없다고 설정하는 경우 우수 사례는 이 클러스터가 예기치 않게 발행/구독 클러스터가 되지 않도록 하는 것입니다. **PSCLUS** 큐 관리자 특성을 사용하여 클러스터의 큐 관리자에 클러스터된 토픽을 추가하지 못하게 하십시오. 90 페이지의 『클러스터된 발행/구독 금지』의 내용을 참조하십시오.

클러스터링할 토픽 결정 방법

클러스터에 추가되는 토픽을 주의깊게 선택하는 것이 중요합니다. 토픽이 토픽 트리의 위에 있을수록 토픽의 용도가 보다 광범위하게 됩니다. 이로 인해 필요 이상의 구독 정보와 발행물이 전파됩니다. 토픽 트리에 일부 분기는 클러스터하고 일부 분기는 클러스터되지 않아야 하는 여러 뚜렷한 분기가 있는 경우 클러스터링이 필요한 각 분기의 루트에 관리 토픽 오브젝트를 작성하고 이 오브젝트를 클러스터에 추가하십시오. 예를 들어, /A, /B, /C 분기의 클러스터링을 해야 하면 각 분기마다 별도의 클러스터된 토픽 오브젝트를 정의하십시오.

참고: 시스템은 클러스터된 토픽 정의가 토픽 트리에 중첩되지 않게 합니다. 각 하위 분기마다 토픽 트리의 한 지점에서만 토픽을 클러스터링할 수 있습니다. 예를 들어, /A 및 /A/B에 대한 클러스터된 토픽 오브젝트를 정의할 수 없습니다. 클러스터된 토픽을 중첩하면 특히, 구독에 와일드카드가 사용될 때 어느 구독에 어느 클러스터된 오브젝트를 적용할지 혼동을 일으킬 수 있습니다. 라우팅 의사결정이 토픽 호스트의 할당을 통해 면밀하게 정의되는 토픽 호스트 라우팅을 사용할 때 이는 더욱 중요합니다.

클러스터된 토픽을 토픽 트리의 위에 추가해야 하지만 클러스터된 지점의 아래에 있는 트리의 일부 분기에 클러스터된 작동이 필요하지 않으면 구독 및 발행 범위 속성을 사용하여 추가 토픽에 대한 구독 및 발행의 공유 레벨을 감소시킬 수 있습니다.

표시된 작동을 고려하지 않고 토픽 루트 노드를 클러스터에 넣으면 안됩니다. 토픽 문자열에 상위 레벨 규정자를 사용하여(예를 들어, /global 또는 /cluster) 가능하면 글로벌 토픽을 명확하게 하십시오.

루트 토픽 노드를 클러스터하지 않으려는 다른 이유가 있습니다. 루트 노드 SYSTEM.BASE.TOPIC 토픽 오브젝트에 대한 로컬 정의가 모든 큐 관리자에 있기 때문입니다. 클러스터의 한 큐 관리자에 이 오브젝트가 클러스터되면 다른 모든 큐 관리자가 이를 인식합니다. 하지만 동일한 오브젝트의 로컬 정의가 존재할 때에는 해당 특성이 클러스터 오브젝트를 대체합니다. 이로 인해 큐 관리자는 토픽이 클러스터되지 않은 것처럼 작동하게 됩니다. 이 문제를 해결하려면 SYSTEM.BASE.TOPIC의 모든 정의를 클러스터링해야 합니다. 직접 라우팅된 정의에는 이를 수행할 수 있지만 토픽 호스트 라우팅된 정의에는 수행할 수 없습니다. 수행할 경우 모든 큐 관리자가 토픽 호스트가 되기 때문입니다.

시스템 크기 조절 방법

발행/구독 클러스터는 일반적으로 상이한 패턴의 클러스터 채널을 통해 클러스터에서 포인트-투-포인트 메시징을 수행합니다. 포인트-투-포인트 모델은 '선택적' 모델이지만 발행/구독 클러스터는 특히 직접 라우트 토픽을 사용할 때 구독 팬 아웃에 대해 더 신중하지 못한 성향이 있습니다. 따라서 발행/구독 클러스터의 어느 큐 관리자가 클러스터 채널을 사용하여 다른 큐 관리자에 연결할지 및 어떤 상황에 처해 있는지를 식별하는 것이 중요합니다.

다음 표에는 정상 실행 조건에서 발행/구독 클러스터의 큐 관리자 역할에 따라, 발행/구독 클러스터의 각 큐 관리자에 예상되는 일반적인 클러스터 송신자 및 수신자 채널 세트가 나열되어 있습니다.

표 *5.* 각 라우팅 메소드에 대한 클러스터 송신자 및 수신자 채널.

큐 관리자 역할	직접 클러스터 수신자	직접 클러스터 송신자	토픽 클러스터 수신자	토픽 클러스터 송신자
전체 저장소	AllQmgrs	AllQmgrs	AllQmgrs	AllQMgrs
토픽 정의 호스트	해당사항 없음	해당사항 없음	AllSubs+AllPubs (1)	AllSubs (1)
구독이 작성됨	AllPubs (1)	AllQMgrs	AllHosts	AllHosts
발행자가 연결됨	AllSubs (1)	AllSubs (1)	AllHosts	AllHosts
발행자 또는 구독자가 없음	AllSubs (1)	없음 (1)	없음 (2)	없음 (2)

키:

AllQmgrs
클러스터에 있는 모든 큐 관리자로(부터)의 채널.

AllSubs
구독이 작성된 모든 큐 관리자로(부터)의 채널.

AllPubs
발행 애플리케이션이 연결된 모든 큐 관리자로(부터)의 채널.

AllHosts
클러스터된 토픽 오브젝트의 정의가 구성된 모든 큐 관리자로(부터)의 채널.

없음
클러스터에 발행/구독 메시징 용도만을 위한 다른 큐 관리자로(부터)의 채널이 없음.

참고:

1. 프록시 구독의 큐 관리자 새로 고치기가 이 큐 관리자로부터 이루어지는 경우 클러스터의 다른 모든 큐 관리자로(부터)의 채널이 자동으로 작성될 수 있습니다.
2. 프록시 구독의 큐 관리자 새로 고치기가 이 큐 관리자로부터 이루어지는 경우 클러스터에서 클러스터된 토픽의 정의를 호스트하는 다른 모든 큐 관리자로(부터)의 채널이 자동으로 작성될 수 있습니다.

위의 표는 토픽 호스트 라우팅이 일반적으로 직접 라우팅보다 상당히 적은 클러스터 송신자 및 수신자 채널을 사용함을 보여줍니다. 클러스터의 특정 큐 관리자에 대한 채널 연결이 관심사인 경우에는 특정 채널을 설정하는 능력(예를 들어, 방화벽을 통해)이나 용량을 이유로, 선호하는 솔루션은 토픽 호스트 라우팅입니다.

발행자 및 구독 위치

클러스터된 발행/구독을 사용하여 한 큐 관리자에 발행된 메시지를 클러스터에 있는 다른 큐 관리자의 구독에 전달할 수 있습니다. 포인트-투-포인트 메시징의 경우 큐 관리자 간의 메시지 전송 비용이 성능에 해가 될 수 있습니다. 따라서 토픽에 대한 구독을 메시지가 발행되고 있는 동일한 큐 관리자에 작성할 것을 고려해야 합니다.

클러스터 내에서 토픽 호스트 라우팅을 사용할 때에는 토픽 호스트 큐 관리자에 관해서 구독 및 발행자 위치를 고려하는 것도 중요합니다. 클러스터된 토픽의 호스트인 큐 관리자에 발행자가 연결되어 있지 않으면 발행된 메시지가 항상 토픽 호스트 큐 관리자에게 송신됩니다. 마찬가지로, 클러스터된 토픽의 토픽 호스트가 아닌 큐 관리자에 구독이 작성되면 클러스터의 다른 큐 관리자에서 발행된 메시지가 항상 토픽 호스트 큐 관리자에게 첫 번째로 송신됩니다. 보다 구체적으로는, 토픽을 호스트하는 큐 관리자에 구독이 있지만 동일한 토픽을 호스트하는 하나 이상의 다른 큐 관리자가 있을 경우 다른 큐 관리자로부터의 발행물 일부가 다른 토픽 호스트 큐 관리자를 통해

라우팅됩니다. 발행자와 구독 간 거리를 최소화하기 위한 토픽 호스트 라우트 발행/구독 클러스터 설계에 대한 자세한 정보는 집중식 발행자 또는 구독자를 사용한 토픽 호스트 라우팅을 참조하십시오.

발행 트래픽

클러스터의 한 큐 관리자에 연결된 애플리케이션이 발행한 메시지는 클러스터 송신자 채널을 사용하여 다른 큐 관리자의 구독에 전송됩니다.

직접 라우팅을 사용할 때에는 발행된 메시지가 큐 관리자 사이의 최단 경로를 사용합니다. 즉, 발행 큐 관리자에서 구독이 있는 각 큐 관리자로 직접 이동합니다. 토픽에 대한 구독이 없는 큐 관리자에는 메시지가 전송되지 않습니다. 발행/구독 네트워크의 프록시 구독을 참조하십시오.

클러스터의 한 큐 관리자와 또 다른 큐 관리자 간 메시지 발행 비율이 높은 경우 두 지점 사이의 클러스터 채널 인프라가 이 비율을 유지보수할 수 있어야 합니다. 사용 중인 전송 큐와 채널의 성능 조정이 여기에 포함될 수 있습니다.

토픽 호스트 라우팅을 사용하면 토픽 호스트가 아닌 큐 관리자에서 발행된 각 메시지가 토픽 호스트 큐 관리자에 전송됩니다. 이와 같은 사실은 클러스터의 어딘가에 하나 이상의 구독이 존재하는지 여부와 무관합니다. 계획 시 고려해야 할 다음과 같은 추가 요인이 초래됩니다.

- 토픽 호스트 큐 관리자에 처음으로 각 발행물을 보낼 때의 추가 대기 시간이 허용 가능합니까?
- 각 토픽 호스트 큐 관리자가 인바운드 및 아웃바운드 발행률을 지탱할 수 있습니까? 여러 다른 큐 관리자에 발행자가 있는 시스템을 고려해보십시오. 이들 모두가 소수의 토픽 호스트 큐 관리자 세트에 메시지를 보내면 토픽 호스트는 메시지를 처리하고 구독하는 큐 관리자로 라우팅하는 과정에서 병목이 될 수 있습니다.
- 상당한 비율의 발행된 메시지에 일치하는 구독자가 없을 것이라 예상됩니까? 없다고 예상되며 이러한 메시지의 발행률이 높으면 발행자의 큐 관리자를 토픽 호스트로 만드는 것이 최상의 방법일 수 있습니다. 이 상황에서, 클러스터에 구독이 존재하지 않는 발행된 메시지는 다른 큐 관리자에 전송되지 않습니다.

여러 토픽 호스트를 도입하여 발행 로드를 분산시켜서 이 문제를 완화시킬 수도 있습니다.

- 뚜렷한 다중 토픽이 있고 각 토픽마다 발행 트래픽의 일부가 있으면 상이한 큐 관리자에서 토픽을 호스트할 것을 고려하십시오.
- 여러 다른 토픽 호스트로 토픽을 분리할 수 없는 경우에는 동일한 토픽 오브젝트를 다수의 큐 관리자에 정의할 것을 고려하십시오. 그러면 큐 관리자 각각에서 라우팅을 위한 발행물의 워크로드 밸런스가 조절됩니다. 하지만 이 방법은 발행 메시지 순서화가 필요하지 않을 때에만 적절합니다.

구독 변경 및 동적 토픽 문자열

또 다른 고려사항은 프록시 구독을 전파하기 위한 시스템의 성능에 미치는 영향입니다. 일반적으로 큐 관리자는 클러스터된 특정 토픽 문자열(구성된 토픽 오브젝트 뿐이 아니라)에 대한 첫 번째 구독이 이 큐 관리자에 작성될 때 클러스터의 기타 특정 큐 관리자에 프록시 구독 메시지를 보냅니다. 마찬가지로, 클러스터된 특정 토픽 문자열에 대한 마지막 구독이 삭제될 때에도 프록시 구독 삭제 메시지가 송신됩니다.

직접 라우팅의 경우 구독이 있는 각 큐 관리자는 클러스터의 다른 모든 큐 관리자에 프록시 구독을 보냅니다. 토픽 호스트 라우팅의 경우에는 구독이 있는 각 큐 관리자가 이 클러스터된 토픽에 대한 정의를 호스트하는 각 큐 관리자에만 프록시 구독을 보냅니다. 따라서 직접 라우팅에서는 클러스터에 큐 관리자가 많을수록 이들 사이의 프록시 구독을 유지보수하는 오버헤드가 더 커집니다. 반면에, 토픽 호스트 라우팅에서는 클러스터의 큐 관리자 수가 요인이 아닙니다.

두 라우팅 모델 모두, 발행/구독 솔루션이 구독 중인 여러 고유 토픽 문자열로 이루어져 있거나 클러스터의 큐 관리자에 대한 토픽이 자주 구독 및 구독 취소되면 프록시 구독의 분배와 삭제에 관한 끊임없는 메시지 생성으로 인해 해당 큐 관리자에 상당한 오버헤드가 표시됩니다. 직접 라우팅에서는 클러스터의 모든 큐 관리자에 이 메시지를 보내야 한다는 점 때문에 이 문제가 더 악화됩니다.

구독의 변경률이 너무 높아서 토픽 호스트 라우트 시스템 내에서조차도 이를 수행할 수 없는 경우 프록시 구독 오버헤드를 줄이기 위한 방법에 대한 정보는 발행/구독 네트워크의 구독 성능을 참조하십시오.

클러스터 토픽 정의

클러스터 토픽은 **cluster** 속성이 정의된 관리 토픽입니다. 클러스터 토픽에 대한 정보는 클러스터의 모든 멤버에게 푸시된 다음 로컬 토픽과 결합되어 여러 큐 관리자에 걸쳐 있는 토픽 공간의 일부분을 구성합니다. 따라서 한 큐 관리자의 토픽에 발행된 메시지를 클러스터에 있는 다른 큐 관리자의 구독으로 전달할 수 있습니다.

큐 관리자에 대한 클러스터 토픽을 정의하면 클러스터 토픽 정의가 전체 저장소 큐 관리자로 송신됩니다. 그러면 전체 저장소에서 클러스터 토픽 정의가 클러스터 내의 모든 큐 관리자로 전파되므로 클러스터의 모든 큐 관리자에 있는 발행자와 구독자가 동일한 클러스터 토픽을 사용할 수 있습니다. 클러스터 토픽이 작성된 큐 관리자를 클러스터 토픽 호스트라고 합니다. 클러스터 토픽은 클러스터의 모든 큐 관리자에서 사용할 수 있지만, 해당 토픽이 정의된 큐 관리자(호스트)에서 수정해야 합니다. 이 경우 수정사항이 전체 저장소를 통해 클러스터의 모든 멤버에게 전파됩니다.

직접 라우팅을 사용할 경우 클러스터의 모든 큐 관리자가 동일한 방식으로 토픽 정의를 사용하기 때문에 클러스터된 토픽 정의의 위치가 시스템의 작동에 직접 영향을 미치지 않습니다. 따라서 토픽이 필요하면 클러스터의 멤버가 되고 전체 저장소 큐 관리자와 정기적으로 접속할 것으로 충분히 신뢰할 수 있는 시스템에 있는 큐 관리자에 토픽을 정의해야 합니다.

토픽 호스트 라우팅을 사용할 때에는 클러스터의 다른 큐 관리자가 이 큐 관리자에 채널을 작성하고 구독 정보와 발행물을 보내기 때문에 클러스터된 토픽 정의의 위치가 매우 중요합니다. 토픽 정의를 호스트할 최상의 큐 관리자를 선택하려면 토픽 호스트 라우팅을 이해해야 합니다. 72 페이지의 『발행/구독 클러스터의 토픽 호스트 라우팅』의 내용을 참조하십시오.

클러스터된 토픽과 로컬 토픽 오브젝트가 있으면 로컬 토픽이 우선합니다. 87 페이지의 『동일한 이름의 다중 클러스터 토픽 정의』의 내용을 참조하십시오.

클러스터 토픽을 표시하는 데 사용할 명령에 대한 정보는 관련 정보를 참조하십시오.

클러스터된 토픽 상속

일반적으로 클러스터된 발행/구독 토폴로지의 발행 및 구독 애플리케이션은 클러스터의 어느 큐 관리자에 연결되어 있는지와 무관하게 동일하게 작동할 것으로 예상됩니다. 이는 클러스터된 관리 토픽 오브젝트가 클러스터의 모든 큐 관리자로 전파되기 때문입니다.

관리 토픽 오브젝트는 토픽 트리에서 더 상층에 있는 다른 관리 토픽 오브젝트로부터 작동을 상속합니다. 토픽 매개변수의 명시적 값이 설정되지 않았을 때 이 상속이 발생합니다.

클러스터된 발행/구독의 경우 연결된 큐 관리자에 따라 발행자와 구독자가 다르게 작동할 가능성이 있으므로 이러한 상속을 고려하는 것이 중요합니다. 클러스터된 토픽 오브젝트가 더 높은 토픽 오브젝트로부터 상속하기 위한 매개변수를 남기지 않으면 클러스터의 상이한 큐 관리자에서 토픽이 다르게 작동할 수 있습니다. 이와 마찬가지로, 토픽 트리에서 클러스터된 토픽 오브젝트 아래에 정의되어 있는 로컬에 정의된 토픽 오브젝트는 더 아래에 있는 토픽이 여전히 클러스터되지만 로컬 오브젝트는 클러스터의 기타 큐 관리자와 차이가 있는 일정 방식으로 작동을 변경할 수 있음을 의미합니다.

와일드카드 구독

프록시 구독은 클러스터된 토픽 오브젝트(또는 그 아래)로 해석되는 토픽 문자열에 대한 로컬 구독이 이루어질 때 작성됩니다. 와일드카드 구독이 클러스터 토픽보다 높은 계층에서 이루어지면 일치하는 클러스터 토픽에 대한 프록시 구독이 클러스터 주변에 송신되지 않아서, 클라이언트의 기타 멤버로부터 발행물을 수신하지 않습니다. 하지만 로컬 큐 관리자로부터 발행물을 수신합니다.

하지만 또 다른 애플리케이션이 클러스터 토픽 또는 그 아래로 해석되는 토픽 문자열을 구독하는 경우에는 프록시 구독이 생성되고 발행물이 이 큐 관리자에게 전파됩니다. 원래의 더 높은 와일드카드 구독이 이 발행물의 적합한 수신자로 간주되어 사본을 수신합니다. 이 작동이 필요하지 않으면 클러스터된 토픽에 **WILDCARD(BLOCK)** 을 설정하십시오. 그러면 원래 와일드카드가 적합한 구독으로 간주되지 않아서 클러스터 토픽 또는 하위 토픽에 대한 발행물이(로컬 또는 클러스터의 어딘가에서) 수신되지 않습니다.

관련 개념

관리 토픽에 대한 작업

구독에 대한 작업

관련 참조

표시 주제

표시 TP상태

표시 등록

클러스터 토픽 속성
토픽 오브젝트에 클러스터 이름 속성 세트가 있으면 클러스터의 모든 큐 관리자에 토픽 정의가 전파됩니다. 각 큐 관리자는 전파된 토픽 속성을 사용하여 발행/구독 애플리케이션의 작동을 제어합니다.

토픽 오브젝트는 발행/구독 클러스터에 적용되는 많은 속성이 있습니다. 일부는 발행 및 구독 애플리케이션의 일반 작동을 제어하고 일부는 클러스터에서 토픽이 사용되는 방식을 제어합니다.

클러스터된 토픽 오브젝트 정의는 클러스터의 모든 큐 관리자가 이 정의를 제대로 사용할 수 있는 방식으로 구성해야 합니다.

예를 들어, 관리 구독(MDURMDL 및 MNDURMDL)에 사용할 모델 큐가 기본값 이외의 큐 이름으로 설정되면 관리 구독이 작성될 모든 큐 관리자에 이름 지정된 모델 큐를 정의해야 합니다.

마찬가지로, 속성이 ASPARENT로 설정된 경우 토픽의 작동은 클러스터의 각 개별 큐 관리자에 대한 토픽 트리의 상위 노드에 따라 달라집니다(관리 토픽 오브젝트 참조). 이로 인해 각기 다른 큐 관리자에서 발행하거나 구독할 때 상이한 작동이 발생할 수 있습니다.

클러스터에서 발행/구독 작동에 직접적으로 관련된 기본 속성은 다음과 같습니다.

CLROUTE
이 매개변수는 발행자가 연결된 큐 관리자와 일치하는 구독이 있는 큐 관리자 간의 메시지 라우팅을 제어합니다.

- 큐 관리자 간에 직접 또는 클러스터된 토픽의 정의를 호스트하는 큐 관리자를 통해 라우트를 구성합니다. 세부사항은 발행/구독 클러스터를 참조하십시오.
- **CLUSTER** 매개변수가 설정되어 있는 동안에는 **CLROUTE**를 변경할 수 없습니다. **CLROUTE**를 변경하려면 먼저 **CLUSTER** 특성을 공백으로 설정하십시오. 그러면 토픽을 사용하는 애플리케이션이 클러스터된 방식의 작동을 멈춥니다. 이를 통해 구독으로의 발행물 전달이 중단되어 변경을 수행하는 동안 발행/구독 메시징도 일시정지해야 합니다.

PROXYSUB
이 매개변수는 프록시 구독이 이루어지는 시기를 제어합니다.

- FIRSTUSE는 기본값으로, 분산 발행/구독 토폴로지에서 큐 관리자의 로컬 구독에 대한 응답으로 프록시 구독을 보내고 더 이상 필요하지 않을 때 이를 취소합니다. FIRSTUSE의 기본값에서 이 속성을 변경하려는 이유에 대한 자세한 내용은 개별 프록시 구독 전달 및 모든 위치에서 발행을 참조하십시오.
- 모든 위치에서 발행을 사용하려면 상위 레벨 토픽 오브젝트에 대한 **PROXYSUB** 매개변수를 FORCE로 설정합니다. 그러면 토픽 트리에서 이 토픽 오브젝트 아래 모든 토픽과 일치하는 단일 와일드카드 프록시 구독이 존재합니다.

참고: 대형 또는 사용 중인 발행/구독 클러스터에 **PROXYSUB(FORCE)** 속성을 설정하면 시스템 자원에 대한 과다한 로드가 발생할 수 있습니다. **PROXYSUB(FORCE)** 속성은 토픽이 정의된 큐 관리자 외에 모든 큐 관리자에 전파됩니다. 따라서 클러스터의 모든 큐 관리자가 와일드카드 프록시 구독을 작성하게 됩니다.

클러스터의 큐 관리자에 발행된 이 토픽에 대한 메시지 사본은 **CLROUTE** 설정에 따라, 토픽 호스트 큐 관리자를 통해 또는 직접 클러스터의 모든 큐 관리자에 송신됩니다.

토픽이 직접 라우팅되면 모든 큐 관리자가 다른 모든 큐 관리자로 향하는 클러스터 송신자 채널을 작성합니다. 토픽이 토픽 호스트 라우팅 방식일 때에는 클러스터의 모든 큐 관리자로부터 각 토픽 호스트 큐 관리자로 향하는 채널이 작성됩니다.

클러스터에 사용되는 **PROXYSUB** 매개변수에 대한 자세한 정보는 직접 라우트 발행/구독 성능을 참조하십시오.

PUBSCOBE 및 SUBSCOPE
이 매개변수는 이 큐 관리자가 토폴로지(발행/구독 클러스터 또는 계층)의 큐 관리자에 발행물을 전파하거나 로컬 큐 관리자로만 범위를 제한하는지 여부를 판별합니다. MQPMO_SCOPE_QMGR 및 MQSO_SCOPE_QMGR을 사용하여 동등한 작업을 프로그래밍 방식으로 수행할 수 있습니다.

PUBSCOPE
클러스터 토픽 오브젝트가 **PUBSCOPE(QMGR)**로 정의된 경우 클러스터와 정의가 공유되지만 이 토픽에 기반한 발행물의 범위는 로컬뿐이어서 클러스터의 다른 큐 관리자에 발행물이 송신되지 않습니다.

SUBSCOPE

클러스터 토픽 오브젝트가 **SUBSCOPE(QMGR)**로 정의된 경우 클러스터와 정의가 공유되지만 이 토픽에 기반한 구독의 범위는 로컬뿐이어서 클러스터의 다른 큐 관리자로 프록시 구독이 송신되지 않습니다.

두 속성은 일반적으로 큐 관리자가 특정 토픽에 대해 클러스터의 기타 멤버와 상호작용하지 못하게 하는 용도로 함께 사용됩니다. 큐 관리자가 클러스터의 기타 멤버로(부터) 이 토픽에 대한 발행물을 발행하지도, 수신하지도 않습니다. 토픽 오브젝트가 하위 토픽에 대해 정의된 경우에는 이 상황에서 발행 또는 구독이 차단되지 않습니다.

SUBSCOPE(ALL)을 설정하여 토픽의 클러스터된 버전을 사용하는 경우에는 토픽의 로컬 정의에 **SUBSCOPE**를 QMGR로 설정해도 클러스터의 기타 큐 관리자가 큐 관리자로 프록시 구독을 전파하지 못하게 차단되지 않습니다. 하지만 로컬 정의에 **PUBSCOPE**도 QMGR로 설정되면 이 큐 관리자로부터 프록시 구독이 송신되지 않습니다.

관련 개념
발행 범위
구독 범위

동일한 이름의 다중 클러스터 토픽 정의
클러스터에서 둘 이상의 큐 관리자에 동일하게 이름 지정된 클러스터 토픽 오브젝트를 정의할 수 있으며 특정 시나리오에서 이는 특정 작동을 가능하게 합니다. 동일한 이름의 여러 클러스터 토픽 정의가 존재할 때에는 대다수의 특성이 일치해야 합니다. 그렇지 않을 경우 불일치의 중요성에 따라 오류 또는 경고가 보고됩니다.

일반적으로 여러 클러스터 토픽 정의의 특성에 불일치가 있으면 경고가 발행되고 클러스터의 각 큐 관리자에 토픽 오브젝트 정의 중 하나가 사용됩니다. 어느 정의가 각 큐 관리자에 사용되는지는 클러스터의 큐 관리자에서 확정적 또는 일관적이지 않습니다. 이러한 불일치는 가능한 빨리 해결해야 합니다.

클러스터 설정 또는 유지보수 중 때로 동일하지 않은 다수의 클러스터 토픽 정의를 작성해야 할 때가 있습니다. 하지만 이는 일시적 방편으로만 유용하므로 잠재적 오류 조건으로 간주해야 합니다.

불일치가 감지되면 각 큐 관리자의 오류 로그에 다음 경고 메시지가 기록됩니다.

- Multi 멀티플랫폼에서는 AMQ9465 및 AMQ9466.
- z/OS z/OS에서는 CSQX465I 및 CSQX466I.

각 큐 관리자의 토픽 문자열에 대한 선택된 특성은 토픽 오브젝트 정의가 아닌 토픽 상태를 보고(예를 들어, **DISPLAY TPSTATUS**를 사용하여) 판별할 수 있습니다.

일부 상황에서 구성 특성의 충돌은 작성 중인 토픽 오브젝트를 중지하거나 불일치하는 오브젝트가 올바르지 않은 것으로 표시되고 클러스터 전체에 전파되지 않도록 하기에 충분히 심각합니다 (DISPLAY TOPIC 의 **CLSTATE** 참조). 이러한 상황은 토픽 정의의 클러스터 라우팅 특성(**CLROUTE**)에 충돌이 있을 때 발생합니다. 또한 토픽 호스트 라우트 정의의 일관성 중요도로 인해 이 문서의 후속 절에 자세히 설명된 대로, 추가 불일치가 거부됩니다.

오브젝트가 정의될 때 충돌이 감지되면 구성 변경이 거부됩니다. 나중에 전체 저장소 큐 관리자에서 감지되는 경우에는 큐 관리자 오류 로그에 다음 경고 메시지가 기록됩니다.

- Multi 멀티플랫폼의 경우: AMQ9879
- z/OS z/OS의 경우: CSQX879E.

동일한 토픽 오브젝트의 여러 정의가 클러스터에 정의되면 로컬에 정의된 정의가 원격 정의보다 우선합니다. 따라서 정의의 차이가 존재할 경우 다수의 정의를 호스트하는 큐 관리자가 서로 다르게 작동합니다.

다른 큐 관리자의 클러스터 토픽과 동일한 이름의 비클러스터 토픽 정의 효과

클러스터에 있는 큐 관리자에 클러스터되지 않은 관리 토픽 오브젝트를 정의하고 이와 동시에 상이한 큐 관리자의 클러스터된 토픽 정의와 동일하게 이름 지정된 토픽 오브젝트를 정의할 수 있습니다. 이 경우 로컬에 정의된 토픽 오브젝트가 동일한 이름의 모든 원격 정의보다 우선합니다.

이는 이 큐 관리자에서 사용할 때 토픽의 클러스터링 작동을 차단하는 효과가 있습니다. 즉, 구독이 원격 발행자로부터 발행물을 수신하지 못하고 발행자로부터의 메시지가 클러스터의 원격 구독에 전파되지 않을 수 있습니다.

이는 혼동을 일으키는 작동을 유발할 수 있으므로 이러한 시스템을 구성하기 전에 주의깊게 고려해야 합니다.

참고: 토픽이 클러스터되었을 때조차도 개별 큐 관리자가 클러스터 주변으로의 발행물 및 구독의 전파를 차단해야 하는 경우 대체 접근법은 발행 및 구독 범위를 로컬 큐 관리자로만 설정하는 것입니다. 86 페이지의 『클러스터 토픽 속성』의 내용을 참조하십시오.

직접 라우트 클러스터의 다중 클러스터 토픽 정의

직접 라우팅의 경우 일반적으로 둘 이상의 클러스터 큐 관리자에 동일한 클러스터 토픽을 정의하지 않습니다. 직접 라우팅은 정의된 큐 관리자와 무관하게 클러스터의 모든 큐 관리자에 토픽을 사용 가능하게 하기 때문입니다. 또한 여러 클러스터 토픽 정의를 추가하면 시스템 활동 및 관리 복잡도가 상당히 증가하여 사용자 실수가 초래될 가능성이 있습니다.

- 각 토픽이 정의되면 기타 클러스터 토픽 호스트 큐 관리자를 포함하여, 추가 클러스터 토픽 오브젝트가 클러스터의 기타 큐 관리자에 푸시됩니다.
- 클러스터의 특정 토픽에 대한 모든 정의가 동일해야 하며 그렇지 않을 경우 큐 관리자에 어떤 토픽 정의가 사용되는지 파악하기 어렵습니다.

또한 클러스터 토픽 정의는 전체 저장소 큐 관리자와 부분 클러스터 저장소의 다른 모든 큐 관리자에서 캐싱되므로 클러스터에서 토픽이 제대로 기능하기 위해 단일 호스트 큐 관리자가 계속해서 사용 가능할 필요는 없습니다. 자세한 정보는 직접 라우팅을 사용하는 토픽 호스트 큐 관리자 가용성을 참조하십시오.

토픽의 기존 호스트를 클러스터에서 제거할 때처럼, 두 번째 큐 관리자에 클러스터 토픽을 임시로 정의해야 할 수 있는 상황에 대해서는 다른 큐 관리자로 클러스터 토픽 정의 이동을 참조하십시오.

클러스터 토픽 정의를 변경해야 하는 경우에는 정의된 동일한 큐 관리자에서 주의하여 수정하십시오. 다른 큐 관리자에서 수정하려 시도하면 충돌하는 토픽 속성이 있는 두 번째 토픽 정의가 우발적으로 작성될 수 있습니다.

토픽 호스트 라우트 클러스터의 다중 클러스터 토픽 정의

클러스터 토픽이 토픽 호스트의 클러스터 라우트로 정의될 때에는 직접 라우팅된 토픽의 경우처럼 클러스터의 모든 큐 관리자에 토픽이 전파됩니다. 또한 이 토픽에 대한 모든 발행/구독 메시징이 토픽이 정의된 큐 관리자를 통해 라우팅됩니다. 따라서 클러스터에 있는 토픽 정의의 위치와 수가 중요합니다(72 페이지의 『발행/구독 클러스터의 토픽 호스트 라우팅』 참조).

적합한 가용성 및 확장성을 보장하기 위해서는 가능하면 여러 토픽 정의를 보유하는 것이 유용합니다. 토픽 호스트 라우팅을 사용하는 토픽 호스트 큐 관리자의 가용성을 참조하십시오.

클러스터에서 토픽 호스트 라우트 토픽의 추가 정의를 추가 또는 제거하는 경우 구성 변경 시의 메시지 플로우를 고려해야 합니다. 변경 시 클러스터에서 토픽에 대한 메시지가 발행되고 있으면 토픽 정의를 추가 또는 제거하기 위해 스테이지별 프로세스가 필요합니다. 다른 큐 관리자로 클러스터 토픽 정의 이동 및 토픽 호스트 라우트 클러스터에 여분의 토픽 호스트 추가를 참조하십시오.

이전에 설명한 대로 여러 정의의 특성이 일치해야 하며 **PUB** 매개변수는 가능한 예외입니다(다음 절에 설명됨). 발행물이 토픽 호스트 큐 관리자를 통해 라우팅될 때에는 여러 정의의 일관성이 훨씬 더 중요합니다. 따라서 토픽 호스트 클러스터 라우팅에 하나 이상의 토픽 정의가 구성된 경우에는 토픽 문자열 또는 클러스터 이름의 감지된 불일치가 거부됩니다.

참고: 토픽 호스트 라우팅에 기존 클러스터된 토픽 정의가 구성되어 있는 토픽 트리에서 또 다른 토픽의 위나 아래에 클러스터 토픽 정의를 구성하려 시도하면 이 정의도 거부됩니다. 이는 와일드카드 구독에 대한 발행물 라우팅을 모호하지 않고 명확하게 합니다.

PUB 매개변수의 특수 핸들링

PUB 매개변수는 애플리케이션이 토픽에 발행할 수 있는 시기를 제어하는 데 사용됩니다. 클러스터의 토픽 호스트 라우팅의 경우 발행물을 라우팅하는 데 사용되는 토픽 호스트 큐 관리자도 제어할 수 있습니다. 이러한 이유로 PUB 매개변수의 설정을 달리 해서, 클러스터에 동일한 토픽 오브젝트의 여러 정의를 보유하는 것이 허용됩니다.

토픽의 여러 원격 클러스터된 정의의 이 매개변수에 대한 설정이 다르면 토픽은 다음 조건이 일치하는 경우에 발행물을 구독에 송신 및 전달하도록 허용합니다.

- 발행자가 연결되어 있는 큐 관리자에 `PUB(DISABLED)`로 설정된 일치하는 토픽 오브젝트가 정의되어 있지 않습니다.

- 클러스터의 여러 토픽 정의 중 하나 이상이 PUB(ENABLED)로 설정되어 있거나 여러 토픽 정의 중 하나 이상이 PUB(ASPARENT)로 설정되어 있으며 발행자가 연결되고 구독이 정의된 로컬 큐 관리자가 토픽 트리의 상층에서 PUB(ENABLED)로 설정되어 있습니다.

토픽 호스트 라우팅에서는 토픽 호스트가 아닌 큐 관리자에 연결된 애플리케이션이 메시지를 발행할 경우 **PUB** 매개변수가 명시적으로 DISABLED로 설정되지 않은 토픽 호스트 큐 관리자에만 메시지가 라우팅됩니다. 따라서 PUB(DISABLED) 설정을 사용하여 특정 토픽 호스트를 통한 메시지 트래픽을 일시정지할 수 있습니다. 큐 관리자의 제거나 유지보수를 준비하기 위해 또는 토픽 호스트 라우트 클러스터에 여분의 토픽 호스트 추가에 설명된 이유로 이 작업을 수행하려 할 수 있습니다.

클러스터 토픽 호스트 큐 관리자의 가용성
토픽 호스트 큐 관리자가 사용 불가능한 경우 클러스터가 토픽에 대한 트래픽을 더 이상 처리할 수 없는 위험성이 최소화되도록 발행/구독 클러스터를 설계하십시오. 토픽 호스트 큐 관리자가 사용 불가능한 데 따른 효과는 클러스터에 토픽 호스트 라우팅 또는 직접 라우팅이 사용되는지 여부에 따라 다릅니다.

직접 라우팅을 사용하는 토픽 호스트 큐 관리자의 가용성

직접 라우팅의 경우 일반적으로 둘 이상의 클러스터 큐 관리자에 동일한 클러스터 토픽을 정의하지 않습니다. 직접 라우팅은 정의된 큐 관리자와 무관하게 클러스터의 모든 큐 관리자에 토픽을 사용 가능하게 하기 때문입니다. 직접 라우트 클러스터의 다중 클러스터 토픽 정의를 참조하십시오.

클러스터에서 클러스터된 오브젝트(예를 들어, 클러스터된 큐 또는 클러스터된 토픽)가 장기간 사용 불가능하게 될 때마다 클러스터의 기타 멤버에서 결국은 이 오브젝트에 대한 정보가 만기됩니다. 클러스터된 토픽의 경우 클러스터 토픽 호스트 큐 관리자가 사용 불가능하게 되면 토픽 호스트 큐 관리자가 전체 저장소 큐 관리자와 마지막으로 통신한 시점부터 최소 60일 동안, 기타 큐 관리자가 직접 클러스터 방식으로(즉, 리모트 큐 관리자의 구독에 발행물을 송신하여) 계속해서 토픽에 대한 발행/구독 요청을 처리합니다. 클러스터 토픽 오브젝트를 정의한 큐 관리자가 다시 사용 가능하게 되지 않으면 결국은 기타 큐 관리자의 캐싱된 토픽 오브젝트가 삭제되고 토픽이 로컬 토픽으로 되돌아갑니다. 이 경우 리모트 큐 관리자에 연결된 애플리케이션으로부터 구독이 발행물의 수신을 중단합니다.

클러스터 토픽 오브젝트를 정의하는 큐 관리자를 복구할 60일의 기간으로, 클러스터 토픽 호스트를 계속 사용 가능하게 하기 위한 특별한 조치가 거의 필요하지 않습니다(하지만 사용 불가능한 클러스터 토픽 호스트에 정의된 구독은 사용 가능하지 않음에 유의). 60일의 기간은 기술적 문제점을 해결하기에 충분하며 관리 오류로 인해서만 초과될 수 있습니다. 이 가능성을 완화하기 위해 클러스터 토픽 호스트가 사용 불가능한 경우 클러스터의 모든 멤버는 매시간, 캐싱된 클러스터 토픽 오브젝트를 새로 고치지 않았음을 알리는 오류 로그 메시지를 씁니다. 클러스터 토픽 오브젝트가 정의된 큐 관리자가 실행 중인지 확인하여 이 메시지에 응답하십시오. 클러스터 토픽 호스트 큐 관리자를 다시 사용 가능하게 할 수 없는 경우에는 클러스터의 다른 큐 관리자에 정확히 같은 속성으로 동일한 클러스터된 토픽 정의를 정의하십시오.

토픽 호스트 라우팅을 사용하는 토픽 호스트 큐 관리자의 가용성

토픽 호스트 라우팅에서는 토픽에 대한 모든 발행/구독 메시징이 토픽이 정의된 큐 관리자를 통해 라우팅됩니다. 이러한 이유로 클러스터에 있는 큐 관리자의 연속된 가용성을 고려하는 것이 매우 중요합니다. 토픽 호스트가 사용 불가능하게 되고 토픽에 대한 다른 호스트가 존재하지 않으면 토픽에 대한 발행자로부터 클러스터의 상이한 큐 관리자에 있는 구독자까지의 트래픽이 즉시 정지됩니다. 추가 토픽 호스트가 사용 가능할 경우 클러스터 큐 관리자는 메시지 라우트의 연속적 가용성을 제공하면서 이 토픽 호스트를 통해 새 발행 트래픽을 라우팅합니다.

직접 토픽의 경우 60일 후에 첫 번째 토픽 호스트가 여전히 사용 불가능하면 이 토픽 호스트의 토픽 정보가 클러스터에서 제거됩니다. 이 정의가 클러스터에서 이 토픽에 대한 남아 있는 마지막 정의인 경우에는 다른 모든 큐 관리자가 라우팅을 위한 토픽 호스트로의 발행물 전달을 중단합니다.

유용할 수 있도록 적합한 가용성 및 확장성을 보장하기 위해서는 가능하면 최소 두 개의 클러스터 큐 관리자에 각 토픽을 정의하십시오. 그러면 사용 불가능하게 된 주어진 토픽 호스트 큐 관리자에 대한 보호가 제공됩니다. 토픽 호스트 라우트 클러스터의 다중 클러스터 토픽 정의도 참조하십시오.

여러 토픽 호스트를 구성할 수 없고(예를 들어, 메시지 순서를 유지해야 하기 때문에) 토픽 호스트를 하나만 구성할 수 없는 경우에는(단일 큐 관리자의 가용성이 클러스터에 있는 모든 큐 관리자의 구독까지의 발행 플로우에 영향을 주어선 안되기 때문에) 토픽을 직접 라우팅된 토픽으로 구성할 것을 고려하십시오. 이는 전체 클러스터의 단일 큐 관리자에 대한 의존도를 피하지만 각 개별 큐 관리자가 로컬에 호스트된 구독 및 발행자를 처리하려면 여전히 사용 가능해야 합니다.

클러스터된 발행/구독 금지

첫 번째 직접 라우트 클러스터 토픽이 클러스터에 소개되면 클러스터의 모든 큐 관리자가 다른 모든 큐 관리자를 인식하게 되어 잠재적으로 이들 서로 간에 채널이 작성됩니다. 채널 작성을 원하지 않으면 대신에 토픽 호스트 라우트 발행/구독을 구성해야 합니다. 직접 라우트 클러스터 토픽이 각 큐 관리자의 스케일링 문제로 인해 클러스터의 안정성에 위해가 될 수 있으면 클러스터의 모든 큐 관리자에서 **PSCLUS**를 DISABLED로 설정하여 클러스터된 발행/구독 기능을 완전히 사용 불가능하게 할 수 있습니다.

68 페이지의 『발행/구독 클러스터의 직접 라우팅』에 설명된 대로, 직접 라우트 클러스터 토픽을 클러스터에 소개하면 모든 부분 저장소에 클러스터의 다른 모든 멤버가 자동으로 통지됩니다. 또한 클러스터 토픽은 다른 모든 노드(예를 들어, **PROXYSUB(FORCE)**가 지정된)에 구독을 작성하며 이로 인해 로컬 구독이 없어도 큐 관리자로부터 많은 수의 채널이 시작될 수 있습니다. 그러면 클러스터의 각 큐 관리자에 즉시 추가 로드가 발생합니다. 이 경우 많은 큐 관리자를 포함하는 클러스터에서는 성능이 크게 저하될 수 있습니다. 따라서 직접 라우팅된 발행/구독을 클러스터에 소개할 때에는 주의깊게 계획해야 합니다.

클러스터가 직접 라우트 발행/구독의 오버헤드를 수용할 수 없음이 파악되면 대신에 토픽 호스트 라우트 발행/구독을 사용할 수 있습니다. 차이점 개요는 66 페이지의 『발행/구독 클러스터 디자인』의 내용을 참조하십시오.

클러스터에 발행/구독 기능을 완전히 사용 불가능하게 하려는 경우 클러스터의 모든 큐 관리자에서 큐 관리자 속성 **PSCLUS**를 DISABLED로 설정하여 수행할 수 있습니다. 이 설정은 큐 관리자 기능의 세 가지 측면을 수정하여 클러스터에서 직접 라우트 및 토픽 호스트 라우트 발행/구독을 모두 사용 불가능하게 합니다.

- 이 큐 관리자의 관리자는 더 이상 Topic 오브젝트를 클러스터된 것으로 정의할 수 없습니다.
- 기타 큐 관리자로부터 수신되는 토픽 정의 또는 프록시 구독이 거부되고 관리자에게 올바르지 않은 구성을 알리는 경고 메시지가 기록됩니다.
- 전체 저장소는 토픽 정의를 수신할 때 모든 큐 관리자에 대한 정보를 다른 모든 부분 저장소와 더 이상 자동으로 공유하지 않습니다.

PSCLUS는 클러스터의 각 개별 큐 관리자 매개변수이지만 클러스터의 큐 관리자 서브세트에서 발행/구독을 선택적으로 사용 불가능하게 하는 용도로 사용되지 않습니다. 이러한 방식을 통해 선택적으로 사용 불가능하게 할 경우 오류 메시지가 자주 표시됩니다. **PSCLUS**가 사용되는 큐 관리자에서 토픽이 클러스터되면 프록시 구독 및 토픽 정의가 항상 표시되고 거부되기 때문입니다.

따라서 클러스터의 모든 큐 관리자에서 **PSCLUS**를 DISABLED로 설정해야 합니다. 하지만 큐 관리자가 언제든지 클러스터를 조인하고 떠날 수 있는 경우처럼 실제로 이 상태를 확립하고 유지하기는 어려울 수 있습니다. 적어고 큐 관리자의 모든 전체 저장소에서 **PSCLUS**가 DISABLED로 설정되었는지 확인해야 합니다. 이렇게 설정하고 클러스터된 토픽이 클러스터의 ENABLED 큐 관리자에 연속으로 정의되는 경우에는 전체 저장소가 모든 큐 관리자에게 다른 모든 큐 관리자를 알리지 않으므로 클러스터에서 모든 큐 관리자의 잠재적 스케일링 문제가 예방됩니다. 이 시나리오에서, 클러스터된 토픽의 원본은 전체 저장소 큐 관리자의 오류 로그에 보고됩니다.

큐 관리자가 하나 이상의 발행/구독 클러스터에 참여하고 하나 이상의 포인트-투-포인트 클러스터에도 참여하는 경우 해당 큐 관리자에서 **PSCLUS**를 ENABLED로 설정해야 합니다. 이러한 이유로 포인트-투-포인트 클러스터와 발행/구독 클러스터가 중첩될 때에는 각 클러스터마다 별도의 전체 저장소 세트를 사용해야 합니다. 이 접근법을 사용하면 토픽 정의와 모든 큐 관리자에 대한 정보가 발행/구독 클러스터에서만 플로우될 수 있습니다.

PSCLUS를 ENABLED에서 DISABLED로 변경할 때 구성 불일치를 피하기 위해 이 큐 관리자가 멤버로 속한 클러스터에 클러스터된 토픽 오브젝트가 존재할 수 없습니다. 이러한 토픽은 원격으로 정의된 토픽일지라도 **PSCLUS**를 DISABLED로 변경하기 전에 삭제해야 합니다.

PSCLUS에 대한 자세한 정보는 ALTER QMGR (PSCLUS)를 참조하십시오.

관련 개념

직접 라우트된 발행/구독 클러스터 성능

발행/구독 및 다중 클러스터

단일 큐 관리자는 둘 이상 클러스터의 멤버가 될 수 있습니다. 이 배열은 때로 중첩 클러스터라 합니다. 이러한 중첩을 통해 여러 클러스터에서 클러스터된 큐에 액세스하고 한 클러스터의 큐 관리자에서 또 다른 클러스터의 큐 관리자까지 포인트-투-포인트 메시지 트래픽을 라우팅할 수 있습니다. 발행/구독 클러스터의 클러스터된 토픽은 동일한 기능을 제공하지 않습니다. 따라서 다수의 클러스터를 사용할 때 이 토픽의 작동이 명확히 이해되어야 합니다.

큐와 다르게, 토픽 정의는 둘 이상의 클러스터와 연관시킬 수 없습니다. 클러스터된 토픽의 범위는 토픽이 정의된 동일한 클러스터의 큐 관리자로 제한됩니다. 이를 통해 동일한 클러스터에 있는 큐 관리자의 구독으로만 발행물이 전파됩니다.

큐 관리자의 토픽 트리

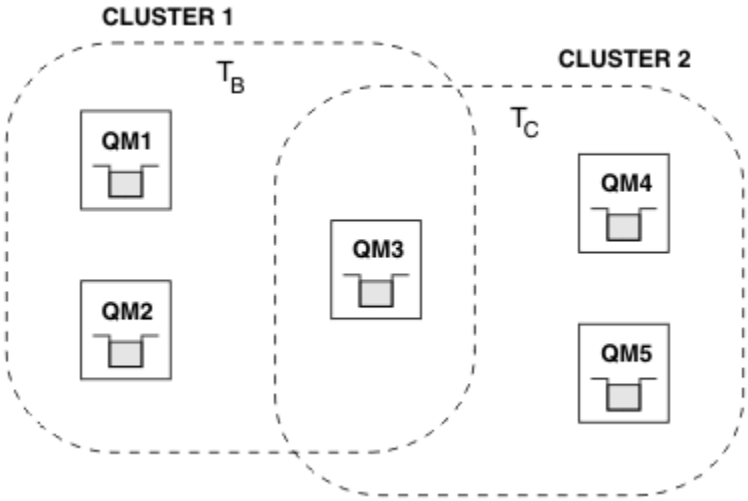


그림 *28.* 중첩 클러스터*:* 각각이 서로 다른 토픽을 구독하는 두 개의 클러스터

큐 관리자는 여러 클러스터의 멤버일 때 각 클러스터에 정의된 모든 클러스터된 토픽을 인식하게 됩니다. 예를 들어, 이전 그림에서 QM3은 T_B 및 T_C 관리 클러스터 토픽 오브젝트를 모두 인식하는 반면 QM1은 T_B만 인식합니다. QM3은 두 토픽 정의를 모두 해당 로컬 토픽에 적용하므로 특정 토픽에 대해 QM1에 다르게 작동합니다. 이러한 이유로 상이한 클러스터에서 클러스터된 토픽은 서로 간섭하지 않는 것이 중요합니다. 간섭은 클러스터된 토픽이 상이한 클러스터의 또 다른 클러스터된 토픽(예를 들어, 토픽 문자열이 `/Sport` 및 `/Sport/Football`임) 위나 아래에 정의될 때 또는 둘 모두의 동일한 토픽 문자열에 대해서도 발생할 수 있습니다. 또 다른 양식의 간섭은 관리 클러스터 토픽 오브젝트가 상이한 클러스터에 동일한 오브젝트 이름으로(상이한 토픽 문자열에 대해서) 정의될 때입니다.

이러한 구성에서는 일치하는 구독으로의 발행물 전달이 클러스터에 관한 발행자 및 구독자의 상대적 위치에 매우 의존하게 됩니다. 이 이유 때문에 이러한 구성에 의존할 수 없으며 간섭하는 토픽을 제거하도록 구성을 변경해야 합니다.

발행/구독 메시징과 함께 중첩 클러스터 토폴로지를 계획할 때 토픽 트리 및 클러스터된 토픽 오브젝트가 토폴로지의 모든 중첩 클러스터에 퍼져 있는 것처럼 처리해서 간섭을 피할 수 있습니다.

다중 발행/구독 클러스터 통합

발행/구독 메시징이 각기 다른 클러스터의 큐 관리자에 퍼져야 한다는 요구사항이 있는 경우 사용 가능한 두 가지 옵션이 있습니다.

- 발행/구독 계층 구성을 사용하여 클러스터를 함께 연결시키십시오. 다중 클러스터의 토픽 공간 결합을 참조하십시오.
- 기존 클러스터를 오버레이하고 특정 토픽을 발행 또는 구독해야 할 큐 관리자를 포함한 추가 클러스터를 작성하십시오.

후자 옵션의 경우 클러스터의 크기 및 가장 효과적인 클러스터 라우팅 메커니즘을 주의깊게 고려해야 합니다. 66 페이지의 『발행/구독 클러스터 디자인』의 내용을 참조하십시오.

발행/구독 클러스터에서 보유된 발행에 대한 디자인 고려사항

보유된 발행물에 대해 작업할 발행/구독 클러스터를 설계할 때 고려할 몇 가지 제한사항이 있습니다.

고려사항

고려사항 *1:* 다음 클러스터 큐 관리자는 항상 보유된 발행의 최신 버전을 저장합니다.

- 발행자의 큐 관리자
- 토픽 호스트 라우트 클러스터에서 토픽 호스트(이 문서의 다음 절에 설명된 대로 토픽의 토픽 호스트가 하나만 있는 경우)
- 보유된 발행의 토픽 문자열에 일치하는 구독이 있는 모든 큐 관리자

고려사항 *2:* 큐 관리자는 구독이 없을 때에는 업데이트된 보유된 발행을 수신하지 않습니다. 따라서 더 이상 토픽을 구독하지 않는 큐 관리자에 저장된 보유된 발행은 실효 상태가 됩니다.

고려사항 *3:* 구독을 작성할 때 토픽 문자열에 대한 보유된 발행의 로컬 사본이 있는 경우 로컬 사본이 구독으로 전달됩니다. 주어진 토픽 문자열의 첫 번째 구독자이면 다음 클러스터 멤버 중 하나에서 일치하는 보유된 발행도 전달됩니다.

- 직접 라우트 클러스터에서 발행자의 큐 관리자
- 토픽 호스트 라우트 클러스터에서 주어진 토픽의 토픽 호스트

토픽 호스트 또는 발행 큐 관리자에서 보유한 발행물을 구독 큐 관리자로 전달하는 작업은 MQSUB 호출과 비동기식으로 수행됩니다. 따라서 MQSUBRQ 호출을 사용하는 경우 MQSUBRQ에 대한 후속 호출이 수행될 때까지 최신 보유된 발행물이 누락될 수 있습니다.

관련사항

발행/구독 클러스터에서 첫 번째 구독이 이루어질 때 로컬 큐 관리자는 보유된 발행의 실효 사본을 저장할 수 있으며 이는 새 구독에 전달되는 사본입니다. 로컬 큐 관리자에 구독이 존재한다는 것은 다음 번에 보유된 발행이 업데이트될 때 이 구독이 해결된다는 의미입니다.

토픽 호스트 라우트 발행/구독 클러스터에서, 주어진 토픽에 대해 둘 이상의 토픽 호스트를 구성하면 새 구독자가 토픽 호스트로부터 최신 보유된 발행을 수신하거나 다른 토픽 호스트로부터(최신 발행이 유실된) 실효 상태의 보유된 발행을 수신할 수 있습니다. 토픽 호스트 라우팅의 경우 주어진 토픽에 대한 다수의 토픽 호스트를 구성하는 것이 일반적입니다. 하지만 애플리케이션이 보유된 발행을 사용할 것으로 예상되면 각 토픽마다 토픽 호스트를 하나만 구성해야 합니다.

주어진 토픽 문자열에 대해 단일 발행자만 사용하고 발행자가 항상 동일한 큐 관리자를 사용하는지 확인해야 합니다. 그렇지 않을 경우 동일한 토픽에 대해 상이한 보유된 발행이 각기 다른 큐 관리자에서 활성화되어 예기치 않은 작동이 초래될 수 있습니다. 여러 프록시 구독이 분배되면 다수의 보유된 발행이 수신될 수 있습니다.

실효 발행을 사용하는 구독자에 대해 여전히 관심이 있으면 각 보유된 발행을 작성할 때 메시지 만기를 설정할 것을 고려하십시오.

CLEAR TOPICSTR 명령을 사용하여 발행/구독 클러스터에서 보유된 발행을 제거할 수 있습니다. 특정 상황에서는 **CLEAR TOPICSTR**에 설명된 대로 발행/구독 클러스터의 여러 멤버에 명령을 발행해야 할 수 있습니다.

와일드카드 구독 및 보유된 발행

와일드카드 구독을 사용 중인 경우 발행/구독 클러스터의 다른 멤버에 전달된 해당 프록시 구독은 첫 번째 와일드카드 문자 바로 앞의 토픽 구분 기호에서 와일드 카드 처리됩니다. 와일드카드 및 클러스터 토픽을 참조하십시오.

따라서 사용된 와일드카드가 구독 애플리케이션을 일치시키는 것보다 더 많은 토픽 문자열과 보유된 발행을 일치시킬 수 있습니다.

이로 인해 보유된 발행에 필요한 스토리지 양이 증가하므로 호스트하는 큐 관리자에 충분한 스토리지 용량이 있는지 확인해야 합니다.

관련 개념

보유된 발행물

개별 프록시 구독 전달 및 모든 위치에서 발행

발행/구독 클러스터에 대한 *REFRESH CLUSTER* 고려사항

REFRESH CLUSTER 명령을 발행하면 큐 관리자가 클러스터 토픽 및 연관된 프록시 구독을 포함하여, 로컬에 보유된 클러스터에 대한 정보를 일시적으로 제거합니다.

REFRESH CLUSTER 명령 발행에서 큐 관리자가 클러스터된 발행/구독에 대한 필요한 전체 정보를 다시 수집한 시점까지 걸리는 시간은 전체 저장소 큐 관리자의 응답성, 가용성, 클러스터의 크기에 따라 다릅니다.

새로 고치기 처리 중에는 발행/구독 클러스터의 발행/구독 트래픽 방해가 발생합니다. 대형 클러스터의 경우, **REFRESH CLUSTER** 명령을 사용하면 진행 중에, 그리고 클러스터 오브젝트가 모든 관련 큐 관리자에 대한 상태 업데이트를 자동으로 송신한 후 27일간격으로 클러스터가 중단될 수 있습니다. 대형 클러스터를 새로 고치면 클러스터의 성능 및 가용성에 영향을 줄 수 있음을 참조하십시오. 이러한 이유로 **REFRESH CLUSTER** 명령은 IBM 지원 센터의 지시가 있을 때에만 발행/구독 클러스터에 사용해야 합니다.

클러스터 방해는 외부에서 보면 다음 증상으로 표시될 수 있습니다.

- 이 큐 관리자의 클러스터 토픽에 대한 구독이 클러스터의 기타 큐 관리자에 연결된 발행자로부터 발행물을 수신하지 않습니다.
- 이 큐 관리자의 클러스터 토픽에 발행된 메시지가 기타 큐 관리자의 구독에 전파되지 않습니다.
- 이 기간 중 작성된 이 큐 관리자의 클러스터 토픽에 대한 구독이 클러스터의 기타 멤버에 프록시 구독을 일관되게 보내지 않습니다.
- 이 기간 중 삭제된 이 큐 관리자의 클러스터 토픽에 대한 구독이 클러스터의 기타 멤버에서 프록시 구독을 일관되게 제거하지 않습니다.
- 메시지 전달 시 10초 이상 일시정지됩니다.
- **MQPUT** 실패(예: MQRC_PUBLICATION_FAILURE).
- 발행물이 데드-레터 큐에 배치되었습니다(이유: MQRC_UNKNOWN_REMOTE_Q_MGR).

위와 같은 이유로 **REFRESH CLUSTER** 명령을 발행하기 전에 발행/구독 애플리케이션을 일시정지해야 합니다.

발행/구독 클러스터의 큐 관리자에 **REFRESH CLUSTER** 명령이 발행되고 나면 모든 클러스터 큐 관리자와 클러스터 토픽이 새로 고쳐질 때까지 대기한 후 프록시 구독 재동기화에 설명된 대로 프록시 구독을 재동기화하십시오. 모든 프록시 구독이 제대로 재동기화되면 발행/구독 애플리케이션을 다시 시작하십시오.

REFRESH CLUSTER 명령 완료에 시간이 오래 걸리는 경우 SYSTEM.CLUSTER.COMMAND.QUEUE의 CURDEPTH를 보고 상황을 모니터하십시오.

관련 개념

61 페이지의 『클러스터링: REFRESH CLUSTER 사용 우수 사례』

REFRESH CLUSTER 명령을 사용하여 로컬에 보유된 클러스터에 대한 모든 정보를 제거하고 클러스터의 전체 저장소에서 이 정보를 다시 빌드합니다. 예외 상황을 제외하고는 이 명령을 사용해서는 안됩니다. 사용해야 하는 경우 사용 방법에 대한 특수 고려사항이 있습니다. 이 정보는 고객의 피드백과 테스트에 기반을 둔 지침입니다.

관련 참조

REFRESH CLUSTER를 실행할 때 표시되는 애플리케이션 문제

MQSC 명령 참조: REFRESH CLUSTER

발행/구독 계층의 라우팅

분산 큐 관리자 토폴로지가 발행/구독 계층이고 큐 관리자에서 구독이 이루어지는 경우 프록시 구독은 기본적으로 계층의 모든 큐 관리자에 작성됩니다. 큐 관리자에 수신된 발행물은 계층을 통해 일치하는 구독을 호스트하는 각 큐 관리자에 라우팅됩니다.

발행/구독 계층의 큐 관리자와 클러스터 간에 메시지가 라우팅되는 방식에 대한 소개는 분산 발행/구독 네트워크를 참조하십시오.

분산 발행/구독 계층에서 토픽에 대한 구독이 이루어지는 경우 큐 관리자는 연결된 큐 관리자에 구독이 전파되는 프로세스를 관리합니다. 프록시 구독은 네트워크의 모든 큐 관리자에 플로우됩니다. 프록시 구독은 해당 토픽에 대한 구독을 호스트하는 큐 관리자에 발행물을 전달해야 한다는 정보를 큐 관리자에게 제공합니다. 발행/구독 계층의 각 큐 관리자는 직접 관계만을 인식합니다. 하나의 큐 관리자에 넣은 발행물은 직접 관계를 통해 구독이 있는 큐 관리자에게 송신됩니다. 이는 다음 그림에 설명되어 있으며 여기서, *Subscriber 1*은 *Asia* 큐 관리자(1)에서

특정 토픽에 대한 구독을 등록합니다. *Asia* 큐 관리자의 이 구독에 대한 프록시 구독은 네트워크의 다른 모든 큐 관리자(2,3,4)에게 전달됩니다.

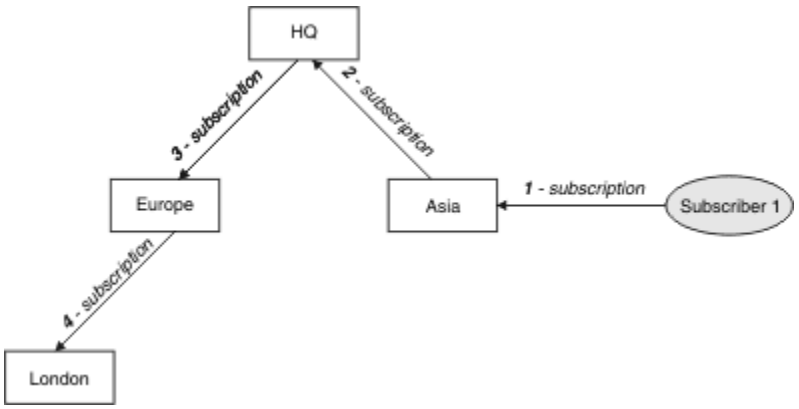


그림 *29.* 큐 관리자 네트워크를 통한 구독 전파

큐 관리자는 로컬 애플리케이션이나 리모트 큐 관리자로부터, 작성된 모든 구독을 통합합니다. 프록시 구독이 아직 존재하지 않으면 인접 항목으로 구독의 토픽에 대한 프록시 구독을 작성합니다. 이는 다음 그림에 설명되어 있으며 여기서, *Subscriber 2*는 *HQ* 큐 관리자(5)에서 94 페이지의 그림 29에서와 동일한 토픽에 대한 구독을 등록합니다. 이 토픽에 대한 구독은 네트워크(6)에 구독이 존재함을 인지하도록 *Asia* 큐 관리자에 전달됩니다. *Europe* 큐 관리자에는 이 토픽에 대한 구독이 이미 등록되어 있으므로 구독이 전달되지 않습니다(94 페이지의 그림 29의 3단계 참조).

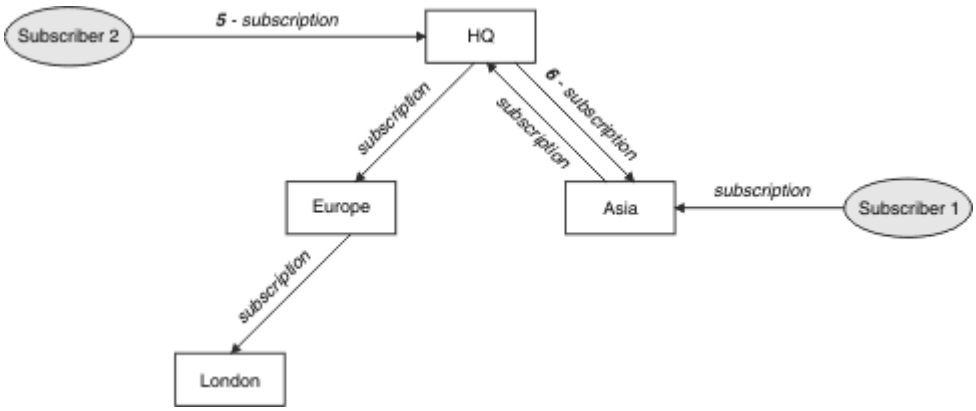


그림 *30.* 다중 구독

애플리케이션이 토픽에 정보를 발행할 때에는 기본적으로 수신 큐 관리자가 토픽에 대한 유효한 구독이 있는 모든 큐 관리자에게 이를 전달합니다. 하나 이상의 중간 큐 관리자를 통해 전달할 수 있습니다. 이는 다음 그림에 설명되어 있으며 여기서, 발행자는 94 페이지의 그림 30에서와 동일한 토픽에 대한 발행물을 *Europe* 큐 관리자(7)에게 보냅니다. 이 토픽에 대한 구독은 *HQ*에서 *Europe*까지 존재하므로 *HQ* 큐 관리자(8)에게 발행물이 전달됩니다. 하지만 *London*에서 *Europe*까지는 구독이 존재하지 않으므로(*Europe*에서 *London*까지만) *London* 큐 관리자에는 발행물이 전달되지 않습니다. *HQ* 큐 관리자는 *Subscriber 2* 및 *Asia* 큐 관리자(9)에게 직접 발행물을 보냅니다. *Asia*(10)에서 *Subscriber 1*로 발행물이 전달됩니다.

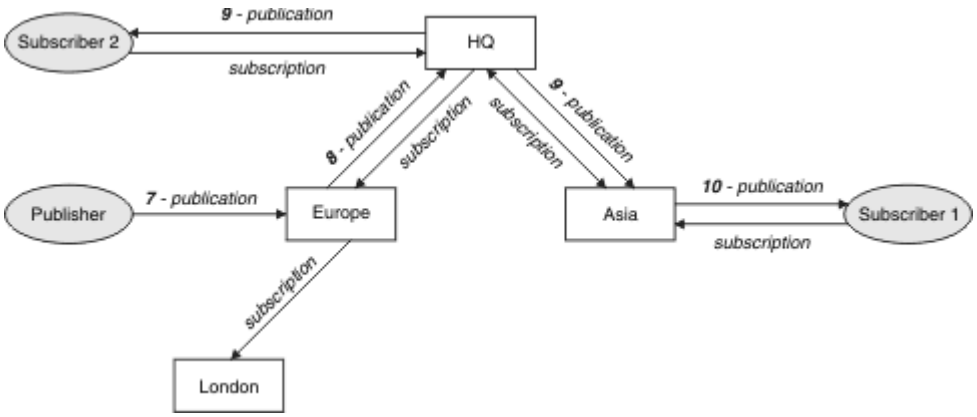


그림 *31.* 큐 관리자 네트워크를 통한 발행물 전파

큐 관리자는 다른 큐 관리자에 발행물이나 구독을 보낼 때 자체 사용자 ID를 메시지에 설정합니다. 발행/구독 계층을 사용하는 경우 및 수신 채널이 사용자 ID의 권한으로 메시지를 넣도록 설정된 경우에는 송신 큐 관리자의 사용자 ID에 권한을 부여해야 합니다. 큐 관리자 계층에 기본 사용자 ID 사용을 참조하십시오.

참고: 대신에 발행/구독 클러스터를 사용하면 클러스터가 권한 부여를 처리합니다.

요약 및 추가 고려사항

발행/구독 계층은 큐 관리자 간의 관계를 세밀히 제어합니다. 작성하고 나면 관리하기 위해 수동 개입이 거의 필요하지 않습니다. 하지만 시스템에 대한 일정한 제한조건도 있습니다.

- 계층에서 더 높은 노드일수록, 특히 루트 노드는 강력하고 가용성이 높으며 성능이 우수한 장비에서 호스트해야 합니다. 노드로 플로우할 것으로 예상되는 발행 트래픽이 더 많기 때문입니다.
- 계층에서 모든 비리프 큐 관리자의 가용성은 발행자로부터 다른 큐 관리자의 구독자에게 메시지를 플로우하는 네트워크의 기능에 영향을 줍니다.
- 기본적으로 구독하는 모든 토픽 문자열은 계층 전반에 전파되고 발행물은 연관된 토픽에 대한 구독이 있는 리모트 큐 관리자에만 전파됩니다. 따라서 구독 세트의 급격한 변경은 제한 요인이 될 수 있습니다. 이 기본 작동을 변경하고 그 대신 전체 큐 관리자에게 모든 발행물이 전파되도록 할 수 있으며 이 경우 프록시 구독이 필요하지 않습니다. 발행/구독 네트워크에서의 구독 성능을 참조하십시오.

  참고: 직접 라우트 클러스터에도 유사한 제한사항이 적용됩니다.
- 발행/구독 큐 관리자의 상호 연결된 속성으로 인해 프록시 구독이 네트워크의 모든 노드에 전파되는 데 시간이 걸립니다. 원격 발행물은 반드시 즉시 구독되는 것은 아니므로 새 토픽 문자열에 대한 구독 후에 이전 발행물이 송신되지 않을 수도 있습니다. 전체 큐 관리자에게 모든 발행물을 전파해서(이 경우 프록시 구독이 필요하지 않음) 구독 지연으로 인한 문제를 피할 수 있습니다. 발행/구독 네트워크에서의 구독 성능을 참조하십시오.

  참고: 직접 라우트 클러스터에도 이 제한사항이 적용됩니다.
- 발행/구독 계층에서 큐 관리자를 추가 또는 제거하려면 해당 큐 관리자의 위치 및 다른 큐 관리자에 대한 신뢰도를 주의깊게 고려하여, 계층에 대한 수동 구성을 수행해야 합니다. 아래에 추가 분기가 없는 계층의 맨 아래에 있는 큐 관리자를 추가 또는 제거하는 경우가 아니면 계층에 다른 큐 관리자도 구성해야 합니다.

발행/구독 계층을 라우팅 메커니즘으로 사용하기 전에 68 페이지의 『발행/구독 클러스터의 직접 라우팅』 및 72 페이지의 『발행/구독 클러스터의 토픽 호스트 라우팅』에 설명된 대체 접근법을 탐색하십시오.

분산 발행/구독 시스템 큐

네 개의 시스템 큐가 발행/구독 메시징을 위한 큐 관리자에 사용됩니다. 문제점 판별 및 용량 계획 용도로만 이 큐의 존재를 인식해야 합니다.

큐 모니터 방법에 대한 지침은 발행/구독 네트워크의 작성자 및 이용자 밸런스 조절을 참조하십시오.

표 6. 멀티플랫폼에서 시스템 큐 발행/구독	
시스템 큐	**목적**
SYSTEM.INTER.QMGR.CONTROL	IBM MQ 분산 발행/구독 제어 큐
SYSTEM.INTER.QMGR.FANREQ	IBM MQ 분산 발행/구독 내부 프록시 구독 팬 아웃 프로세스 입력 큐
SYSTEM.INTER.QMGR.PUBS	IBM MQ 분산 발행/구독 발행물
SYSTEM.HIERARCHY.STATE	IBM MQ 분산 발행/구독 계층 관계 상태

z/OS z/OS에서는 CSQINP2 초기화 입력 데이터 세트에 CSQ4INSX, CSQ4INSR 및 CSQ4INSG 샘플을 포함시켜 큐 관리자를 작성할 때 필요한 시스템 오브젝트를 설정합니다. 자세한 정보는 태스크 13: 초기화 입력 데이터 세트 사용자 정의를 참조하십시오.

발행/구독 시스템 큐의 속성은 96 페이지의 표 7에 표시되어 있습니다.

표 7. 발행/구독 시스템 큐의 속성	
속성	**기본값**
DEFPSIST	예
DEFSOPT	SHARED
MAXMSGL	Multi 멀티플랫폼의 경우: ALTER QMGR 명령의 MAXMSGL 매개변수 값 z/OS z/OS의 경우: 4194304 (즉, 4MB)
MAXDEPTH	999999999
SHARE	해당사항 없음
z/OS z/OS STGCLASS	이 속성은 z/OS 플랫폼에서만 사용됩니다.

참고: 애플리케이션이 넣은 메시지를 포함하는 유일한 큐는 SYSTEM.INTER.QMGR.PUBS입니다. **MAXDEPTH**은(는) 이 큐의 최대값으로 설정되어 중단 또는 초과 로드 중에 발행된 메시지를 임시로 축적할 수 있습니다. 큐 용량을 포함할 수 없는 시스템에서 큐 관리자가 실행 중인 경우 이 값을 조정해야 합니다.

관련 태스크

분산 발행/구독 문제점 해결

분배된 발행/구독 시스템 큐 오류

분산 발행/구독 큐 관리자 큐를 사용할 수 없으면 오류가 발생할 수 있습니다. 이는 발행/구독 네트워크에서 구독 정보를 전파하고 리모트 큐 관리자의 구독에 발행물을 전파하는 데 영향을 미칩니다.

팬아웃 요청 큐 SYSTEM.INTER.QMGR.FANREQ을(를) 사용할 수 없는 경우 구독을 작성하면 오류가 발생할 수 있으며 프록시 구독을 직접 연결된 큐 관리자에 전달해야 하는 경우 오류 메시지가 큐 관리자 오류 로그에 기록됩니다.

계층 관계 상태 큐 SYSTEM.HIERARCHY.STATE가 사용 불가능하면 큐 관리자 오류 로그에 오류 메시지가 기록되고 발행/구독 엔진은 COMPAT 모드가 됩니다. 발행/구독 모드를 보려면 DISPLAY QMGR PSMODE 명령을 사용하십시오.

다른 SYSTEM.INTER.QMGR 큐를 사용할 수 없는 경우 오류 메시지가 큐 관리자 오류 로그에 기록되고, 기능이 사용 안함으로 설정되지는 않지만 발행/구독 메시지가 이 큐 관리자 또는 원격 큐 관리자의 큐에 축적될 가능성이 높습니다.

상위, 하위 또는 발행/구독 클러스터 큐 관리자에 대한 필수 전송 큐나 발행/구독 시스템 큐가 사용 불가능한 경우 다음 결과가 발생합니다.

- 발행물이 전달되지 않고 발행 애플리케이션이 오류를 수신할 수 있습니다. 발행 애플리케이션이 오류를 수신하는 시기에 대한 자세한 내용은 **DEFINE TOPIC** 명령의 다음 매개변수를 참조하십시오. **PMSGDLV**, **NPMSGDLV**, **USEDLQ**.
- 수신된 큐 관리자 간 발행물이 입력 큐로 백아웃되고 이어서 재시도됩니다. 백아웃 임계값에 도달할 경우 전달되지 않은 발행물은 데드-레터 큐에 위치합니다. 문제점에 대한 자세한 내용은 큐 관리자 오류 로그에 있습니다.
- 전달되지 않은 프록시 구독이 팬아웃 요청 큐로 백아웃되고 이어서 다시 시도됩니다. 백아웃 임계값에 도달할 경우 전달되지 않은 프록시 구독은 연결된 큐 관리자에 전달되지 않고 데드-레터 큐에 위치합니다. 필요한 정정 관리 조치 세부사항을 포함하여, 문제점에 대한 자세한 내용은 큐 관리자 오류 로그에 있습니다.
- 계층 관계 프로토콜 메시지가 실패하고 연결 상태가 ERROR(으)로 플래그 지정됩니다. 연결 상태를 보려면 **DISPLAY PUBSUB** 명령을 사용하십시오.

관련 태스크

분산 발행/구독 문제점 해결

Multi 멀티플랫폼에서 스토리지 및 성능 요구사항 계획

IBM MQ 시스템에 맞는 실제적이고 도달 가능한 스토리지 및 성능 목표를 설정해야 합니다. 링크를 사용하여 플랫폼의 스토리지 및 성능에 영향을 미치는 요인을 알아보십시오.

요구사항은 IBM MQ를 사용 중인 시스템 및 사용하려는 컴포넌트에 따라 다릅니다.

지원되는 하드웨어 및 소프트웨어 환경에 대한 최신 정보는 IBM MQ 의 시스템 요구사항의 내용을 참조하십시오.

IBM MQ는 파일 시스템에 큐 관리자 데이터를 저장합니다. 다음 링크를 사용하여 IBM MQ에 사용할 디렉토리 구조의 계획 및 구성을 알아보십시오.

- 100 페이지의 『멀티플랫폼에서 파일 시스템 지원 계획』
- 100 페이지의 『멀티플랫폼에서 공유 파일 시스템에 대한 요구사항』
- 109 페이지의 『멀티플랫폼에서 IBM MQ 파일 공유』
- Linux AIX 111 페이지의 『AIX and Linux 시스템의 디렉토리 구조』
- Windows 120 페이지의 『Windows 시스템의 디렉토리 구조』
- IBM i 123 페이지의 『IBM i의 디렉토리 구조』

AIX and Linux에서의 시스템 자원, 공유 메모리, 프로세스 우선순위에 대한 정보를 보려면 다음 링크를 사용하십시오.

- Linux AIX 127 페이지의 『IBM MQ 및 UNIX System V IPC 자원』
- AIX 127 페이지의 『AIX의 공유 메모리』
- Linux AIX 127 페이지의 『IBM MQ 및 UNIX 프로세스 우선순위』

로그 파일에 대한 정보를 보려면 다음 링크를 사용하십시오.

- 126 페이지의 『멀티플랫폼에서 순환 또는 선형 로깅 선택』
- 로그의 크기 계산

관련 개념

128 페이지의 『Planning your IBM MQ environment on z/OS』

When planning your IBM MQ environment, you must consider the resource requirements for data sets, page sets, Db2, Coupling Facilities, and the need for logging, and backup facilities. Use this topic to plan the environment where IBM MQ runs.

관련 태스크

5 페이지의 『IBM MQ 아키텍처 계획』

IBM MQ 환경을 계획할 때, IBM MQ가 단일 및 다중 큐 관리자 아키텍처와, 포인트-투-포인트 및 발행/구독 메시징 스타일에 대해 제공하는 지원을 고려하십시오. 또한 자원 요구사항과, 로깅 및 백업 기능 사용을 계획하십시오.

관련 참조

AIX and Linux의 하드웨어 및 소프트웨어 요구사항

Windows의 하드웨어 및 소프트웨어 요구사항

Multi 멀티플랫폼에서 디스크 공간 요구사항

IBM MQ의 스토리지 요구사항은 설치하는 컴포넌트와 필요한 작업 공간의 양에 따라 달라집니다.

디스크 스토리지는 필요한 필수조건 컴포넌트를 포함하여 설치하도록 선택한 선택적 컴포넌트에 필수적입니다. 총 스토리지 요구사항은 사용하는 큐의 수, 큐에 있는 메시지의 수와 크기, 메시지가 지속적인지 여부에 따라 다릅니다. 자체 애플리케이션 프로그램에 대한 공간 외에 디스크, 테이프 또는 기타 매체의 아카이브 용량도 필요합니다.

다음 표에서는 여러 플랫폼에 다양한 제품 조합을 설치할 때 필요한 대략적인 디스크 공간 크기를 표시합니다. (값은 5MB 근사값으로 반올림되며 1MB는 1048576바이트입니다.)

- LTS 98 페이지의 『Long Term Support의 디스크 공간 요구사항』
- CD 99 페이지의 『Continuous Delivery의 디스크 공간 요구사항』

Long Term Support의 디스크 공간 요구사항

LTS V 9.4.0

표 *8. IBM MQ* 멀티플랫폼용 디스크 공간 요구사항 *Long Term Support*

플랫폼	클라이언트 설치 98 페이지의 『1』	서버 설치 98 페이지의 『2』	전체 설치 99 페이지의 『3』
AIX AIX	335MB	375MB	1810MB
IBM i IBM i(IBM i를 위한 추가 참고 참조)	485MB	845MB	1965 MB
Linux Linux for x86-64	270MB	295MB	2010MB
Linux Linux on Power® Systems - Little Endian	170MB	190MB	1400MB
Linux Linux for IBM Z®	255MB	290MB	1485MB
Windows Windows(64비트 설치) 99 페이지의 『4』	295MB	425MB	2310MB

참고:

1. 클라이언트 설치는 다음 컴포넌트를 포함합니다.
   - 런타임
   - 클라이언트
2. 서버 설치는 다음 컴포넌트를 포함합니다.

- 런타임
- 서버

3. 전체 설치는 사용 가능한 모든 컴포넌트를 포함합니다.
4. Windows 여기에 나열된 모든 컴포넌트가 Windows 시스템에 설치 가능한 기능이 아닙니다. 일부 기능은 때로 기타 기능에 포함됩니다. Windows 시스템의IBM MQ 기능을 참조하십시오.

IBM i를 위한 추가 참고: IBM i

1. IBM i에서는 서버에서 고유 클라이언트를 분리할 수 없습니다. 표의 서버 그림은 Java가 없는 5724H72*BASE용이며 영어 언어 로드(2924)가 포함되어 있습니다. 22개의 가능한 고유 언어 로드가 있습니다.
2. 표의 그림은 Java가 없는 고유 클라이언트 5725A49 *BASE용입니다.
3. Java 및 JMS 클래스를 서버와 클라이언트 바인딩 모두에 추가할 수 있습니다. 이러한 기능을 포함하려면 110MB를 추가하십시오.
4. 클라이언트나 서버에 샘플 소스를 추가하는 경우 10MB가 추가됩니다.
5. Java 및 JMS 클래스에 샘플을 추가하면 5MB가 더 추가됩니다.

Continuous Delivery의 디스크 공간 요구사항

CD V 9.4.0

표 *9. IBM MQ* 멀티플랫폼용 디스크 공간 요구사항 *Continuous Delivery*

플랫폼/CD 릴리스	클라이언트 설치 99 페이지의 『1』	서버 설치 100 페이지의 『2』	전체 설치 100 페이지의 『3』
AIX **AIX**			
V 9.4.0 IBM MQ 9.4.0	355MB	390MB	1440 MB
Linux **Linux for x86-64(64비트)**			
V 9.4.0 IBM MQ 9.4.0	280MB	295MB	1195MB
Linux **Linux on Power Systems - Little Endian**			
V 9.4.0 IBM MQ 9.4.0	170MB	195MB	1075MB
Linux **IBM Z의 경우 Linux**			
V 9.4.0 IBM MQ 9.4.0	260MB	290MB	1160MB
Windows **Windows(64비트 설치)** 100 페이지의 『4』			
V 9.4.0 IBM MQ 9.4.0	300MB	425MB	1785MB

참고:

1. 클라이언트 설치는 다음 컴포넌트를 포함합니다.
   - 런타임
   - 클라이언트

2. 서버 설치는 다음 컴포넌트를 포함합니다.
   - 런타임
   - 서버
3. 전체 설치는 사용 가능한 모든 컴포넌트를 포함합니다.
4. Windows 여기에 나열된 모든 컴포넌트가 Windows 시스템에 설치 가능한 기능이 아닙니다. 일부 기능은 때로 기타 기능에 포함됩니다. Windows 시스템의IBM MQ 기능을 참조하십시오.

관련 개념

IBM MQ 컴포넌트 및 기능

Multi 멀티플랫폼에서 파일 시스템 지원 계획

큐 관리자 데이터는 파일 시스템에 저장됩니다. 큐 관리자는 파일 시스템 잠금을 이용하여 다중 인스턴스 큐 관리자의 다중 인스턴스가 동시에 활성화되지 않게 합니다.

공유 파일 시스템

공유 파일 시스템을 통해 여러 시스템이 동일한 물리적 스토리지 디바이스에 동시에 액세스할 수 있습니다. 잠금 및 동시성 제어를 시행할 어떤 수단 없이 여러 시스템이 동일한 물리적 스토리지 디바이스에 직접 액세스한 경우에는 손상이 발생합니다. 운영 체제는 로컬 파일 시스템에 로컬 프로세스에 대한 잠금 및 동시성 제어를 제공하고 네트워크 파일 시스템은 분산 시스템에 대한 잠금 및 동시성 제어를 제공합니다.

네트워크 파일 시스템은 과거에 로깅 메시지의 요구사항을 충족시킬 정도로 충분히 빠르게 수행되지 않았거나 충분한 잠금 및 동시성 제어를 제공하지 않았습니다. 하지만 현재는, 네트워크 파일 시스템이 우수한 성능 및 신뢰할 수 있는 네트워크 파일 시스템 프로토콜(예: *RFC 3530, NFS(Network File System)* 버전 *4* 프로토콜)의 구현을 제공하고 로깅 메시지에 대한 요구사항을 확실히 충족시킬 수 있습니다.

공유 파일 시스템 및 IBM MQ

다중 인스턴스 큐 관리자의 큐 관리자 데이터는 공유 네트워크 파일 시스템에 저장됩니다. AIX, Linux, and Windows 시스템에서는 큐 관리자의 데이터 파일 및 로그 파일을 공유 네트워크 파일 시스템에 두어야 합니다.

IBM i IBM i에서 저널은 로그 파일 대신 사용되고 저널은 공유할 수 없습니다. IBM i의 멀티 인스턴스 큐 관리자는 저널 복제 또는 전환 가능한 저널을 사용하여 서로 다른 큐 관리자 인스턴스 사이에서 저널을 사용할 수 있도록 합니다.

IBM MQ 는 잠금을 사용하여 동일한 다중 인스턴스 큐 관리자의 다중 인스턴스가 동시에 활성 상태가 되지 않도록 합니다. 동일한 잠금을 통해 두 개의 개별 큐 관리자가 실수로 동일한 큐 관리자 데이터 파일 세트를 사용할 수 없게 하기도 합니다. 한 번에 큐 관리자의 한 인스턴스만 잠금을 보유할 수 있습니다. 결과적으로 IBM MQ가 공유 파일 시스템으로 액세스한 네트워크 스토리지에 저장된 큐 관리자 데이터를 지원합니다.

네트워크 파일 시스템의 모든 잠금 프로토콜이 견고하지는 않으며 파일 시스템이 데이터 무결성보다는 성능에 맞게 구성되었을 수 있으므로 **amqmfsck** 명령을 실행하여 네트워크 파일 시스템이 큐 관리자 데이터 및 로그에 대한 액세스를 제대로 제어하는지 테스트해야 합니다. 이 명령은 UNIX, Linux 및 IBM i 시스템에 적용됩니다. Windows에서는 지원되는 네트워크 파일 시스템이 하나만 있으므로 **amqmfsck** 명령이 필요하지 않습니다.

관련 태스크

102 페이지의 『멀티플랫폼에서 공유 파일 시스템 작동 확인』

amqmfsck를 실행하여 AIX, Linux 또는 IBM i의 공유 파일 시스템이 다중 인스턴스 큐 관리자의 큐 관리자 데이터를 저장하기 위한 요구사항을 충족하는지 확인하십시오. (Windows 구성의 유일한 요구사항은 공유 스토리지 프로비저닝을 위해 SMB 3을 사용하는 것입니다.)

Multi 멀티플랫폼에서 공유 파일 시스템에 대한 요구사항

공유 파일 시스템은 IBM MQ에 대한 신뢰할 수 있는 작업을 위해 데이터 쓰기 무결성, 보장된 독점 파일 액세스, 실패 시의 잠금 해제를 제공해야 합니다.

공유 파일 시스템이 충족해야 하는 요구사항

공유 파일 시스템이 IBM MQ와 함께 안정적으로 작동하기 위해 충족해야 하는 세 가지 기본 요구사항이 있습니다.

1. 데이터 쓰기 무결성

   데이터 쓰기 무결성을 비울 때 디스크에 쓰기라고도 합니다. 큐 관리자가 물리적 디바이스에 커미트되는 데이터와 동기화시킬 수 있어야 합니다. 트랜잭션 시스템에서는 다른 처리를 계속하기 전에 일부 쓰기가 안전하게 커미트되었는지 확인해야 합니다.

   보다 구체적으로 말하면 IBM MQ for AIX or Linux 플랫폼은 *O_SYNC* 열기 옵션과 `fsync()` 시스템 호출을 사용하여 명시적으로 복구 가능한 미디어에 강제로 기록하고 쓰기 조작은 올바르게 작동하는 이러한 옵션에 따라 달라집니다.

   주의: Linux 동기 쓰기 옵션을 여전히 지원하며 sync 옵션보다 더 나은 성능을 제공하는 async 옵션으로 파일 시스템을 마운트해야 합니다.

   하지만 파일 시스템을 Linux에서 내보낸 경우에는 여전히 sync 옵션을 사용하여 파일 시스템을 내보내야 함에 유의하십시오.

2. 보장된 독점 파일 액세스

   여러 큐 관리자를 동기화하려면 파일에 대한 배타적 잠금을 확보할 큐 관리자의 메커니즘이 있어야 합니다.

3. 실패 시 잠금 해제

   큐 관리자가 실패하는 경우나 파일 시스템과의 통신 장애가 있는 경우 큐 관리자가 잠근 파일을 잠금 해제해서 큐 관리자가 파일 시스템에 다시 연결할 때까지 대기하지 않고 기타 프로세스에 사용 가능하게 해야 합니다.

IBM MQ가 확실히 작동하려면 공유 파일 시스템이 이 요구사항을 충족시켜야 합니다. 그렇지 않으면 다중 인스턴스 큐 관리자 구성에 공유 파일 시스템을 사용할 때 큐 관리자 데이터와 로그가 손상됩니다.

Microsoft Windows의 다중 인스턴스 큐 관리자의 경우 Microsoft Windows 네트워크에서 사용하는 SMB (Server Message Block) 프로토콜이 네트워크 스토리지에 액세스해야 합니다. SMB (Server Message Block) 클라이언트는 Microsoft Windows이외의 플랫폼에서 시맨틱을 잠그기 위한 IBM MQ 요구사항을 충족하지 않으므로 Microsoft Windows 이외의 플랫폼에서 실행 중인 다중 인스턴스 큐 관리자는 SMB (Server Message Block) 를 공유 파일 시스템으로 사용하지 않아야 합니다.

기타 지원 플랫폼의 다중 인스턴스 큐 관리자의 경우에는 Posix를 준수하며 임대형 잠금을 지원하는 네트워크 파일 시스템 프로토콜을 통해 스토리지에 액세스해야 합니다. 네트워크 파일 시스템 4는 이 요구사항을 충족시킵니다. 실패 후에 잠금을 해제할 신뢰할 수 있는 메커니즘이 없는 네트워크 파일 시스템 버전 3과 같은 이전 파일 시스템에는 다중 인스턴스 큐 관리자를 사용해서는 안됩니다.

공유 파일 시스템이 요구사항을 충족하는지에 대한 확인

사용할 계획인 공유 파일 시스템이 이 요구사항을 충족시키는지 확인해야 합니다. 파일 시스템이 신뢰할 수 있게 제대로 구성되었는지도 확인해야 합니다. 공유 파일 시스템은 때로 신뢰성 대신 성능을 개선할 구성 옵션을 제공합니다.

자세한 정보는 IBM MQ 다중 인스턴스 큐 관리자 파일 시스템에 대한 테스트 명령문을 참조하십시오.

정상 상황에서는 IBM MQ가 속성 캐싱으로 제대로 작동하며 예를 들어, NFS 마운트에 NOAC를 설정해서 캐싱을 사용 불가능하게 할 필요가 없습니다. 여러 파일 시스템 클라이언트가 파일 시스템 서버의 동일한 파일에 대한 쓰기 액세스를 두고 경합할 때 각 클라이언트에 사용되는 캐싱된 속성이 서버의 속성과 같지 않을 수 있기 때문에 속성 캐싱은 문제를 야기할 수 있습니다. 이러한 방식으로 액세스하는 파일의 예로는 다중 인스턴스 큐 관리자에 대한 큐 관리자 오류 로그가 있습니다. 큐 관리자 오류 로그는 활성 및 대기 큐 관리자 인스턴스 모두에 기록될 수 있으며 캐싱된 파일 속성으로 인해 파일 롤오버가 발생하기 전에 오류 로그가 예상보다 더 커질 수 있습니다.

파일 시스템 검사에 도움을 주려면 공유 파일 시스템 작동 확인 태스크를 실행하십시오. 이 태스크는 공유 파일 시스템이 요구사항 2 및 3을 충족하는지 확인합니다. 공유 파일 시스템 문서에서 또는 디스크에 로그 데이터 기록을 시험해서 요구사항 1을 확인해야 합니다.

디스크 결함은 디스크에 쓸 때 IBM MQ가 FFDC(First Failure Data Capture) 오류로 보고하는 오류를 야기할 수 있습니다. 운영 체제의 파일 시스템 검사기를 실행하여 공유 파일 시스템의 디스크 결함을 확인할 수 있습니다. 예를 들면, 다음과 같습니다.

- Linux AIX AIX and Linux에서는 이 파일 시스템 검사기가 fsck입니다.
- Windows Windows 플랫폼에서는 파일 시스템 검사기가 CHKDSK 또는 SCANDISK입니다.

NFS 서버 보안

참고:

- IBM MQ 설치 디렉토리를 보유하는 데 사용되는 마운트 포인트에 대해 **nosuid** 또는 **noexec** 옵션을 사용할 수 없습니다. 이는 IBM MQ 에 setuid/setgid 실행 가능 프로그램이 포함되어 있고 이 프로그램이 제대로 실행되는 것을 막을 수 없기 때문입니다.
- NFS(Network File System) 서버에만 큐 관리자 데이터를 넣을 때 마운트 명령과 함께 다음 세 가지 옵션을 사용하여 큐 관리자의 실행에 해로운 영향을 주지 않고 시스템을 안전하게 만들 수 있습니다.

  noexec
  이 옵션을 사용하면 NFS에서 2진 파일의 실행을 중지하여 원격 사용자가 시스템에서 원하지 않는 코드를 실행하지 못하게 합니다.

  nosuid
  이 옵션을 사용하면 set-user-identifier 및 set-group-identifier 비트 사용을 방지하여 원격 사용자가 상위 권한을 얻지 못하게 합니다.

  nodev
  이 옵션을 사용하면 문자 및 블록 특수 디바이스의 사용 또는 정의를 중지하여 원격 사용자가 chroot에서 벗어나지 않게 합니다.

IBM i Linux AIX 멀티플랫폼에서 공유 파일 시스템 작동 확인

amqmfsck를 실행하여 AIX, Linux 또는 IBM i의 공유 파일 시스템이 다중 인스턴스 큐 관리자의 큐 관리자 데이터를 저장하기 위한 요구사항을 충족하는지 확인하십시오. (Windows 구성의 유일한 요구사항은 공유 스토리지 프로비저닝을 위해 SMB 3을 사용하는 것입니다.)

시작하기 전에

네트워크 스토리지가 있는 서버 하나와 이 서버에 연결되어 있고 IBM MQ가 설치된 다른 두 개의 서버가 필요합니다. 파일 시스템을 구성하려면 관리자 (루트) 권한이 있어야 하며 **amqmfsck**를 실행하려면 IBM MQ 관리자여야 합니다.

이 태스크 정보

100 페이지의 『멀티플랫폼에서 공유 파일 시스템에 대한 요구사항』 에 다중 인스턴스 큐 관리자에 공유 파일 시스템 사용을 위한 파일 시스템 요구사항이 설명되어 있습니다. IBM MQ 기술 노트 IBM MQ 다중 인스턴스 큐 관리자 파일 시스템에 대한 테스트 명령문은 IBM가 이미 테스트된 공유 파일 시스템을 나열합니다. 이 태스크의 프로시저는 나열되지 않은 파일 시스템이 데이터 무결성을 유지하는지 평가하는 데 도움이 되는 파일 시스템 테스트 방법을 설명합니다.

다중 인스턴스 큐 관리자의 장애 복구는 큐 관리자의 데이터 또는 로그 파일 쓰기를 차단하는 네트워킹 문제점을 포함하여, 하드웨어 또는 소프트웨어 장애를 통해 트리거될 수 있습니다. 사용자는 주로 파일 서버의 장애 요인에 관심이 있습니다. 하지만 잠금이 제대로 해제되는지 테스트하려면 IBM MQ 서버의 실패도 야기해야 합니다. 공유 파일 시스템에서 확신을 가지려면 다음의 장애와 환경 고유의 다른 장애를 모두 테스트하십시오.

1. 디스크 동기화를 포함하여 파일 서버의 운영 체제 종료.
2. 디스크 동기화를 포함하지 않은 파일 서버의 운영 체제 정지.
3. 각 서버에서 재설정 단추 누르기.
4. 각 서버의 네트워크 케이블 분리.
5. 각 서버의 전원 케이블 분리.
6. 각 서버 끄기.

큐 관리자 데이터와 로그를 공유하기 위해 사용할 네트워크 스토리지에 디렉토리를 작성하십시오. 디렉토리 소유자는 IBM MQ 관리자, 즉 AIX and Linux의 mqm 그룹 멤버여야 합니다. 테스트를 실행하는 사용자에게는 IBM MQ 관리자 권한이 있어야 합니다.

Linux 에서 다중 인스턴스 큐 관리자 작성 또는 IBM i 에서 저널 미러링 및 NetServer 를 사용하여 다중 인스턴스 큐 관리자 작성 에서 파일 시스템을 내보내고 마운트하는 예를 사용하면 파일 시스템을 구성하는 데 도움이 됩니다. 여러 다른 파일 시스템에 각기 다른 구성 단계가 필요합니다. 파일 시스템 문서를 읽으십시오.

참고: IBM MQ MQI client 샘플 프로그램 **amqsfhac**를 **amqmfsck**와 병렬로 실행하여 큐 관리자가 실패 중에 메시지 무결성을 유지하는지 증명합니다.

프로시저

각 검사에서 파일 시스템 검사기가 실행되는 동안 이전 목록의 모든 장애를 유발하십시오. **amqsfhac**를 **amqmfsck**와 동시에 실행하려면 107 페이지의 『메시지 무결성을 테스트하기 위해 amqsfhac 실행』 태스크를 이 태스크와 병렬로 수행하십시오.

1. 두 개의 IBM MQ 서버에 내보낸 디렉토리를 마운트하십시오.

   파일 시스템 서버에서 공유 디렉토리 shared 및 다중 인스턴스 큐 관리자에 대한 데이터를 저장할 서브디렉토리 qmdata를 작성하십시오. Linux에서 다중 인스턴스 큐 관리자의 공유 디렉토리를 설정하는 예는 Linux 에서 다중 인스턴스 큐 관리자 작성 을 참조하십시오.
2. 기본 파일 시스템 작동을 검사하십시오.
   하나의 IBM MQ 서버에서 매개변수 없이, 파일 시스템 검사기를 실행하십시오.

   IBM MQ 서버 1:

   ```
   amqmfsck /shared/qmdata
   ```
3. 두 개의 IBM MQ 서버 모두에서 동일한 디렉토리에 동시 쓰기를 검사하십시오.
   두 개의 IBM MQ 서버 모두에서 -c 옵션과 함께 파일 시스템 검사기를 동시에 실행하십시오.

   IBM MQ 서버 1:

   ```
   amqmfsck -c /shared/qmdata
   ```

   IBM MQ 서버 2:

   ```
   amqmfsck -c /shared/qmdata
   ```
4. 두 개의 IBM MQ 서버 모두에서 잠금 대기 및 해제를 검사하십시오.
   두 개의 IBM MQ 서버 모두에서 -w 옵션과 함께 파일 시스템 검사기를 동시에 실행하십시오.

   IBM MQ 서버 1:

   ```
   amqmfsck -w /shared/qmdata
   ```

   IBM MQ 서버 2:

   ```
   amqmfsck -w /shared/qmdata
   ```
5. 데이터 무결성을 확인하십시오.

   a) 테스트 파일을 포맷하십시오.

   테스트 중인 디렉토리에 큰 파일을 작성하십시오. 후속 단계가 정상적으로 완료될 수 있도록 파일이 포맷됩니다. 장애 복구를 시뮬레이트하기 위해 두 번째 단계를 인터럽트할 시간이 충분하도록 파일이 충분히 커야 합니다. 기본값 262144페이지(1GB)를 시도해 보십시오. 느린 파일 시스템에서는 포맷이 약 60초 안에 완료되도록 프로그램이 자동으로 이 기본값을 줄입니다.

IBM MQ 서버 1:

```
amqmfsck -f /shared/qmdata
```

서버는 다음 메시지로 응답합니다.

```
Formatting test file for data integrity test.

Test file formatted with 262144 pages of data.
```

b) 장애를 유발시키는 동안 파일 시스템 검사기를 사용하여 테스트 파일에 데이터를 쓰십시오.

두 개 서버에서 동시에 테스트 프로그램을 실행하십시오. 실패가 발생할 서버에서 테스트 프로그램을 시작한 후 실패에서 생존할 서버에서 테스트 프로그램을 시작하십시오. 조사하고 있는 장애를 유발하십시오.

첫 번째 테스트 프로그램이 오류 메시지로 중지됩니다. 두 번째 테스트 프로그램은 테스트 파일에 대한 잠금을 확보하고 첫 번째 테스트 프로그램이 중단한 위치에서 시작하여 테스트 파일에 데이터를 씁니다. 두 번째 테스트 프로그램이 완료할 때까지 실행되게 하십시오.

표 *10.* 두 개의 서버에서 동시에 데이터 무결성 검사기 실행

IBM MQ 서버 1	**IBM MQ 서버 2**
`amqmfsck -a /shared/qmdata`	

`Please start this program on a second machine`
`with the same parameters.`
 	
`File lock acquired.`
 	
`Start a second copy of this program with the same parameters on another server.`
 	
`Writing data into test file.`
 	
`To increase the effectiveness of the test, interrupt the writing by ending the process,`
`temporarily breaking the network connection`
`to the networked storage,`
`rebooting the server or turning off the power.`
	`amqmfsck -a /shared/qmdata`
	
	`Waiting for lock...`
	
	`Waiting for lock...`
	
	`Waiting for lock...`
	
	`Waiting for lock...`
	
	`Waiting for lock...`
	
	`Waiting for lock...`

Turn the power off here.	

표 *10.* 두 개의 서버에서 동시에 데이터 무결성 검사기 실행 *(계속)*

IBM MQ 서버 1	**IBM MQ 서버 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.`

테스트 타이밍은 파일 시스템의 작동에 따라 다릅니다. 예를 들어, 첫 번째 프로그램이 가동 중단 후에 확보한 파일 잠금을 파일 시스템이 해제하려면 일반적으로 30 - 90초가 걸립니다. 첫 번째 테스트 프로그램이 파일을 채우기 전에 실패를 유발할 시간이 없으면 **amqmfsck**의 -x 옵션을 사용하여 테스트 파일을 삭제하십시오. 더 큰 테스트 파일로 처음부터 테스트를 시도하십시오.

c) 테스트 파일에서 데이터의 무결성을 확인하십시오.

IBM MQ 서버 2:

```
amqmfsck -i /shared/qmdata
```

서버는 다음 메시지로 응답합니다.

```
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. 테스트 파일을 삭제하십시오.

IBM MQ 서버 2:

```
amqmfsck -x /shared/qmdata
Test files deleted.
```

서버는 다음 메시지로 응답합니다.

```
Test files deleted.
```

결과

이 프로그램은 테스트가 정상적으로 완료되면 엑시트 코드 0을 리턴하고 그렇지 않을 경우 0 이외의 값을 리턴합니다.

예

세 가지 예의 첫 번째 세트는 최소 출력을 생성하는 명령을 보여줍니다.

한 서버의 기본 파일 잠금 테스트 성공

```
> amqmfsck /shared/qmdata
The tests on the directory completed successfully.
```

한 서버의 기본 파일 잠금 테스트 실패

```
> amqmfsck /shared/qmdata
AMQ6245: Error Calling 'write()[2]' on file '/shared/qmdata/amqmfsck.lck' error '2'.
```

두 개의 서버에서 잠금 테스트 성공

표 *11.* 두 개의 서버에서 잠금 성공

IBM MQ 서버 1	**IBM MQ 서버 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`

세 가지 예의 두 번째 세트는 상세 모드를 사용하여 동일한 명령을 표시합니다.

한 서버의 기본 파일 잠금 테스트 성공

```
> 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.
```

한 서버의 기본 파일 잠금 테스트 실패

```
> 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).
```

두 개의 서버에서 잠금 테스트 성공

표 *12.* 두 개의 서버에서 잠금 성공 - 상세 모드

IBM MQ 서버 1	**IBM MQ 서버 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`

관련 참조

고가용성 샘플 프로그램

Multi 메시지 무결성을 테스트하기 위해 *amqsfhac* 실행

IBM MQ MQI client 샘플 프로그램 **amqsfhac**를 **amqmfsck**와 병렬로 실행하여 큐 관리자가 실패 중에 메시지 무결성을 유지하는지 증명합니다.

시작하기 전에

이 테스트를 위해 네 개의 서버가 필요합니다. 다중 인스턴스 큐 관리자에 두 개의 서버, 파일 시스템에 하나의 서버, 그리고 **amqsfhac**를 IBM MQ MQI client 애플리케이션으로 실행하는 데 하나의 서버가 필요합니다.

102 페이지의 『멀티플랫폼에서 공유 파일 시스템 작동 확인』 에 있는 단계 103 페이지의 『1』 에 따라 다중 인스턴스 큐 관리자에 대한 파일 시스템을 설정하십시오.

이 태스크 정보

IBM MQ MQI client 샘플 프로그램 **amqsfhac**는 네트워크 스토리지를 사용하는 큐 관리자가 실패에 따른 데이터 무결성을 유지보수하는지 확인합니다. **amqsfhac** 를 **amqmfsck** 와 병렬로 실행하여 실패 중에 큐 관리자가 메시지 무결성을 유지보수하는지 확인하십시오.

프로시저

1. 프로시저의 103 페이지의 『1』 단계에서 작성한 파일 시스템을 사용하여 다른 서버 QM1에 다중 인스턴스 큐 관리자를 작성하십시오.
   다중 인스턴스 큐 관리자 작성을 참조하십시오.
2. 큐 관리자의 가용성을 높게 하여 두 서버 모두에서 큐 관리자를 시작하십시오.

   서버 1에서:

   ```
   strmqm -x QM1
   ```

   서버 2에서:

   ```
   strmqm -x QM1
   ```

3. **amqsfhac**를 실행할 클라이언트 연결을 설정하십시오.
   a) *IBM MQ* 설치 확인의 프로시저 또는 다시 연결 가능한 클라이언트 샘플의 예제 스크립트를 사용하여 클라이언트 연결을 설정하십시오.
   b) 클라이언트 채널을 IP 주소가 두 개가 되도록 수정하십시오. 이 주소는 QM1을 실행하는 두 개의 서버에 해당합니다.
   예 스크립트에서 다음을

   ```
   DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
   CONNAME('LOCALHOST(2345)') QMNAME(QM1) REPLACE
   ```

   이를 다음 값으로 변경하십시오.

   ```
   DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
   CONNAME('server1(2345),server2(2345)') QMNAME(QM1) REPLACE
   ```

   여기서, server1 및 server2는 두 개 서버의 호스트 이름이고 2345는 채널 리스너가 대기 중인 포트입니다. 일반적으로 기본값은 1414입니다. 1414를 기본 리스너 구성에 사용할 수 있습니다.
4. 테스트를 위해 QM1에 두 개의 로컬 큐를 작성하십시오.
   다음 MQSC 스크립트를 실행하십시오.

   ```
   DEFINE QLOCAL(TARGETQ) REPLACE
   DEFINE QLOCAL(SIDEQ) REPLACE
   ```

5. **amqsfhac**로 구성을 테스트하십시오.

   ```
   amqsfhac QM1 TARGETQ SIDEQ 2 2 2
   ```

6. 파일 시스템의 무결성을 테스트하는 동안 메시지 무결성을 테스트하십시오.

   102 페이지의 『멀티플랫폼에서 공유 파일 시스템 작동 확인』의 103 페이지의 『5』 단계 중에 **amqsfhac** 를 실행하십시오.

   ```
   amqsfhac QM1 TARGETQ SIDEQ 10 20 0
   ```

활성 큐 관리자 인스턴스를 중지하면 **amqsfhac**가 활성화된 후 다른 큐 관리자 인스턴스에 다시 연결합니다. 다음 테스트에서 실패를 되돌릴 수 있도록 중지된 큐 관리자 인스턴스를 다시 시작하십시오. 장애 복구가 발생하기에 충분한 시간 동안 테스트 프로그램이 실행하도록 환경에 대한 시험을 기반으로 반복 수를 늘려야 합니다.

결과

108 페이지의 『6』 단계의 **amqsfhac** 실행 예가 다음 예에 표시됩니다. 이 예에서, 테스트는 성공입니다.

```
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
```

테스트에서 문제가 감지된 경우 출력에 실패가 보고됩니다. 일부 테스트 실행 MQRC_CALL_INTERRUPTED에서 "Resolving to backed out"을 보고할 수 있습니다. 결과는 차이가 없습니다. 실패가 발생하기 이전 또는 이후에 네트워크 파일 스토리지에서 디스크 쓰기가 커미트되었는지 여부에 따라 결과가 다릅니다.

관련 참조

amqmfsck(파일 시스템 검사)

고가용성 샘플 프로그램

Multi 멀티플랫폼에서 IBM MQ 파일 공유

일부 IBM MQ 파일은 활성 큐 관리자가 독점 액세스하고 다른 파일은 공유됩니다.

IBM MQ 파일은 프로그램 파일과 데이터 파일로 나뉩니다. 프로그램 파일은 일반적으로 IBM MQ를 실행하는 각 서버에 로컬로 설치됩니다. 큐 관리자는 기본 데이터 디렉토리의 데이터 파일과 디렉토리에 대한 액세스를 공유합니다. 110 페이지의 그림 32에 표시된 각 qmgrs 및 log 디렉토리에 포함된 자체 큐 관리자 디렉토리 트리에 대한 독점 액세스를 요구합니다.

110 페이지의 그림 32은 IBM MQ 디렉토리 구조의 상위 레벨 보기입니다. 큐 관리자 간에 공유 가능하며 원격으로 할 수 있는 디렉토리를 표시합니다. 플랫폼별로 세부사항이 다릅니다. 점선은 구성 가능한 경로를 표시합니다.

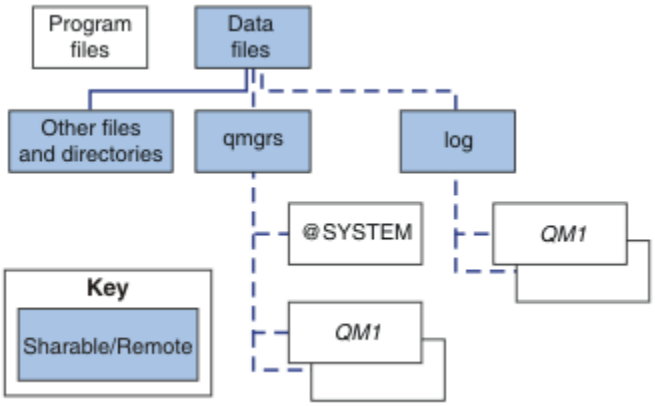


그림 *32. IBM MQ* 디렉토리 구조의 전체 보기

프로그램 파일

프로그램 파일 디렉토리는 일반적으로 기본 위치이고 로컬이며 서버의 모든 큐 관리자에 공유됩니다.

데이터 파일

일반적으로 데이터 파일 디렉토리는 AIX and Linux에서는 기본 위치인 `/var/mqm`에 로컬이며 Windows에서는 설치 시 구성 가능합니다. 큐 관리자 간에 공유됩니다. 기본 위치를 원격으로 할 수 있지만 IBM MQ의 여러 다른 설치에서 기본 위치를 공유하지는 않습니다. IBM MQ 구성의 `DefaultPrefix` 속성은 이 경로를 가리킵니다.

qmgrs

큐 관리자 데이터의 위치를 지정하는 두 가지 대체 방법이 있습니다.

Prefix 속성 사용

Prefix 속성은 `qmgrs` 디렉토리의 위치를 지정합니다. IBM MQ는 큐 관리자 이름에서 큐 관리자 디렉토리 이름을 구성하고 이 이름을 `qmgrs` 디렉토리의 서브디렉토리로 작성합니다.

Prefix 속성은 `mqs.ini` 파일의 QueueManager 스탠자 에 있으며 모든 큐 관리자 스탠자의 **DefaultPrefix** 속성 값에서 상속됩니다. 기본적으로 간편한 관리를 위해 큐 관리자는 일반적으로 동일한 `qmgrs` 디렉토리를 공유합니다.

큐 관리자의 `qmgrs` 디렉토리 위치를 변경할 경우 **Prefix** 속성의 값을 변경해야 합니다.

AIX and Linux 플랫폼의 경우 110 페이지의 그림 32 에 있는 `QM1` 디렉토리의 **Prefix** 속성은 다음과 같습니다.

```
Prefix=/var/mqm
```

DataPath 속성 사용

DataPath 속성은 큐 관리자 데이터 디렉토리의 위치를 지정합니다.

DataPath 속성은 큐 관리자 데이터 디렉토리의 이름을 포함하여 전체 경로를 지정합니다. **DataPath** 속성은 큐 관리자 데이터 디렉토리에 대한 불완전한 경로를 지정하는 **Prefix** 속성과 다릅니다.

DataPath 속성 (지정된 경우) 은 `mqs.ini` 파일의 QueueManager 스탠자 에 있습니다. 지정된 경우에는 **Prefix** 속성의 값보다 우선합니다.

큐 관리자의 큐 관리자 데이터 디렉토리 위치를 변경할 경우 `DataPath` 속성의 값을 변경해야 합니다.

Linux 또는 AIX 플랫폼의 경우 110 페이지의 그림 32의 `QM1` 디렉토리에 대한 `DataPath` 속성은 다음과 같습니다.

```
DataPath=/var/mqm/qmgrs/QM1
```

log

로그 디렉토리는 큐 관리자 구성의 로그 스탠자 에서 각 큐 관리자에 대해 별도로 지정됩니다. 큐 관리자 구성은 qm.ini에 있습니다.

***DataPath*/*QmgrName*/@IPCC 서브디렉토리**

DataPath/*QmgrName*/@IPCC 서브디렉토리는 공유 디렉토리 경로에 있습니다. IPC 파일 시스템 오브젝트에 대한 디렉토리 경로를 구성하는 데 사용됩니다. 시스템에서 큐 관리자가 공유될 때 큐 관리자의 네임스페이스를 구별해야 합니다.

IPC 파일 시스템 오브젝트는 시스템에 의해 구별되어야 합니다. 큐 관리자가 실행하는 각 시스템에 대한 서브디렉토리가 디렉토리 경로에 추가됩니다(111 페이지의 그림 33 참조).

```
DataPath/QmgrName/@IPCC/esem/myHostName/
```

그림 *33.* 예제 *IPC* 서브디렉토리

*myHostName*은 운영 체제에서 리턴된 호스트 이름의 처음 20자까지입니다. 일부 시스템에서는 자르기 전의 호스트 이름 길이가 최대 64자일 수 있습니다. *myHostName*의 생성된 값은 다음과 같은 두 가지 이유로 문제점을 발생시킬 수 있습니다.

1. 처음 20자가 고유하지 않습니다.
2. 시스템에 항상 동일한 호스트 이름을 할당하지는 않는 DHCP 알고리즘을 통해 호스트 이름이 생성됩니다.

이 경우 환경 변수 **MQS_IPC_HOST**를 사용하여 *myHostName* 를 설정하십시오. 111 페이지의 그림 34의 내용을 참조하십시오.

```
export MQS_IPC_HOST= myHostName
```

그림 *34.* 예*: **MQS_IPC_HOST*** 설정

기타 파일 및 디렉토리

추적 파일을 포함한 디렉토리와 같은 기타 파일 및 디렉토리와 공통 오류 로그는 일반적으로 공유되며 로컬 파일 시스템에 보관됩니다.

공유 파일 시스템의 지원을 통해 IBM MQ 는 파일 시스템 잠금을 사용하여 이러한 파일에 대한 독점 액세스를 관리합니다. 파일 시스템 잠금을 사용하면 한 번에 특정 큐 관리자의 한 인스턴스만 활성화할 수 있습니다.

특정 큐 관리자의 첫 번째 인스턴스를 시작하면 이 큐 관리자가 큐 관리자 디렉토리의 소유권을 가져갑니다. 두 번째 인스턴스를 시작하는 경우 첫 번째 인스턴스가 중지되어 있으면 두 번째가 소유권을 가져갈 수 있습니다. 첫 번째 큐 관리자가 여전히 실행 중이면 두 번째 인스턴스는 시작에 실패하고 큐 관리자가 실행 중이라고 보고합니다. 첫 번째 큐 관리자가 중지된 경우에는 두 번째 큐 관리자가 큐 관리자 파일의 소유권한을 가져가고 실행 큐 관리자가 됩니다.

두 번째 큐 관리자가 첫 번째 큐 관리자로부터 인계받는 프로시저를 자동화할 수 있습니다. 다른 큐 관리자의 인계를 허용하는 strmqm -x 옵션으로 첫 번째 큐 관리자를 시작하십시오. 그러면 두 번째 큐 관리자는 큐 관리자 파일의 소유권 인계를 시도하기 전에 큐 관리자 파일이 잠금 해제될 때까지 대기한 후에 시작합니다.

Linux AIX AIX and Linux 시스템의 디렉토리 구조

AIX and Linux 시스템의 IBM MQ 디렉토리 구조는 보다 쉽게 관리하고 성능을 향상시키며 신뢰성을 향상시키기 위해 다른 파일 시스템에 맵핑될 수 있습니다.

공유 파일 시스템을 사용하여 다중 인스턴스 큐 관리자를 실행하려면 IBM MQ의 유연한 디렉토리 구조를 사용하십시오.

crtmqm *QM1* 명령을 사용하여 112 페이지의 그림 35 에 표시된 디렉토리 구조를 작성하십시오. 여기서 R 는 제품의 릴리스입니다. 이는 IBM MQ 시스템에서 작성된 큐 관리자의 일반적인 디렉토리 구조입니다. 일부 디렉토

리, 파일, .ini 속성 설정은 명확한 표시를 위해 생략되며 다른 큐 관리자 이름은 맹글링을 통해 변경될 수 있습니다. 파일 시스템의 이름은 시스템에 따라 다릅니다.

일반 설치에서는 작성한 모든 큐 관리자가 로컬 파일 시스템의 공용 `log` 및 `qmgrs` 디렉토리를 가리킵니다. 다중 인스턴스 구성에서는 `log` 및 `qmgrs` 디렉토리가 IBM MQ의 다른 설치와 공유하는 네트워크 파일 시스템에 있습니다.

112 페이지의 그림 35 에서는 IBM MQ v7.R on AIX 여기서 R 는 제품의 릴리스입니다. 대체 다중 인스턴스 구성의 예는 116 페이지의 『AIX and Linux 시스템의 예 디렉토리 구성』 의 내용을 참조하십시오.

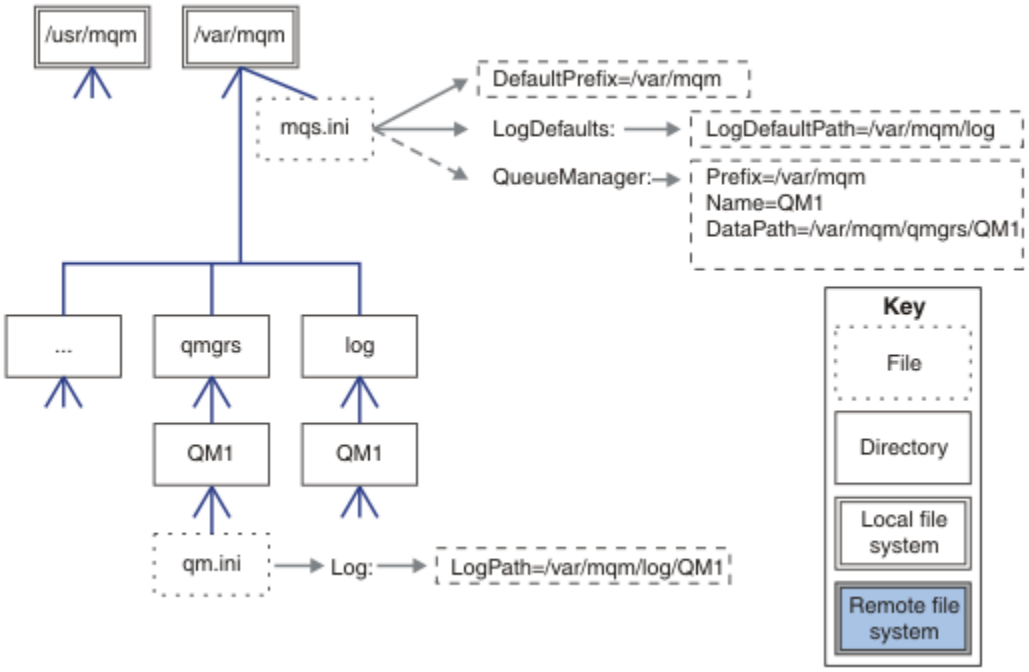


그림 *35. AIX and Linux* 시스템의 기본 *IBM MQ* 디렉토리 구조 예제

이 제품은 기본적으로 AIX에서는 `/usr/mqm`에 설치되고 기타 시스템에서는 `/opt/mqm`에 설치됩니다. 작업 디렉토리는 `/var/mqm` 디렉토리에 설치됩니다.

참고: IBM MQ를 설치하기 전에 `/var/mqm` 파일 시스템을 작성한 경우에는 mqm 사용자에게 전체 디렉토리 권한(예: 파일 모드 755)이 있는지 확인하십시오.

참고: 큐 관리자가 생성한 FFDC가 `/var/mqm`을 포함한 파일 시스템을 채우지 못하게 하려면 `/var/mqm/errors` 디렉토리가 개별 파일 시스템이어야 합니다.

자세한 정보는 AIX and Linux 시스템에 파일 시스템 작성을 참조하십시오.

`log` 및 `qmgrs` 디렉토리는 `mqs.ini` 파일에서 `LogDefaultPath` 및 `DefaultPrefix` 속성의 기본값에 정의된 해당 기본 위치에 표시됩니다. 큐 관리자가 작성될 때 기본적으로 큐 관리자 데이터 디렉토리는 *`DefaultPrefix`*`/qmgrs`에 작성되고 로그 파일 디렉토리는 *`LogDefaultPath`*`/log`에 작성됩니다. `LogDefaultPath` 및 `DefaultPrefix`는 기본적으로 큐 관리자 및 로그 파일이 작성되는 위치에만 영향을 줍니다. 큐 관리자 디렉토리의 실제 위치는 `mqs.ini` 파일에 저장되고 로그 파일 디렉토리의 위치는 `qm.ini` 파일에 저장됩니다.

큐 관리자의 로그 파일 디렉토리는 `qm.ini` 파일의 `LogPath` 속성에 정의됩니다. **`crtmqm`** 명령에 `-ld` 옵션을 사용하여 큐 관리자에 대한 `LogPath` 속성을 설정하십시오 (예: **`crtmqm`** `-ld` *`LogPath`* `QM1`). `ld` 매개변수를 생략하면 `LogDefaultPath` 값이 대신 사용됩니다.

큐 관리자 데이터 디렉토리는 `mqs.ini` 파일에 있는 `QueueManager` 스탠자의 `DataPath` 속성에 정의됩니다. **`crtmqm`** 명령에 `-md` 옵션을 사용하여 큐 관리자에 대한 `DataPath` 를 설정하십시오 (예: **`crtmqm`** `-md` *`DataPath`* `QM1`). `md` 매개변수를 생략하면 `DefaultPrefix` 또는 `Prefix` 속성 값이 대신 사용됩니다. `Prefix`가 `DefaultPrefix`보다 우선합니다.

보통은, 로그와 데이터 디렉토리를 모두 단일 명령으로 지정하여 QM1을 작성하십시오.

```
 crtmqm
-md DataPath -ld
LogPath QM1
```

큐 관리자가 중지될 때 qm.ini 파일의 DataPath 및 LogPath 속성을 편집하여 기존 큐 관리자의 큐 관리자 로그 및 데이터 디렉토리 위치를 수정할 수 있습니다.

errors 디렉토리의 경로는 /var/mqm의 다른 모든 디렉토리의 경로와 마찬가지로 수정이 불가능합니다. 하지만 이 디렉토리를 다른 파일 시스템에 마운트하거나 다른 디렉토리에 기호 링크할 수 있습니다.

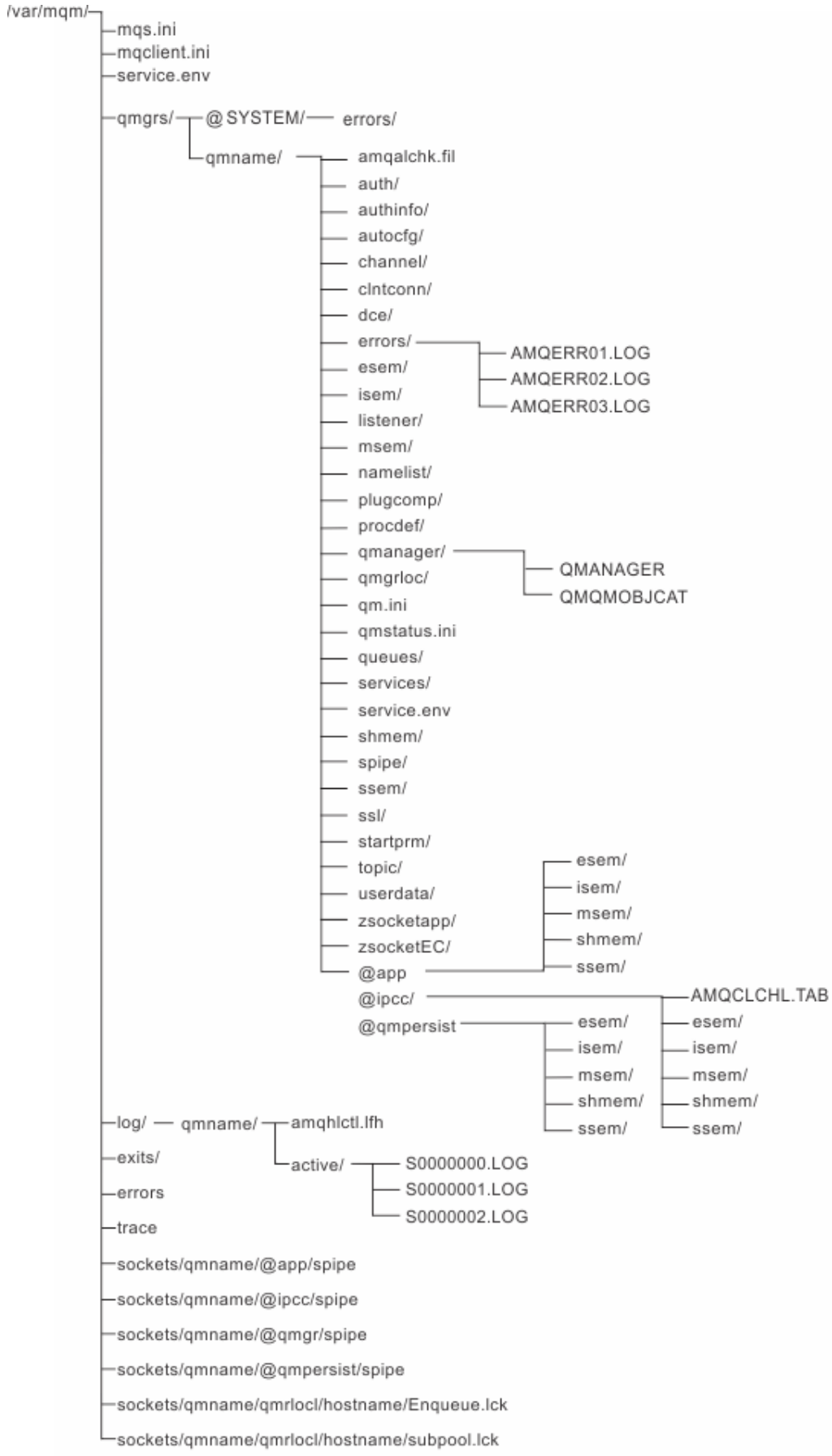
Linux AIX *AIX and Linux* 시스템의 디렉토리 컨텐츠

큐 관리자와 연관된 디렉토리의 컨텐츠

제품 파일 위치에 대한 자세한 정보는 설치 위치 선택을 참조하십시오.

대체 디렉토리 구성에 대한 정보는 100 페이지의 『멀티플랫폼에서 파일 시스템 지원 계획』의 내용을 참조하십시오.

다음 디렉토리 구조는 큐 관리자가 일정 시간 동안 사용된 후 IBM MQ 를 나타냅니다. 실제 구조는 큐 관리자에 발생한 조작에 따라 다릅니다.


/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 디렉토리에는 개별 큐 관리자가 아닌 IBM MQ 설치에 전체적으로 적용되는 구성 파일 및 출력 디렉토리가 포함되어 있습니다.

표 *13. AIX and Linux*에서 */var/mqm* 디렉토리의 문서화된 컨텐츠

디렉토리 또는 파일 이름	컨텐츠
mqs.ini	큐 관리자가 시작될 때 읽는 IBM MQ 설치 전반의 구성 파일. **AMQ_MQS_INI_LOCATION** 환경 변수를 사용하여 수정 가능한 파일 경로. **strmqm** 명령이 실행되는 쉘에서 설정되고 내보내는지 확인하십시오.
mqclient.ini	IBM MQ MQI client 프로그램이 읽는 기본 클라이언트 구성 파일. **MQCLNTCF** 환경 변수를 사용하여 수정 가능한 파일 경로.
service.env	서비스 프로세스에 대한 시스템 범위 환경 변수를 포함합니다. 수정된 파일 경로.
errors/	시스템 범위 오류 로그 및 FFST 파일. 수정된 디렉토리 경로. FFST: IBM MQ for UNIX 및 Linux 시스템도 참조하십시오.
sockets/	시스템 용도로만 각 큐 관리자에 대한 정보를 포함합니다.
trace/	추적 파일. 수정된 디렉토리 경로.
웹/	mqweb 서버 디렉토리
exits/	사용자 채널 엑시트 프로그램을 포함한 기본 디렉토리. mqs.ini 파일의 ApiExit 스탠자에서 수정 가능한 위치.
exits64/	

/var/mqm/qmgrs/*qmname*/

/var/mqm/qmgrs/*qmname*/에는 큐 관리자에 대한 디렉토리 및 파일이 포함되어 있습니다. 활성 큐 관리자 인스턴스의 독점 액세스를 위해 디렉토리가 잠깁니다. 디렉토리 경로는 mqs.ini 파일에서 또는 **crtmqm** 명령의 **md** 옵션을 사용하여 직접 수정할 수 있습니다.

표 *14. AIX and Linux*에서 */var/mqm/qmgrs/qmname* 디렉토리의 문서화된 컨텐츠

디렉토리 또는 파일 이름	컨텐츠
qm.ini	큐 관리자가 시작될 때 읽는 큐 관리자 구성 파일.
errors/	큐 관리자 범위 오류 로그. *qmname* = @system은 알 수 없거나 사용 불가능한 큐 관리자에 대한 채널 관련 메시지를 포함합니다.
@ipcc/AMQCLCHL.TAB	IBM MQ 서버가 작성하고 IBM MQ MQI client 프로그램이 읽는 기본 클라이언트 채널 제어 테이블. **MQCHLLIB** 및 **MQCHLTAB** 환경 변수를 사용하여 수정 가능한 파일 경로.

표 *14. AIX and Linux*에서 */var/mqm/qmgrs/qmname* 디렉토리의 문서화된 컨텐츠 *(계속)*	
디렉토리 또는 파일 이름	**컨텐츠**
qmanager	큐 관리자 오브젝트 파일: QMANAGER 오브젝트 관리자 오브젝트 카탈로그: QMQMOBJCAT
authinfo/	큐 관리자 내에 정의된 각 오브젝트는 이 디렉토리의 파일과 연관됩니다. 파일 이름은 정의 이름과 거의 일치합니다(IBM MQ 파일 시스템 이해 참조).
채널/	
clntconn/	
리스너 /	
이름 목록 /	
procdef/	
큐 /	
services/	
topics/	
...	IBM MQ에 사용되는 기타 디렉토리(예를 들어, @ipcc), IBM MQ로만 수정됩니다.
사용자 데이터/	애플리케이션의 지속적 상태를 저장하는 데 사용할 수 있습니다(큐 관리자를 다른 노드로 이동할 때 RDQM에서 사용될 수 있음 - 지속적 애플리케이션 상태 저장 참조).
***DataPath*\autocfg**	자동 구성에 사용됨

/var/mqm/log/*qmname*/

/var/mqm/log/*qmname*/에는 큐 관리자 로그 파일이 포함되어 있습니다. 활성 큐 관리자 인스턴스의 독점 액세스를 위해 디렉토리가 잠깁니다. 디렉토리 경로는 qm.ini 파일에서 또는 **crtmqm** 명령의 **ld** 옵션을 사용하여 수정할 수 있습니다.

표 *15. AIX and Linux*에서 */var/mqm/log/qmname* 디렉토리의 문서화된 컨텐츠	
디렉토리 또는 파일 이름	**컨텐츠**
amqhlctl.lfh	로그 제어 파일.
active/	이 디렉토리는 S0000000.LOG, S0000001.LOG, S0000002.LOG 등으로 번호가 지정된 로그 파일을 포함합니다.

/opt/mqm

/opt/mqm은 기본적으로 대부분의 플랫폼에서 설치 디렉토리입니다. 엔터프라이즈가 사용하는 단일 또는 복수 플랫폼의 설치 디렉토리에 필요한 공간 크기에 대한 자세한 정보는 98 페이지의 『멀티플랫폼에서 디스크 공간 요구사항』의 내용을 참조하십시오.

Linux AIX *AIX and Linux* 시스템의 예 디렉토리 구성

AIX and Linux 시스템의 대체 파일 시스템 구성 예.

IBM MQ 디렉토리 구조를 다양한 방식으로 사용자 정의하여 많은 다른 목표를 달성할 수 있습니다.

- 다중 인스턴스 큐 관리자를 구성하려면 qmgrs 및 log 디렉토리를 원격 공유 파일 시스템에 배치하십시오.

- 입출력 경합을 줄임으로써 성능을 향상시키려면 데이터 및 로그 디렉토리에 분리된 파일 시스템을 사용하고 디렉토리를 서로 다른 디스크에 할당하십시오.
- 성능에 큰 영향을 미치는 디렉토리에 대해 보다 빠른 스토리지 디바이스를 사용하십시오. 종종 디바이스가 로컬 또는 원격으로 마운트되었는지 여부보다 물리적 디바이스 대기 시간이 지속 메시징 성능에 있어서 보다 중요한 요인입니다. 다음 목록은 성능에 미치는 영향이 최소 및 최대인 디렉토리를 보여줍니다.

  1. `log`
  2. `qmgrs`
  3. 기타 디렉토리(`/usr/mqm` 포함)
- 예를 들어, 중복 디스크 어레이와 같이 복원성이 뛰어난 스토리지에 할당된 파일 시스템에 `qmgrs` 및 `log` 디렉토리를 작성하십시오.
- 네트워크 파일 시스템에 관련된 오류를 로깅할 수 있도록 공용 오류 로그는 네트워크 파일 시스템보다 `var/mqm/errors`에 로컬로 저장하는 것이 더 좋습니다.

117 페이지의 그림 36은 대체 IBM MQ 디렉토리 구조가 파생된 템플리트입니다. 템플리트에서 점선은 구성 가능한 경로를 나타냅니다. 예에서 점선은 AMQ_MQS_INI_LOCATION 환경 변수와 `mqs.ini` 및 `qm.ini` 파일에 저장된 구성 정보에 해당하는 실선으로 대체됩니다.

참고: 경로 정보는 `mqs.ini` 또는 `qm.ini` 파일에 나타나는 대로 표시됩니다. **crtmqm** 명령에 경로 매개변수를 제공하는 경우 큐 관리자 디렉토리의 이름을 생략하십시오. IBM MQ에서 경로에 큐 관리자 이름을 추가합니다.

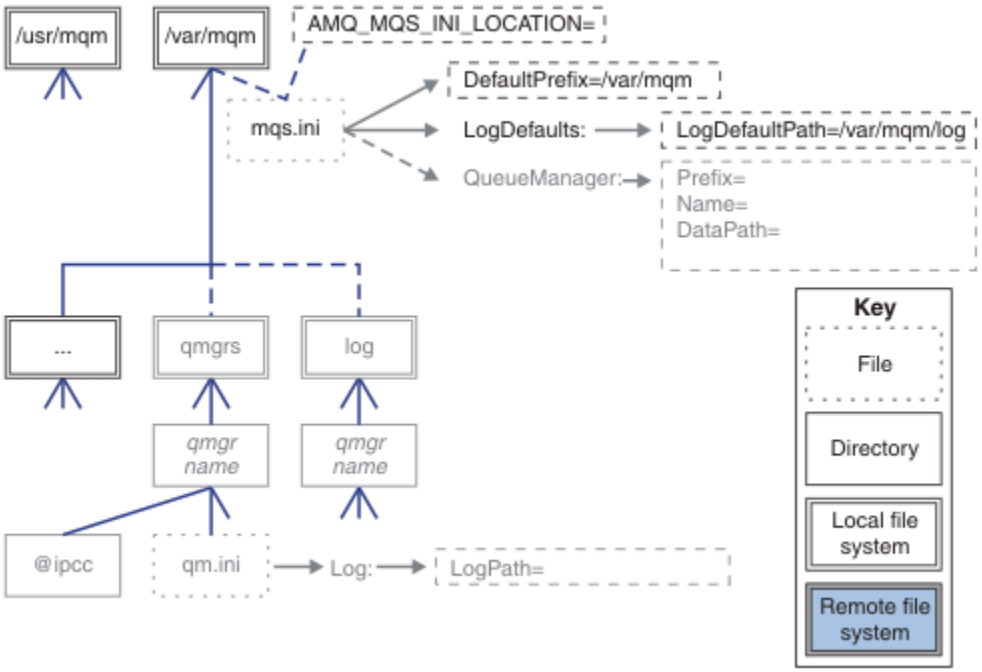


그림 *36.* 디렉토리 구조 패턴 템플리트

IBM MQ 의 일반 디렉토리 구조

118 페이지의 그림 37 는 **crtmqm** *QM1*명령을 실행하여 IBM MQ 에서 작성되는 기본 디렉토리 구조입니다.

`mqs.ini` 파일에는 `DefaultPrefix`의 값을 참조하여 작성된 QM1 큐 관리자에 대한 스탠자가 있습니다. `qm.ini` 파일의 `Log` 스탠자에는 `mqs.ini`의 `LogDefaultPath`를 참조하여 설정된 `LogPath`의 값이 있습니다.

선택적 **crtmqm** 매개변수를 사용하여 `DataPath` 및 `LogPath`의 기본값을 대체하십시오.

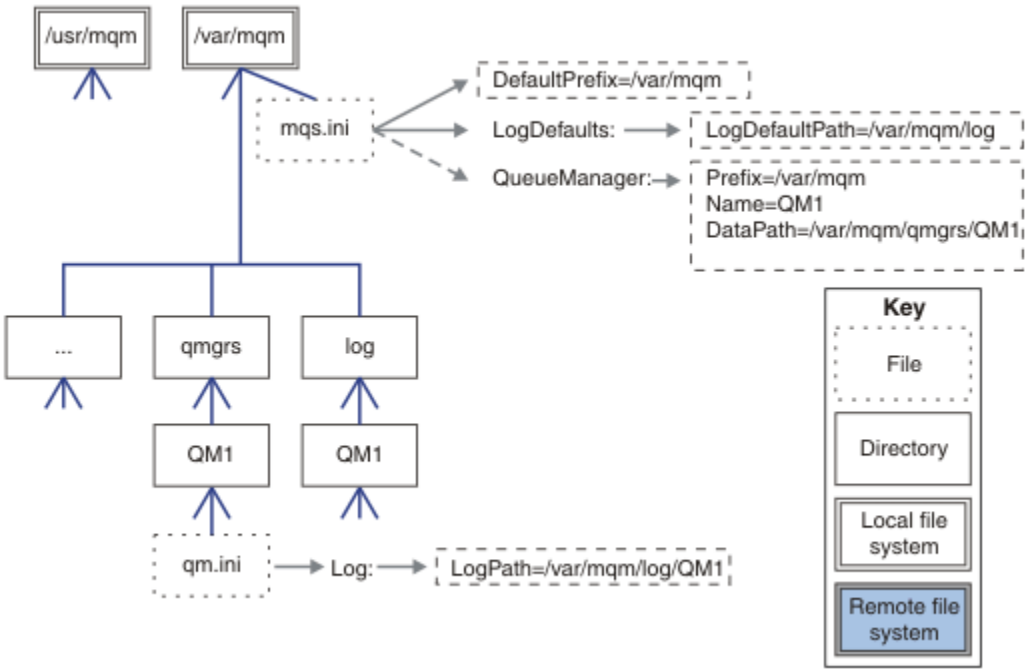


그림 *37. AIX and Linux* 시스템의 기본 *IBM MQ* 디렉토리 구조 예제

기본 qmgrs 및 log 디렉토리 공유

119 페이지의 『모두 공유』의 대안은 `qmgrs` 및 `log` 디렉토리를 별도로 공유하는 것입니다 (118 페이지의 그림 38). 이 구성에서는 기본 mqs.ini 가 로컬 `/var/mqm` 파일 시스템에 저장되므로 `AMQ_MQS_INI_LOCATION` 를 설정할 필요가 없습니다. `mqclient.ini` 및 `mqserver.ini`와 같은 파일과 디렉토리도 공유되지 않습니다.

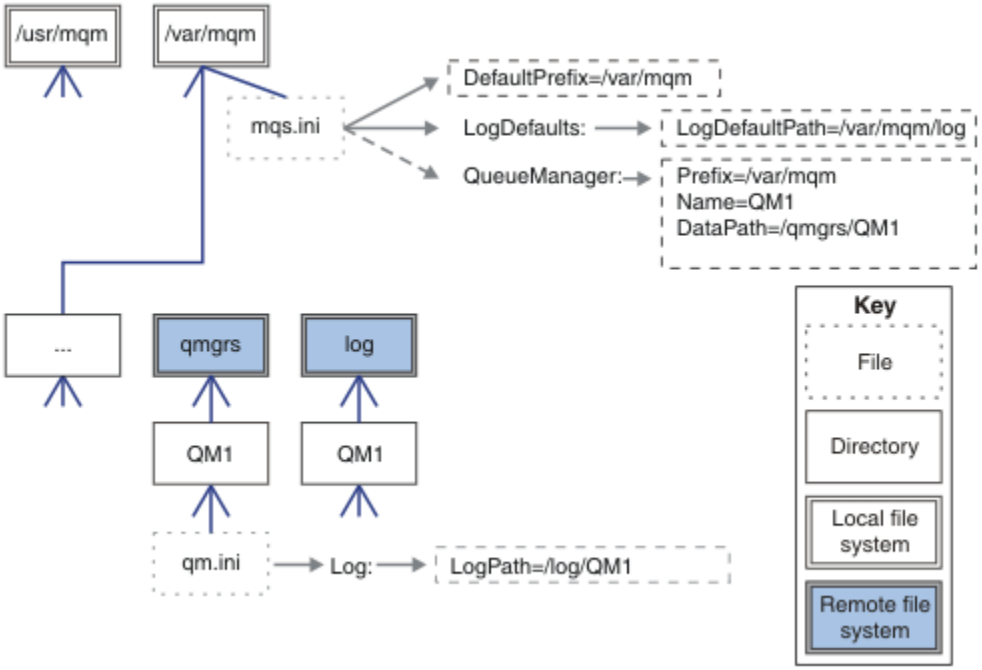


그림 *38. qmgrs* 및 *log* 디렉토리 공유

이름 지정된 qmgrs 및 log 디렉토리 공유

119 페이지의 그림 39의 구성은 log 및 qmgrs를 /ha라는 이름 지정된 공용 원격 공유 파일 시스템에 배치합니다. 동일한 실제 구성을 두 가지 다른 방식으로 작성할 수 있습니다.

1. LogDefaultPath=/ha를 설정한 후 **crtmqm** - md */ha/qmgrs* QM1 명령을 실행하십시오. 결과는 119 페이지의 그림 39의 설명과 정확히 같습니다.
2. 기본 경로를 변경하지 않은 채 그대로 두고 **crtmqm** - ld */ha/log* - md */ha/qmgrs* QM1 명령을 실행하십시오.

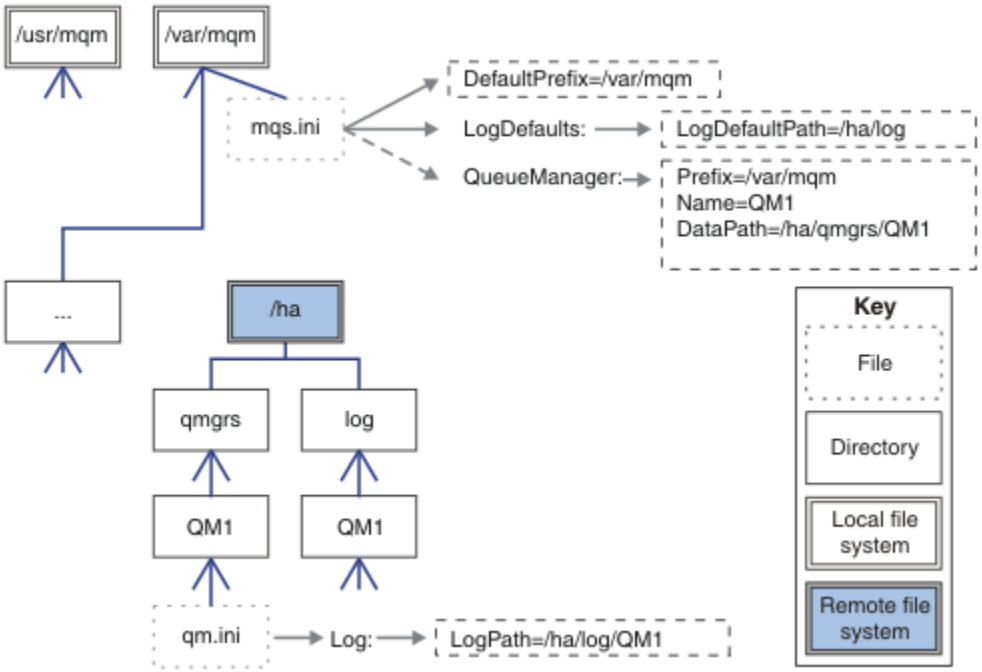


그림 *39.* 이름 지정된 *qmgrs* 및 *log* 디렉토리 공유

모두 공유

120 페이지의 그림 40은 빠른 네트워크 파일 스토리지가 있는 시스템의 단순 구성입니다.

/var/mqm을 원격 공유 파일 시스템으로 마운트하십시오. 기본적으로 QM1이 시작되면 /var/mqm을 검색하여 공유 파일 시스템에서 찾고 /var/mqm에서 mqs.ini 파일을 읽습니다. 모든 서버의 큐 관리자에 단일 /var/mqm/mqs.ini 파일을 사용하는 대신, 각 서버에 AMQ_MQS_INI_LOCATION 환경 변수를 설정하여 상이한 mqs.ini 파일을 가리킬 수 있습니다.

참고: /var/mqm/errors/의 일반 오류 파일 컨텐츠는 여러 서버의 큐 관리자 간에 공유됩니다.

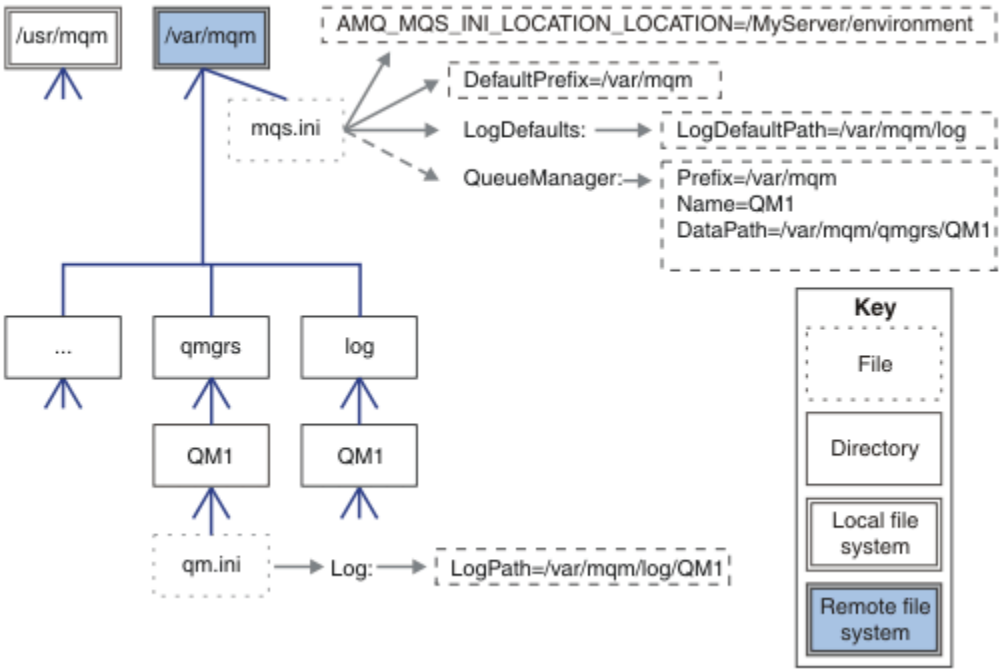


그림 *40.* 모두 공유

다중 인스턴스 큐 관리자에는 이를 사용할 수 없음에 유의하십시오. 다중 인스턴스 큐 관리자의 각 호스트는 세마포어 및 공유 메모리와 같은 로컬 데이터를 추적하기 위해 `/var/mqm`의 자체 로컬 사본을 보유해야 하기 때문입니다. 이 엔티티는 호스트에서 공유할 수 없습니다.

Windows Windows 시스템의 디렉토리 구조

Windows의 큐 관리자 구성 정보 및 디렉토리를 찾는 방법을 설명합니다.

IBM MQ for Windows 설치의 기본 디렉토리는 다음과 같습니다.

프로그램 디렉토리
```
C:\Program Files\IBM\MQ
```
데이터 디렉토리
```
씨:\ProgramData \IBM \MQ
```

중요사항: Windows Windows 설치의 경우, 디렉토리는 레지스트리 입력 항목이나 큐 관리자 또는 모두가 포함된 제품의 이전 설치가 없는 경우 명시된 대로입니다. 이 경우 새 설치에서 이전 데이터 디렉토리 위치가 사용됩니다. 자세한 정보는 프로그램 및 데이터 디렉토리 위치를 참조하십시오.

사용 중인 설치 디렉토리 및 데이터 디렉토리를 알고 싶으면 dspmqver 명령을 실행하십시오.

설치 디렉토리는 **InstPath** 필드에 나열되고 데이터 디렉토리는 **DataPath** 필드에 나열됩니다.

dspmqver 명령을 실행하면 다음과 같은 정보가 표시됩니다.

```
>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
```

다중 인스턴스 큐 관리자

다중 인스턴스 큐 관리자를 구성하려면 로그 및 데이터 디렉토리가 큐 관리자의 인스턴스를 실행 중인 서버와 다른 서버의 네트워크 스토리지에 있어야 합니다.

큐 관리자 데이터와 로그 디렉토리의 위치를 쉽게 지정할 수 있도록 **crtmqm** 명령에 두 개의 매개변수, **-md** 및 **-ld**가 제공됩니다. **-md** 매개변수를 지정하는 효과는 네 배입니다.

1. mqs.ini 스탠자 QueueManager*QmgrName*에 새로운 변수 *DataPath*가 포함되며, 이는 큐 관리자 데이터 디렉토리를 가리킵니다. *Prefix* 변수와 다르게, 큐 관리자 디렉토리의 이름이 경로에 포함됩니다.
2. mqs.ini 파일에 저장된 큐 관리자 구성 정보는 *Name*, *Prefix*, *Directory*, *DataPath*로 감소합니다.

Windows 디렉토리 컨텐츠

IBM MQ 디렉토리의 위치 및 컨텐츠를 나열합니다.

IBM MQ 구성은 세 개의 주요 파일 및 디렉토리 세트를 포함합니다.

1. 유지보수가 적용될 때에만 업데이트되는 실행 파일 및 읽기 전용 파일. 예를 들면, 다음과 같습니다.
   - Readme 파일
   - IBM MQ 탐색기 플러그인 및 도움말 파일
   - 라이센스 파일

   이 파일은 121 페이지의 표 16에 설명되어 있습니다.
2. 부분 큐 관리자에 고유하지 않은 잠재적으로 수정 가능한 파일 및 디렉토리. 이 파일과 디렉토리는 122 페이지의 표 17에 설명되어 있습니다.
3. 서버의 각 큐 관리자에 고유한 파일 및 디렉토리. 이 파일과 디렉토리는 122 페이지의 표 18에 설명되어 있습니다.

자원 디렉토리 및 파일

자원 디렉토리와 파일은 큐 관리자를 실행하기 위한 모든 실행 코드와 자원을 포함합니다. 설치 특정 IBM MQ 구성 레지스트리 키에 있는 *FilePath*변수에는 자원 디렉토리에 대한 경로가 포함되어 있습니다.

표 *16. FilePath* 디렉토리에 있는 디렉토리 및 파일

파일 경로	**컨텐츠**
FilePath\bin	명령 및 DLL
FilePath\bin64	명령 및 DLL(64비트)
FilePath\conv	데이터 변환 테이블
FilePath\doc	마법사 도움말 파일
FilePath\MQExplorer	탐색기 및 탐색기 도움말 Eclipse 플러그인
FilePath\gskit8	글로벌 보안 킷
FilePath\java	JRE를 포함한, Java 자원
FilePath\licenses	라이센스 정보
FilePath\Non_IBM_License	라이센스 정보
FilePath\properties	내부적으로 사용됨
FilePath\Tivoli	
FilePath\tools	개발 자원 및 샘플

표 *16. FilePath* 디렉토리에 있는 디렉토리 및 파일 *(계속)*	
파일 경로	**컨텐츠**
FilePath\web	편집 불가능한 파일에 대한 IBM MQ Console 및 REST API 설치 컴포넌트 파일 구조에서 설명합니다.
FilePath\Uninst	내부적으로 사용됨
FilePath\README.TXT	Readme 파일

큐 관리자에 고유하지 않은 디렉토리

일부 디렉토리는 추적 파일 및 오류 로그와 같이 특정 큐 관리자에 고유하지 않은 파일을 포함합니다. *DefaultPrefix* 변수는 이러한 디렉토리의 경로를 포함합니다. *DefaultPrefix*는 AllQueueManagers 스탠자의 일부입니다.

표 *17. DefaultPrefix* 디렉토리에 있는 디렉토리 및 파일	
파일 경로	**컨텐츠**
DefaultPrefix\config	내부적으로 사용됨
DefaultPrefix\conv	ccsid_part2.tbl 및 ccsid.tbl data 변환 제어 파일(데이터 변환에 설명됨)
DefaultPrefix\errors	큐 관리자에 해당되지 않은 오류 로그 AMQERR *nn*.LOG
DefaultPrefix\exits	채널 엑시트 프로그램
DefaultPrefix\exits64	채널 엑시트 프로그램(64비트)
DefaultPrefix\ipc	사용되지 않음
DefaultPrefix\qmgrs	122 페이지의 표 18에 설명됨
DefaultPrefix\trace	추적 파일
DefaultPrefix\web	사용자 편집 가능 파일에 대한 IBM MQ Console 및 REST API 설치 컴포넌트 파일 구조에서 설명합니다.
DefaultPrefix\amqmjpse.txt	내부적으로 사용됨

큐 관리자 디렉토리

큐 관리자를 작성하면 큐 관리자에 고유한 디렉토리의 새로운 세트가 작성됩니다.

-md *filepath* 매개변수를 사용하여 큐 관리자를 작성하는 경우 경로는 mqs.ini 파일의 큐 관리자 스탠자에 있는 *DataPath* 변수에 저장됩니다. **-md** *filepath* 매개변수를 설정하지 않고 큐 관리자를 작성하는 경우, 큐 관리자 디렉토리는 *DefaultPrefix*에 저장된 경로에 작성되며 경로는 mqs.ini 파일의 큐 관리자 스탠자에 있는 *Prefix* 변수로 복사됩니다.

표 *18. DataPath* 및 *Prefix**qmgrs**QmgrName* 디렉토리에 있는 디렉토리 및 파일	
파일 경로	**컨텐츠**
DataPath\@ipcc	클라이언트 연결 테이블, AMQCLCHL.TAB의 기본 위치
DataPath\authinfo	내부적으로 사용됨
DataPath\channel	
DataPath\clntconn	
DataPath\errors	오류 로그, AMQERR *nn*.LOG

표 *18. DataPath* 및 *Prefix*\qmgrs*QmgrName* 디렉토리에 있는 디렉토리 및 파일 *(계속)*	
파일 경로	**컨텐츠**
DataPath\listener	내부적으로 사용됨
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	큐 관리자 구성
DataPath\qmstatus.ini	큐 관리자 상태
DataPath\userdata	애플리케이션의 지속적 상태를 저장하는 데 사용할 수 있습니다.
Prefix\qmgrs*QmgrName*	내부적으로 사용됨
Prefix\qmgrs\@SYSTEM	사용되지 않음
Prefix\qmgrs\@SYSTEM\errors	
*DataPath**autocfg*	자동 구성에 사용됨

IBM i IBM i의 디렉토리 구조

IFS에 대한 설명이 제공되고 IBM MQ IFS 디렉토리 구조가 서버, 클라이언트 및 Java에 대해 설명됩니다.

통합 파일 시스템 (IFS) 은 서버에 저장된 모든 정보에 대한 통합 구조를 제공하면서 개인용 컴퓨터, AIX and Linux 운영 체제와 유사한 스트림 입/출력 및 스토리지 관리를 지원하는 IBM i 의 일부입니다.

IBM i 디렉토리 이름은 @ (at) 문자가 아니라 & (ampersand) 문자로 시작합니다. 예를 들어 IBM i의 @system은(는) &system입니다.

IBM MQ 서버의 IFS 루트 파일 시스템

IBM MQ Server for IBM i를 설치하면 IFS 루트 파일 시스템에 다음 디렉토리가 작성됩니다.

ProdData:

개요

```
QIBM
   '-- ProdData
      '-- mqm
```

```
'-- doc
'-- inc
'-- lib
'-- samp
'-- licenses
'-- LicenseDoc
'-- 5724H72_V8R0M0
```

/QIBM/ProdData/mqm
이 디렉토리 아래의 서브디렉토리는 모든 제품 데이터(예: C++ 클래스, 추적 형식 파일, 라이센스 파일)를 포함합니다. 제품을 설치할 때마다 이 디렉토리의 데이터는 삭제되고 대체됩니다.

/QIBM/ProdData/mqm/doc
CL 명령에 대한 명령 참조서는 HTML 형식으로 제공되고 여기에 설치됩니다.

/QIBM/ProdData/mqm/inc
C 또는 C++ 프로그램 컴파일을 위한 헤더 파일.

/QIBM/ProdData/mqm/lib
MQ에 사용되는 보조 파일.

/QIBM/ProdData/mqm/samp
추가 샘플.

/QIBM/ProdData/mqm/licenses
라이센스 파일. 각 언어의 두 파일은 LA_ *xx* 및 LI_ *xx* 과(와) 같은 이름으로 지정됩니다. 여기서, *xx* 은(는) 제공된 각 언어의 2문자 언어 ID입니다.

다음 디렉토리도 라이센스 계약 파일을 저장합니다.

/QIBM/ProdData/LicenseDoc/5724H72_V8R0M0
라이센스 파일. 파일 이름은 5724H72_V8R0M0_ *xx* 과(와) 같이 지정됩니다. 여기서, *xx* 은(는) 제공된 각 언어의 2 - 5자 언어 ID입니다.

UserData:

개요

```
QIBM
   '-- UserData
      '-- mqm
      '-- errors
      '-- trace
      '-- qmgrs
      '-- &system
      '-- qmgrname1
      '-- qmgrname2
      '-- and so on
```

/QIBM/UserData/mqm
이 디렉토리 아래의 서브디렉토리는 큐 관리자에 관련된 모든 사용자 데이터를 포함합니다.

제품을 설치할 때 /QIBM/UserData/mqm/ 디렉토리에 mqs.ini 파일이 작성됩니다(이전 설치에서 이미 작성된 경우 제외).

큐 관리자를 작성할 때 qm.ini 파일은 /QIBM/UserData/mqm/qmgrs/ *QMGRNAME* / 디렉토리에 작성됩니다(여기서, *QMGRNAME*은 큐 관리자의 이름).

제품이 삭제될 때 디렉토리의 데이터가 보유됩니다.

IBM MQ MQI client의 IFS 루트 파일 시스템

IBM MQ MQI client for IBM i를 설치하면 IFS 루트 파일 시스템에 다음 디렉토리가 작성됩니다.

ProdData:

개요

```
QIBM
   '-- ProdData
      '-- mqm
      '-- lib
```

/QIBM/ProdData/mqm
이 디렉토리 아래의 서브디렉토리는 모든 제품 데이터를 포함합니다. 제품을 바꿀 때마다 이 디렉토리의 데이터는 삭제되고 대체됩니다.

UserData:

개요

```
QIBM
   '-- UserData
      '-- mqm
      '-- errors
      '-- trace
```

/QIBM/UserData/mqm
이 디렉토리 아래의 서브디렉토리는 모든 사용자 데이터를 포함합니다.

IBM MQ Java의 IFS 루트 파일 시스템

IBM MQ Java를 IBM i에 설치하면 IFS 루트 파일 시스템에 다음 디렉토리가 작성됩니다.

ProdData:

개요

```
QIBM
   '-- ProdData
      '-- mqm
      '-- java
      '--samples
      '-- bin
      '-- lib
```

/QIBM/ProdData/mqm/java
이 디렉토리 아래의 서브디렉토리는 Java 클래스를 포함하여 모든 제품 데이터를 포함합니다. 제품을 바꿀 때마다 이 디렉토리의 데이터는 삭제되고 대체됩니다.

/QIBM/ProdData/mqm/java/samples
이 디렉토리 아래의 서브디렉토리는 모든 샘플 Java 클래스 및 데이터를 포함합니다.

서버 및 클라이언트 설치로 작성되는 라이브러리

IBM MQ 서버 또는 클라이언트를 설치하면 다음 라이브러리가 작성됩니다.

- QMQM

  제품 라이브러리.

- QMQMSAMP

  샘플 라이브러리(샘플을 설치하도록 선택한 경우).

- QMxxxx

  서버 전용.

큐 관리자를 작성할 때마다 IBM MQ에서는 QMxxxx과(와) 같은 이름으로 연관된 라이브러리를 자동으로 작성합니다. 여기서 xxxx은(는) 큐 관리자 이름에서 파생됩니다. 이 라이브러리는 저널 및 연관된 수신자를 포함하여 큐 관리자 고유의 오브젝트를 포함합니다. 기본적으로 이 라이브러리의 이름은 문자 QM으로 접두부가 붙은 큐 관리자의 이름에서 파생됩니다. 예를 들어, TEST라는 큐 관리자의 경우 라이브러리는 QMTEST가 됩니다.

참고: 큐 관리자를 작성할 때 원하는 경우 라이브러리의 이름을 지정할 수 있습니다. 예를 들면, 다음과 같습니다.

```
CRTMQM MQMNAME(TEST) MQMLIB(TESTLIB)
```

WRKLIB 명령을 사용하여 IBM MQ for IBM i가 작성한 모든 라이브러리를 나열할 수 있습니다. 큐 관리자 라이브러리에 대해 QMGR: QMGRNAME 텍스트가 표시됩니다. 명령의 형식은 다음과 같습니다.

```
WRKLIB LIB(QM*)
```

제품을 삭제해도 큐 관리자 연관 라이브러리는 보유됩니다.

Multi 멀티플랫폼의 MFT 에 대한 파일 시스템 지원 계획

IBM MQ Managed File Transfer MFT 에이전트를 사용하여 파일 시스템의 파일로 (부터) 데이터를 전송할 수 있습니다. 또한 에이전트 내에서 실행 중인 자원 모니터를 파일 시스템의 파일을 모니터하도록 구성할 수 있습니다.

MFT 에는 이러한 파일이 잠금을 지원하는 파일 시스템에 저장되어야 한다는 요구사항이 있습니다. 여기에는 다음과 같은 두 가지 이유가 있습니다.

- 에이전트는 파일에서 데이터를 읽거나 데이터를 쓰기 시작한 후에도 변경되지 않도록 파일을 잠급니다.
- 자원 모니터 잠금 파일을 사용하여 현재 다른 프로세스에서 사용하고 있지 않은지 확인합니다.

에이전트 및 자원 모니터는 Java 메소드 **FileChannel.tryLock()** 를 사용하여 잠금을 수행하며, 파일 시스템은 이 호출을 사용하여 잠금을 요청할 때 파일을 잠글 수 있어야 합니다.

중요사항: 다음 파일 시스템은 MFT의 기술 요구사항을 충족하지 않으므로 지원되지 않습니다.

- GlusterFS
- NFS 버전 3

Multi 멀티플랫폼에서 순환 또는 선형 로깅 선택

IBM MQ에서 순환 또는 선형 로깅을 선택할 수 있습니다. 다음 정보에서 이 두 가지 유형에 대한 개요를 제공합니다.

순환 로깅의 이점

순환 로깅의 주요 이점은 다음과 같습니다.

- 더 용이한 관리.

  워크로드에 대한 순환 로깅을 올바로 구성한 경우 추가 관리가 필요하지 않습니다. 반면 선형 로깅의 경우 미디어 이미지를 기록해야 하고 더 이상 필요하지 않은 로그 익스텐트는 아카이브하거나 삭제해야 합니다.

- 성능 개선.

  순환 로깅은 선형 로깅보다 성능이 개선됩니다. 순환 로깅은 이미 포맷된 로그 익스텐트를 다시 사용할 수 있기 때문입니다. 반면 선형 로깅은 새 로그 익스텐트를 할당하고 이를 포맷해야 하기 때문입니다.

자세한 정보는 로그 관리를 참조하십시오.

선형 로깅의 이점

선형 로깅의 기본 이점은 선형 로깅은 더 많은 장애에 대한 보호 기능을 제공한다는 것입니다.

순환 로깅이나 선형 로깅 모두 애플리케이션이나 관리자가 삭제한 메시지나 큐 또는 손상되거나 삭제된 로그를 보호하지 않습니다.

선형 로깅을 사용하면 손상된 오브젝트의 복구가 가능하지만, 순환 로깅에서는 불가능합니다. 따라서 선형 로깅은 큐 파일을 손상이나 삭제로부터 보호합니다. 이렇게 손상된 큐를 선형 로그에서 복구할 수 있기 때문입니다.

순환 로깅 및 선형 로깅 모두 정전 또는 통신 장애 복구에서 설명한대로 정전 및 통신 장애로부터 보호합니다.

기타 고려사항

선형 또는 순환 로깅을 선택하는지 여부는 필요한 중복성 양에 따라 결정됩니다.

선형 로깅에서는 성능 비용 및 관리 비용으로 인해 더 많은 중복성을 선택해야 하는 손실이 있습니다.

자세한 정보는 로깅 유형을 참조하십시오.

AIX AIX의 공유 메모리

특정 애플리케이션 유형이 AIX 메모리 제한으로 인해 연결에 실패하면 대부분의 경우 환경 변수 EXTSHM=ON을 설정하여 이를 해결할 수 있습니다.

AIX의 일부 32비트 프로세스는 IBM MQ 큐 관리자에 연결하는 기능에 영향을 미치는 운영 체제 제한 문제가 있을 수 있습니다. IBM MQ의 모든 표준 연결은 공유 메모리를 사용하지만 기타 UNIX 플랫폼과 다르게, AIX는 32비트 프로세스가 11개의 공유 메모리 세트만을 첨부하도록 허용합니다.

대부분의 32비트프로세스에서는 이 한계가 발생하지 않지만 메모리 요구사항이 높은 애플리케이션은 이유 코드 2102: MQRC_RESOURCE_PROBLEM으로 IBM MQ 에 연결하는 데 실패할 수 있습니다. 다음 애플리케이션 유형이 이 오류를 표시할 수 있습니다.

- 32비트 Java 가상 머신에서 실행하는 프로그램
- 대용량 또는 초대용량 메모리 모델을 사용하는 프로그램
- 많은 큐 관리자 또는 데이터베이스에 연결하는 프로그램
- 자체적으로 공유 메모리 세트를 첨부한 프로그램

AIX는 32비트 프로세스가 더 많은 공유 메모리를 첨부할 수 있도록 이 프로세스에 대한 확장 공유 메모리 기능을 제공합니다. 이 기능과 함께 애플리케이션을 실행하려면 큐 관리자 및 프로그램을 시작하기 전에 환경 변수 EXTSHM=ON을 내보내십시오. EXTSHM=ON 기능은 대부분의 경우에 이 오류를 차단하지만 shmctl 함수의 SHM_SIZE 옵션을 사용하는 프로그램에는 호환되지 않습니다.

IBM MQ MQI client 애플리케이션 및 모든 64비트 프로세스는 이 제한의 영향을 받지 않습니다. EXTSHM이 설정되었는지 여부와 상관 없이 IBM MQ 큐 관리자에 연결할 수 있습니다.

Linux AIX IBM MQ 및 UNIX System V IPC 자원

큐 관리자는 일부 IPC 자원을 사용합니다. **ipcs -a**를 사용하여 사용 중인 자원을 알아보십시오.

이 정보는 AIX and Linux 시스템에서 실행 중인 IBM MQ 에만 적용됩니다.

IBM MQ는 System V 프로세스 간 통신(IPC) 자원(세마포어 및 공유 메모리 세그먼트)을 사용하여 시스템 컴포넌트 간에 데이터를 저장하고 전달합니다. 이러한 자원은 큐 관리자에 연결하는 애플리케이션 및 큐 관리자 프로세스에 사용됩니다. IBM MQ MQI clients는 IBM MQ 추적 제어를 제외하고 IPC 자원을 사용하지 않습니다. 시스템에 현재 사용 중인 IPC 자원의 크기와 수에 대한 전체 정보를 보려면 UNIX 명령 **ipcs -a**를 사용하십시오.

Linux AIX IBM MQ 및 UNIX 프로세스 우선순위

프로세스 우선순위 *nice* 값을 설정할 때의 우수 사례입니다.

이 정보는 AIX and Linux 시스템에서 실행 중인 IBM MQ 에만 적용됩니다.

백그라운드에서 프로세스를 실행하는 경우 쉘 호출을 통해 해당 프로세스에 보다 높은 *nice* 값(및 그에 따른 낮은 우선순위)이 제공될 수 있습니다. 이는 일반적인 IBM MQ 성능 제한사항이 될 수 있습니다. 스트레스가 심한 상황에서, 높은 우선순위의 여러 실행 준비된 스레드가 있고 낮은 우선순위의 일부 스레드가 있으면 운영 체제 스케줄링 특성이 낮은 우선순위 스레드의 프로세서 시간을 빼앗을 수 있습니다.

큐 관리자와 연관된 독립적으로 시작된 프로세스 (예: **runmqlsr**)가 연관된 큐 관리자와 같은 *nice* 값을 가지는 것은 좋은 사례입니다. 쉘이 이러한 백그라운드 프로세스에 높은 *nice* 값을 지정하지 않는지 확인하십시오. 예를

들어, ksh에서 "`set +o bgnice`" 설정을 사용하여 ksh가 백그라운드 프로세스의 *nice* 값을 높이는 것을 중지하십시오. "ps -efl" 목록의 *NI* 열을 검토하여 실행 프로세스의 *nice* 값을 확인할 수 있습니다.

또한 큐 관리자와 동일한 *nice* 값으로 IBM MQ 애플리케이션 프로세스를 시작하십시오. 상이한 *nice* 값으로 실행할 경우 애플리케이션 스레드가 큐 관리자 스레드를 차단하여(또는 그 반대로) 성능이 저하될 수 있습니다.

z/OS Planning your IBM MQ environment on z/OS

When planning your IBM MQ environment, you must consider the resource requirements for data sets, page sets, Db2, Coupling Facilities, and the need for logging, and backup facilities. Use this topic to plan the environment where IBM MQ runs.

Before you plan your IBM MQ architecture, familiarize yourself with the basic IBM MQ for z/OS concepts, see the topics in IBM MQ for z/OS concepts.

When planning your queue manager, you might need to work with different people in your organization. It is usually a good idea to involve those people early, as change control procedures can take a long time. They might also be able to tell you what parameters you need to configure IBM MQ for z/OS.

For example you might need to work with the:

- Storage administrator, to determine the high level qualifier of queue manager data sets, and to allocate enough space for queue manager data sets.
- z/OS system programmer to define the IBM MQ subsystem to z/OS and APF authorize the IBM MQ for z/OS libraries.
- Network administrator to determine which TCP/IP stack and ports should be used for IBM MQ for z/OS.
- Security administrator to set up access to queue manager data sets, security profiles for IBM MQ for z/OS resources, and TLS certificates.
- Db2 administrator to set up Db2 tables when configuring a queue sharing group.

Related concepts

IBM MQ Technical overview

Related tasks

"IBM MQ 아키텍처 계획" on page 5
IBM MQ 환경을 계획할 때, IBM MQ가 단일 및 다중 큐 관리자 아키텍처와, 포인트-투-포인트 및 발행/구독 메시징 스타일에 대해 제공하는 지원을 고려하십시오. 또한 자원 요구사항과, 로깅 및 백업 기능 사용을 계획하십시오.

Configuring z/OS

Administering IBM MQ for z/OS

z/OS Planning for your queue manager

When you are setting up a queue manager, your planning should allow for the queue manager to grow, so that the queue manager meets the needs of your enterprise.

The best way to configure a queue manager is in steps:

1. Configure the base queue manager
2. Configure the channel initiator which does queue manager to queue manager communications, and remote client application communication
3. If you want to encrypt and protect messages, configure Advanced Message Security
4. If you want to use File Transfer over IBM MQ, configure Managed File Transfer for z/OS.
5. If you want to use the administrative or messaging REST API, or the IBM MQ Console to manage IBM MQ from a web browser, configure the mqweb server.

Some enterprises have hundreds of thousands of queue managers in their environment. You need to consider your IBM MQ network now, and in five years time.

On z/OS, some queue managers process thousands of messages a second, and log over 100 MB a second. If you expect very high volumes you may need to consider having more than one queue manager.

On z/OS, IBM MQ can run as part of a queue sharing group (QSG) where messages are stored in the Coupling Facility, and any queue manager in the queue sharing group can access the messages. If you want to run in a queue sharing group you need to consider how many queue managers you need. Typically, there is one queue manager for each LPAR. You might also have one queue manager to backup CF structures regularly.

Some changes to configuration are easy to do, such as defining a new queue. Some are harder, such as making logs and page sets bigger; and some configuration cannot be changed, such as the name of a queue manager or the queue sharing group name.

There is performance and tuning information available in the MP16 performance SupportPac .

Naming conventions

You need to have a naming convention for the queue manager data sets.

Many enterprises use the release number in the name of the load libraries, and so on. You might want to consider having an alias of `MQM.SCSQAUTH` pointing to the version currently in use, such as `MQM.V930.SCSQAUTH`, so you do not have to change CICS®, Batch, and IMS JCL when you migrate to a new version of IBM MQ.

You can use a symbolic link in z/OS UNIX System Services to reference the installation directory for the version of IBM MQ currently in use.

The data sets used by the queue manager (logs, page sets, JCL libraries) need a naming convention to simplify the creation of security profiles, and the mapping of data sets to SMS storage classes that control where the data sets are placed on disk, and the attributes they have.

Note, that putting the version of IBM MQ into the name of the page sets or logs, is not a good idea. One day you might migrate to a new version, and the data set will have the "wrong" names.

Applications

You need to understand the business applications and the best way to configure IBM MQ. For example if applications have logic to provide recovery and repeat capability, then non persistent messages might be good enough. If you want IBM MQ to handle the recovery, then you need to use persistent messages and put and get messages in syncpoint.

You need to isolate queues from different business transactions. If a queue for one business application fills up, you do not want this impacting other business applications. Isolate the queues in different page sets and buffer pools, or structures, if possible.

You need to understand the profile of messages. For many applications the queues have only a few messages. Other applications can have queues build up during the day, and be processed overnight. A queue which normally has only a few messages on it, might need to hold many hours worth of messages if there is a problem and messages are not processed. You need to size the CF structures and page sets to allow for your expected peak capacity.

Post configuration

Once you have configured your queue manager (and components) you need to plan for:

- Backing up page sets.
- Backing up definitions of objects.
- Automating the backup of any CF structures.
- Monitoring IBM MQ messages, and taking action when a problem is detected.
- Collecting the IBM MQ statistics data.

- Monitoring resource usage, such as virtual storage, and amount of data logged per hour. With this you can see if your resource usage is increasing and if you need to take actions, such as setting up a new queue manager

z/OS Planning your storage and performance requirements on z/OS

You must set realistic and achievable storage, and performance goals for your IBM MQ system. Use this topic help you understand the factors which affect storage, and performance.

This topic contains information about the storage and performance requirements for IBM MQ for z/OS. It contains the following sections:

- z/OS performance options for IBM MQ
- Determining z/OS workload management importance and velocity goals
- "Library storage" on page 130
- "System LX usage" on page 131
- "Storage configuration" on page 132
- "Disk storage" on page 136

See, "Where to find more information about storage and performance requirements" on page 137 for more information.

z/OS performance options for IBM MQ

With workload management, you define performance goals and assign a business importance to each goal. You define the goals for work in business terms, and the system decides how much resource, such as processor and storage, should be given to the work to meet its goal. Workload management controls the dispatching priority based on the goals you supply. Workload management raises or lowers the priority as needed to meet the specified goal. Thus, you need not fine-tune the exact priorities of every piece of work in the system and can focus instead on business objectives.

The three kinds of goals are:

Response time
How quickly you want the work to be processed

Execution velocity
How fast the work should be run when ready, without being delayed for processor, storage, I/O access, and queue delay

Discretionary
A category for low priority work for which there are no performance goals

Response time goals are appropriate for end-user applications. For example, CICS users might set workload goals as response time goals. For IBM MQ address spaces, velocity goals are more appropriate. A small amount of the work done in the queue manager is counted toward this velocity goal but this work is critical for performance. Most of the work done by the queue manager counts toward the performance goal of the end-user application. Most of the work done by the channel initiator address space counts toward its own velocity goal. The receiving and sending of IBM MQ messages, which the channel initiator accomplishes, is typically important for the performance of business applications using them.

Determining z/OS workload management importance and velocity goals

See "Determining z/OS workload management importance" on page 131 for more information.

Library storage

You must allocate disk storage for the product libraries. The exact figures depend on your configuration, and should include both the target and distribution libraries, as well as the SMP/E libraries.

The target libraries used by IBM MQ for z/OS use PDSE formats. Ensure that any PDSE target libraries are not shared outside a sysplex. For more information about the required libraries and their sizes and the required format, see the Program Directory. 프로그램 디렉토리에 대한 다운로드 링크는 다음을 참조하세요.IBM MQ for z/OS 프로그램 디렉토리 PDF 파일 .

System LX usage

Each defined IBM MQ subsystem reserves one system linkage index (LX) at IPL time, and a number of non-system linkage indexes when the queue manager is started. The system linkage index is reused when the queue manager is stopped and restarted. Similarly, distributed queuing reserves one non-system linkage index. In the unlikely event of your z/OS system having inadequate system LXs defined, you might need to take these reserved system LXs into account.

If required, the number of system LXs can be increased by setting the *NSYSLX* parameter in SYS1.PARMLIB member IEASYSxx.

z/OS *Determining z/OS workload management importance*

For full information about workload management and defining goals through the service definition, see the .z/OS product documentation.

This topic suggests how to set the z/OS workload management importance and velocity goals relative to other important work in your system. See *z/OS MVS Planning: Workload Management* for more information.

The queue manager address space needs to be defined with high priority as it provides subsystem services. The channel initiator is an application address space, but is usually given a high priority to ensure that messages being sent to a remote queue manager are not delayed. Advanced Message Security (AMS) also provides subsystem services and needs to be defined with high priority.

Use the following service classes:

The default SYSSTC service class

- VTAM and TCP/IP address spaces
- IRLM address space (IRLMPROC)

Note: The VTAM, TCP/IP, and IRLM address spaces must have a higher dispatching priority than all the DBMS address spaces, their attached address spaces, and their subordinate address spaces. Do not allow workload management to reduce the priority of VTAM, TCP/IP, or IRLM to (or below) that of the other DBMS address spaces

A high velocity goal and importance of 1 for a service class with a name that you define, such as PRODREGN, for the following:

- IBM MQ queue manager, channel initiator and AMS address spaces
- Db2 (all address spaces, except for the Db2-established stored procedures address space)
- CICS (all region types)
- IMS (all region types except BMPs)

A high velocity goal is good for ensuring that startups and restarts are performed as quickly as possible for all these address spaces.

The velocity goals for CICS and IMS regions are only important during startup or restart. After transactions begin running, workload management ignores the CICS or IMS velocity goals and assigns priorities based on the response time goals of the transactions that are running in the regions. These transaction goals should reflect the relative priority of the business applications they implement. They might typically have an importance value of 2. Any batch applications using IBM MQ should similarly have velocity goals and importance reflecting the relative priority of the business applications they implement. Typically the importance and velocity goals will be less than those for PRODREGN.

z/OS *Storage configuration*

V 9.4.0 In a 64 bit address space, there is a virtual line called "the bar" that marks the 2GB address. The bar separates storage below the 2GB address, called "below the bar", from storage above the 2GB address, called "above the bar". Storage below the bar uses 31 bit addressability, storage above the bar uses 64 bit addressability.

V 9.4.0

You can specify the limit of 31-bit storage by using the JCL REGION parameter, and the limit of 64-bit storage by using the MEMLIMIT parameter. These specified values can be overridden by z/OS exits.

Suggested storage configuration

The following table shows suggested **REGION** and **MEMLIMIT** values for the queue manager, channel initiator, and AMS address spaces. These suggestions should be used as a starting point and adjusted using the information in:

- "Queue manager storage configuration" on page 132
- "Channel initiator storage configuration from IBM MQ 9.4.0" on page 135

Table 19. Suggested definitions for REGION and MEMLIMIT

Address space	Storage configuration
Queue manager	REGION=0M, MEMLIMIT=3G
V 9.4.0 Channel initiator from IBM MQ 9.4.0	REGION=0M, MEMLIMIT=2G
AMS address space	REGION=0M

Managing the MEMLIMIT and REGION size

Other mechanisms, for example the **MEMLIMIT** parameter in the SMFPRMxx member of SYS1.PARMLIB or the IEFUSI exit might be used at your installation to provide a default amount of virtual storage above the bar for z/OS address spaces. See Memory management above the bar for full details about limiting storage above the bar.

z/OS *Queue manager storage configuration*

The queue manager address space is likely to be the major user of 64-bit storage in an IBM MQ installation. Each connection to the queue manager requires common storage to be allocated as described in the following text. In addition to 64-bit storage, you should allow the queue manager to use all available 31-bit storage by specifying REGION=0M on the queue manager JCL.

Common storage

Each IBM MQ for z/OS subsystem has the following approximate storage requirements:

- CSA 4KB
- ECSA 800KB, plus the size of the trace table that is specified in the **TRACTBL** parameter of the CSQ6SYSP system parameter macro. For more information, see Using CSQ6SYSP.

In addition, each concurrent logical connection to the queue manager requires about 5 KB of ECSA. When a task ends, other IBM MQ tasks can reuse this storage.

IBM MQ does not release the storage until the queue manager is shut down, so you can calculate the maximum amount of ECSA required by multiplying the maximum number of concurrent connections by 5KB. The number of concurrent logical connections is the sum of the number of:

- Tasks (TCBs) in Batch, TSO, z/OS UNIX System Services, IMS, and Db2 stored procedure address space (SPAS) regions that are connected to IBM MQ, but not disconnected.
- CICS transactions that have issued an IBM MQ request, but have not terminated
- JMS Connections, Sessions, TopicSessions or QueueSessions that have been created (for bindings connection), but not yet destroyed or garbage collected.
- Active IBM MQ channels

You can set a limit to the common storage, used by logical connections to the queue manager, with the **ACELIM** configuration parameter. The **ACELIM** control is primarily of interest to sites where Db2 stored procedures cause operations on IBM MQ queues.

When driven from a stored procedure, each IBM MQ operation can result in a new logical connection to the queue manager. Large Db2 units of work, for example due to table load, can result in an excessive demand for common storage.

ACELIM is intended to limit common storage use and to protect the z/OS system, by limiting the number of connections in the system. You should only set **ACELIM** on queue managers that have been identified as using excessive quantities of ECSA storage. See the **ACELIM** section in *Using CSQ6SYSP* for more information.

To set a value for **ACELIM**, firstly determine the amount of storage currently in the subpool controlled by the **ACELIM** value. This information is in the SMF 115 subtype 5 records produced by statistics CLASS(3) trace.

IBM MQ SMF data can be formatted using SupportPac MP1B. The number of bytes in use in the subpool controlled by **ACELIM** is displayed in the STGPOOL DD, on the line titled *ACE/PEB*.

For more information about SMF 115 statistics records, see Interpreting IBM MQ for z/OS performance statistics.

Increase the normal value by a sufficient margin to provide space for growth and workload spikes. Divide the new value by 1024 to yield a maximum storage size in KB for use in the **ACELIM** configuration.

Private storage

The queue manager address space uses 64-bit storage for many internal control blocks. The **MEMLIMIT** parameter of the queue manager JCL defines the maximum amount of 64-bit storage available. 3GB of storage, `MEMLIMIT=3G`, is the minimum you should use, however, depending on your configuration significantly more might be required.

You should specify a specific **MEMLIMIT** value rather than `MEMLIMIT=NOLIMIT` to prevent potential problems. If you specify `NOLIMIT` or a very large value, then there is the potential to use up all of the available z/OS virtual storage, which leads to paging in your system. When increasing the value of **MEMLIMIT** you should discuss the new setting with your z/OS system programmer in case there is a system-wide limit on the amount of on storage that can be used.

If you have a large value for **MEMLIMIT** you might need to increase the size of your dump data sets as more data is captured in a dump.

You can monitor the address space storage usage from the CSQY220I message that indicates the amount of 31 and 64-bit private storage in use, and the remaining free amount.

Buffer pools

Buffer pools are a significant user of private storage in the queue manager address space. Each buffer pool size is determined at queue manager initialization time, and storage is allocated for the buffer pool when a page set that is using that buffer pool is connected. The parameter **LOCATION (ABOVE|BELOW)** is used to specify where the buffers are allocated. You can use the ALTER BUFFPOOL command to dynamically change the size of buffer pools.

When calculating a value for **MEMLIMIT** it is critical that you take into account the buffer pool sizes if they are configured with **LOCATION(ABOVE)**. You should perform the calculation as follows.

Calculate the value of **MEMLIMIT** as 2GB plus the size of the buffer pools configured with **LOCATION(ABOVE)**, rounded up to the nearest GB. Set MEMLIMIT to a minimum of 3GB and increase this as necessary when you need to increase the size of your buffer pools.

For example, for three buffer pools configured with **LOCATION(ABOVE)**, buffer pool one has 10,000 buffers, and buffer pools two and three have 50,000 buffers each. Memory usage above the bar equals 110,000 (total number of buffers) * 4096 = 450,560,000 bytes = 430MB.

All buffer pools regardless of **LOCATION** make use of 64-bit storage for control structures. As the number of buffer pools and number of buffers in those pools increase this can become significant. Each buffer requires around an additional 200 bytes of 64-bit storage. For the preceding configuration that would require: 200 * 110,000 = 22,000,000 bytes = 21MB.

Therefore, in this scenario 3GB can be used for the **MEMLIMIT**, which allows scope for growth: 21MB + 430MB + 2GB which rounds up to 3GB.

For some configurations there can be significant performance benefits to using buffer pools that have their buffers permanently backed by real storage. You can achieve this by specifying the FIXED4KB value for the **PAGECLAS** attribute of the buffer pool. However, you should only do this if there is sufficient real storage available on the LPAR, otherwise other address spaces might be affected. For information about when you should use the FIXED4KB value for **PAGECLAS**, see IBM MQ Support Pac MP16: IBM MQ for z/OS - Capacity planning & tuning.

Making the buffer pools so large that there is MVS™ paging might adversely affect performance. You might consider using a smaller buffer pool that does not page, with IBM MQ moving the message to and from the page set.

Indexed queues

On z/OS, local queues are indexed if the queue has an **INDXTYPE** attribute that has not been set to NONE. The indexes for shared queues are held in a coupling facility, but for private queues the index is held in 64 bit storage. For each message on an indexed queue 136 bytes of data are used to index the message. For very deep queues this can result in a significant amount of 64 bit storage being allocated. For example, 10 million messages on an indexed queue will use 1.27 GB of 64 bit storage in order to maintain the index.

If you expect to have a large number of messages on indexed queues you should allow for this when setting **MEMLIMIT**. To calculate an upper limit for the amount of storage required for indexes, multiply the **MAXDEPTH** attribute for each indexed queue by 136 and sum the value. This value should be added to your existing **MEMLIMIT**.

V 9.4.0 RECOVER CFSTRUCT

From IBM MQ 9.4.0 the **RECOVER CFSTRUCT** command makes greater use of 64-bit storage. In many cases there should be spare 64-bit storage available and so use of the command does not require an increase in the value of **MEMLIMIT**. However, if you are likely to have large structure backups, containing more than a few million messages, you should increase the **MEMLIMIT** for all queue managers which might process the **RECOVER CFSTRUCT** command by 500MB.

For example if you had MEMLIMIT=3G already, you should consider using MEMLIMIT=4G as the **MEMLIMIT** parameter does not allow for decimal points.

Shared Message Data Set (SMDS) buffers and MEMLIMIT

When running messaging workloads using shared message data sets, there are two levels of optimizations that can be achieved by adjusting the DSBUFS and DSBLOCK attributes.

The amount of above bar queue manager storage used by the SMDS buffer is DSBUFS x DSBLOCK. This means that by default, 100 x 256KB (25MB) is used for each CFLEVEL(5) structure in the queue manager.

Although this value is not too high, if your enterprise, or enterprises have many CFSTRUCTs, some of them might allocate a high value of MEMLIMIT for buffer pools, and sometimes they have deep indexed queues, so in total, they might run out of storage above the bar.

V 9.4.0 z/OS *Channel initiator storage configuration from IBM MQ 9.4.0*

The channel initiator typically uses much less 64-bit storage than the queue manager. However, from IBM MQ 9.4.0 the usage has increased. In addition to 64-bit storage, you should allow the channel initiator to use all available 31-bit storage by specifying REGION=0M on the queue manager JCL.

Common storage

The channel initiator typically requires ECSA usage of up to 160KB.

31-bit private storage

The 31-bit storage available to the channel initiator limits the number of concurrent connections the CHINIT can have.

Every channel uses approximately 170KB of extended private region in the channel initiator address space. For message channels, for example, sender or receiver channels, storage is increased by message size if messages larger than 32KB are transmitted. This increased storage is freed when:

- A sending or client channel requires less than half the current buffer size for 10 consecutive messages.
- A heartbeat is sent or received.

The storage is freed for reuse within the Language Environment, however, the storage is not seen as free by the z/OS virtual storage manager. This means that the upper limit for the number of channels is dependent on message size and arrival patterns, and on limitations of individual user systems on extended private region size.

The upper limit on the number of channels is likely to be approximately 9000 on many systems because the extended region size is unlikely to exceed 1.6GB.

The channel initiator trace is written to a data space. The size of the data space storage, is controlled by the **TRAXTBL** parameter. See ALTER QMGR.

64-bit private storage

The MEMLIMIT parameter of the channel initiator JCL defines the maximum amount of 64-bit storage available. 2 GB of storage, MEMLIMIT=2 GB, is the minimum value you should use. Depending on your configuration significantly more might be required.

You should specify a sensible MEMLIMIT value rather than MEMLIMIT=NOLIMIT to prevent potential problems. If you specify NOLIMIT or a very large value, then there is the potential to use up all of the available z/OS virtual storage, leading to paging in your system. When increasing the value of MEMLIMIT you should discuss the new setting with your z/OS system programmer in case there is a system-wide limit on the amount of on storage that can be used.

If you have a large value for MEMLIMIT you might need to increase the size of your dump data sets as more data is captured in a dump.

There are two users of 64-bit storage in the channel initiator: SMF and server-connection channels.

SMF

If enabled, SMF class 4 accounting, or statistics, require 64-bit storage. A minimum of 256MB storage is required. If sufficient storage is not available, the channel initiator issues the CSQX124E message and class 4 accounting and statistics are not available.

Server-connection channels

From IBM MQ 9.4.0 server-connection channels allocate message buffers in 64-bit storage, if they are transferring messages larger than 32 KB in size.

These buffers are freed if the channels require less than half the current buffer size for 10 consecutive messages, or a heartbeat is sent or received.

The value of MEMLIMIT sets an upper limit on how many concurrent server-connection channels can run. You should use a minimum value of MEMLIMIT=2G to ensure that the same number of channels can run as in earlier versions of IBM MQ, as well as providing some capacity for growth.

You can calculate an approximate value for MEMLIMIT by working out the peak maximum number of concurrently active server-connection channels, and for those channels the maximum message size you expect them to transfer. You should use MEMLIMIT=2GB as a starting point and round up.

For example, if you set the maximum number of concurrent server-connection channels to be 2,000 and each channel to have a maximum message size of 1MB, then server-connection channels are using a maximum of just under 2GB of 64-bit storage. As this is very close to 2GB then you should round up to MEMLIMIT=3G.

z/OS *Disk storage*

Use this topic when planning your disk storage requirements for log data sets, Db2 storage, coupling facility storage, and page data sets.

Work with your storage administrator to determine where to put the queue manager data sets. For example, your storage administrator may give you specific DASD volumes, or SMS storage classes, data classes, and management classes for the different data set types.

- Log data sets must be on DASD. These logs can have high I/O activity with a small response time and do not need to be backed up.
- Archive logs can be on DASD or tape. After they have been created, they might never be read again except in an abnormal situation, such as recovering a page set from a backup. They should have a long retention date.
- Page sets might have low to medium activity and should be backed up regularly. On a high use system, they should be backed up twice a day.
- BSDS data sets should be backed up daily; they do not have high I/O activity.

All data sets are similar to those used by Db2, and similar maintenance procedures can be used for IBM MQ.

See the following sections for details of how to plan your data storage:

- **Logs and archive storage**

  "How long do I need to keep archive logs" on page 154 describes how to determine how much storage your active log and archive data sets require, depending on the volume of messages that your IBM MQ system handles and how often the active logs are offloaded to your archive data sets.

- **Db2 storage**

  "Db2 storage" on page 171 describes how to determine how much storage Db2 requires for the IBM MQ data.

- **coupling facility storage**

  "Defining coupling facility resources" on page 162 describes how to determine how large to make your coupling facility structures.

- **Page set and message storage**

  "Planning your page sets and buffer pools" on page 137 describes how to determine how much storage your page data sets require, depending on the sizes of the messages that your applications exchange, on the numbers of these messages, and on the rate at which they are created or exchanged.

z/OS *Where to find more information about storage and performance requirements*

Use this topic as a reference to find more information about storage and performance requirements.

You can find more information from the following sources:

Table 20. Where to find more information about storage requirements

Topic	Where to look
System parameters	Using CSQ6SYSP and Customizing your queue managers
Storage required to install IBM MQ	Program Directory. 프로그램 디렉토리에 대한 다운로드 링크는 다음을 참조하세요.IBM MQ for z/OS 프로그램 디렉토리 PDF 파일 .
IEALIMIT and IEFUSI exits	See IEALIMIT and IEFUSI in the *z/OS:MVS Installation Exits* documentation.
Latest information	IBM MQ SupportPac website IBM MQ 및 기타 프로젝트 영역의 경우 SupportPacs.
Workload management and defining goals through the service definition	*z/OS MVS Planning: Workload Management*

z/OS Planning your page sets and buffer pools

Information to help you with planning the initial number, and sizes of your page data sets, and buffer pools.

This topic contains the following sections:

Plan your page sets

Page set usage

For short-lived messages, few pages are normally used on the page set and there is little or no I/O to the data sets except at startup, during a checkpoint, or at shutdown.

For long-lived messages, those pages containing messages are normally written out to disk. This operation is performed by the queue manager in order to reduce restart time.

Separate short-lived messages from long-lived messages by placing them on different page sets and in different buffer pools.

Number of page sets

Using several large page sets can make the role of the IBM MQ administrator easier because it means that you need fewer page sets, making the mapping of queues to page sets simpler.

Using multiple, smaller page sets has a number of advantages. For example, they take less time to back up, and I/O can be carried out in parallel during backup and restart. However, consider that this adds a significant performance cost to the role of the IBM MQ administrator, who is required to map each queue to one of a much greater number of page sets.

Define at least five page sets, as follows:

- A page set reserved for object definitions (page set zero)
- A page set for system-related messages
- A page set for performance-critical long-lived messages
- A page set for performance-critical short-lived messages
- A page set for all other messages

"Defining your buffer pools" on page 142 explains the performance advantages of distributing your messages on page sets in this way.

Size of page sets

Define sufficient space in your page sets for the expected peak message capacity. Consider for any unexpected peak capacity, such as when a build-up of messages develops because a queue server program is not running. You can do this by allocating the page set with secondary extents or, alternatively, by enabling dynamic page set expansion. For more information, see "Enabling dynamic page set expansion" on page 140. It is difficult to make a page set smaller, so it is often better to allocate a smaller page set, and allow it to expand when needed.

When planning page set sizes, consider all messages that might be generated, including non-application message data. For example, trigger messages, event messages and any report messages that your application has requested.

The size of the page set determines the time taken to recover a page set when restoring from a backup, because a large page set takes longer to restore.

Note: Recovery of a page set also depends on the time the queue manager takes to process the log records written since the backup was taken; this time period is determined by the backup frequency. For more information, see "Planning for backup and recovery" on page 173.

Note: Page sets larger than 4 GB require the use of SMS extended addressability.

Planning for z/OS data set encryption

You can apply the z/OS data set encryption feature to page sets for queue managers running at IBM MQ for z/OS 9.1.4 or later.

You must allocate these page sets with EXTENDED attributes, and a data set key label that ensures the data is AES encrypted.

See the section, confidentiality for data at rest on IBM MQ for z/OS with data set encryption. for more information.

Calculate the size of your page sets

For queue manager object definitions (for example, queues and processes), it is simple to calculate the storage requirement because these objects are of fixed size and are permanent. For messages however, the calculation is more complex for the following reasons:

- Messages vary in size.
- Messages are transitory.

- Space occupied by messages that have been retrieved is reclaimed periodically by an asynchronous process.

Large page sets of greater than 4 GB that provide extra capacity for messages if the network stops, can be created if required. It is not possible to modify the existing page sets. Instead, new page sets with extended addressability and extended format attributes, must be created. The new page sets must be the same physical size as the old ones, and the old page sets must then be copied to the new ones. If backward migration is required, page set zero must not be changed. If page sets less than 4 GB are adequate, no action is needed.

Page set zero

Page set zero is reserved for object definitions.

For page set zero, the storage required is:

```
   (maximum number of local queue definitions x 1010)
      (excluding shared queues)
+  (maximum number of model queue definitions x 746)
+  (maximum number of alias queue definitions x 338)
+  (maximum number of remote queue definitions x 434)
+  (maximum number of permanent dynamic queue definitions x 1010)
+  (maximum number of process definitions x 674)
+  (maximum number of namelist definitions x 12320)
+  (maximum number of message channel definitions x 2026)
+  (maximum number of client-connection channel definitions x 5170)
+  (maximum number of server-connection channel definitions x 2026)
+  (maximum number of storage class definitions x 266)
+  (maximum number of authentication information definitions x 1010)
+  (maximum number of administrative topic definitions x 15000)
+  (total length of topic strings defined in administrative topic definitions)
```

Divide this value by 4096 to determine the number of records to specify in the cluster for the page set data set.

You do not need to allow for objects that are stored in the shared repository, but you must allow for objects that are stored or copied to page set zero (objects with a disposition of GROUP or QMGR).

The total number of objects that you can create is limited by the capacity of page set zero. The number of local queues that you can define is limited to 524 287.

Page sets 01 - 99

For page sets 01 - 99, the storage required for each page set is determined by the number and size of the messages stored on that page set. (Messages on shared queues are not stored on page sets.)

Divide this value by 4096 to determine the number of records to specify in the cluster for the page set data set.

Calculating the storage requirement for messages

This section describes how messages are stored on pages. Understanding this can help you calculate how much page set storage you must define for your messages. To calculate the approximate space required for all messages on a page set you must consider maximum queue depth of all the queues that map to the page set and the average size of messages on those queues.

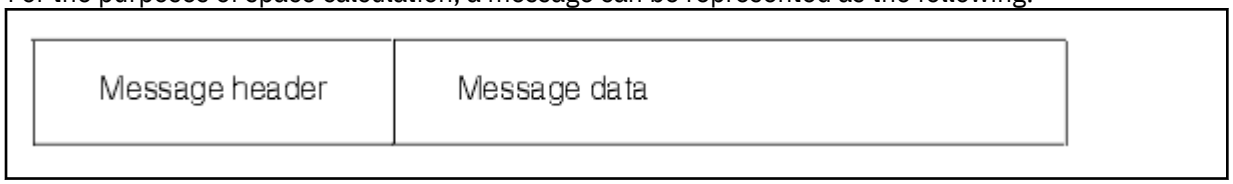
Note: The sizes of the structures and control information given in this section are liable to change between major releases. For details specific to your release of IBM MQ, refer to SupportPac MP16 - z/OS 용 IBM MQ 용량 계획 및 튜닝 and IBM MQ 제품군-성능 보고서

You must allow for the possibility that message "gets" might be delayed for reasons outside the control of IBM MQ (for example, because of a problem with your communications protocol). In this case, the "put" rate of messages might far exceed the "get" rate. This can lead to a large increase in the number of messages stored in the page sets and a consequent increase in the storage size demanded.

Each page in the page set is 4096 bytes long. Allowing for fixed header information, each page has 4057 bytes of space available for storing messages.

When calculating the space required for each message, the first thing you must consider is whether the message fits on one page (a short message) or whether it needs to be split over two or more pages (a long message). When messages are split in this way, you must allow for additional control information in your space calculations.

For the purposes of space calculation, a message can be represented as the following:



The message header section contains the message descriptor and other control information, the size of which varies depending on the size of the message. The message data section contains all the actual message data, and any other headers (for example, the transmission header or the IMS bridge header).

A minimum of two pages are required for page set control information which, is typically less than 1% of the total space required for messages.

Short messages

A short message is defined as a message that fits on one page.

Small messages are stored one on each page.

Long messages

If the size of the message data is greater than 3596 bytes, but not greater than 4 MB, the message is classed as a long message. When presented with a long message, IBM MQ stores the message on a series of pages, and stores control information that points to these pages in the same way that it would store a short message. This is shown in Figure 41 on page 140:

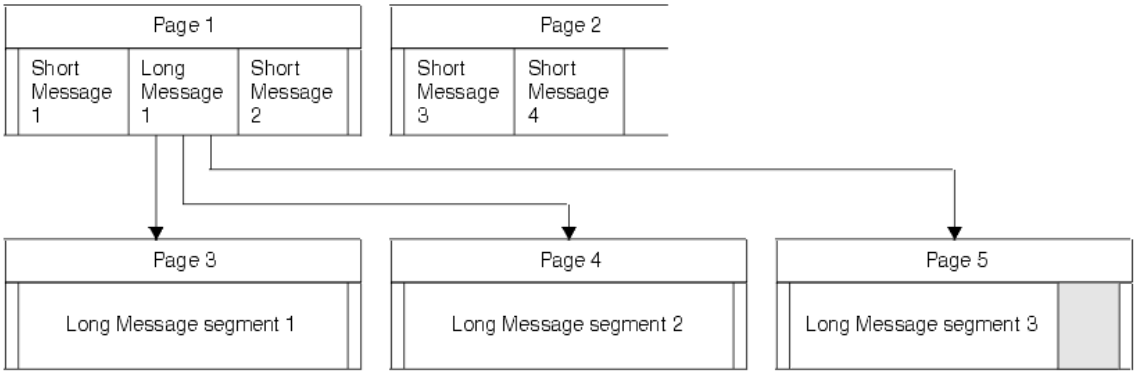


Figure 41. How IBM MQ stores long messages on page sets

Very long messages

Very long messages are messages with a size greater than 4 MB. These are stored so that each 4 MB uses 1037 pages. Any remainder is stored in the same way as a long message, as described above.

z/OS *Enabling dynamic page set expansion*

Page sets can be extended dynamically while the queue manager is running. A page set can have 123 extents, and can be spread over multiple disk volumes.

Each time a page set expands, a new data set extent is used. The queue manager continues to expand a page set when required, until the maximum number of extents has been reached, or until no more storage is available for allocation on eligible volumes.

Once page set expansion fails for one of the reasons above, the queue manager marks the page set for no further expansion attempts. This marking can be reset by altering the page set to EXPAND(SYSTEM).

Page set expansion takes place asynchronously to all other page set activity, when 90% of the existing space in the page set is allocated.

The page set expansion process formats the newly allocated extent and makes it available for use by the queue manager. However, none of the space is available for use, until the entire extent has been formatted. This means that expansion by a large extent is likely to take some time, and putting applications might 'block' if they fill the remaining 10% of the page set before the expansion has completed.

Sample thlqual.SCSQPROC(CSQ4PAGE) shows how to define the secondary extents.

To control the size of new extents, you use one of the following options of the EXPAND keyword of the DEFINE PSID and ALTER PSID commands:

- USER
- SYSTEM
- NONE

USER

Uses the secondary extent size specified when the page set was allocated. If a value was not specified, or if a value of zero was specified, dynamic page set expansion cannot occur.

Page set expansion occurs when the space in the page is 90% used, and is performed asynchronously with other page set activity.

This may lead to expansion by more than a single extent at a time.

Consider the following example: you allocate a page set with a primary extent of 100,000 pages and a secondary extent of 5000 pages. A message is put that requires 9999 pages. If the page set is already using 85,000 pages, writing the message crosses the 90% full boundary (90,000 pages). At this point, a further secondary extent is allocated to the primary extent of 100,000 pages, taking the page set size to 105,000 pages. The remaining 4999 pages of the message continue to be written. When the used page space reaches 94,500 pages, which is 90% of the updated page set size of 105,000 pages, another 5000 page extent is allocated, taking the page set size to 110,000 pages. At the end of the MQPUT, the page set has expanded twice, and 94,500 pages are used. None of the pages in the second page set expansion have been used, although they were allocated.

At restart, if a previously used page set has been replaced with a data set that is smaller, it is expanded until it reaches the size of the previously used data set. Only one extent is required to reach this size.

SYSTEM

Ignores the secondary extent size that was specified when the page set was defined. Instead, the queue manager sets a value that is approximately 10% of the current page set size. The value is rounded up to the nearest cylinder of DASD.

If a value was not specified, or if a value of zero was specified, dynamic page set expansion can still occur. The queue manager sets a value that is approximately 10% of the current page set size. The new value is rounded up depending on the characteristics of the DASD.

Page set expansion occurs when the space in the page set is approximately 90% used, and is performed asynchronously with other page set activity.

At restart, if a previously used page set has been replaced with a data set that is smaller, it is expanded until it reaches the size of the previously used data set.

NONE

No further page set expansion is to take place.

Related reference
ALTER PSID
DEFINE PSID
DISPLAYUSAGE

Defining your buffer pools

Use this topic to help plan the number of buffer pools you should define, and their settings.

This topic is divided into the following sections:

1. "Decide on the number of buffer pools to define" on page 142
2. "Decide on the settings for each buffer pool" on page 143
3. "Monitor the performance of buffer pools under expected load" on page 143
4. "Adjust buffer pool characteristics" on page 143

Decide on the number of buffer pools to define

You should define four buffer pools initially:

Buffer pool 0

Use for object definitions (in page set zero) and performance critical, system related message queues, such as the SYSTEM.CHANNEL.SYNCQ queue and the SYSTEM.CLUSTER.COMMAND.QUEUE and SYSTEM.CLUSTER.REPOSITORY.QUEUE queues.

However it is important to consider point "7" on page 144 in *Adjust buffer pool characteristics* if a large number of channels, or clustering, is to be used.

Use the remaining three buffer pools for user messages.

Buffer pool 1

Use for important long-lived messages.

Long-lived messages are those that remain in the system for longer than two checkpoints, at which time they are written out to the page set. If you have many long-lived messages, this buffer pool should be relatively small, so that page set I/O is evenly distributed (older messages are written out to DASD each time the buffer pool becomes 85% full).

If the buffer pool is too large, and the buffer pool never gets to 85% full, page set I/O is delayed until checkpoint processing. This might affect response times throughout the system.

If you expect few long-lived messages only, define this buffer pool so that it is sufficiently large to hold all these messages.

Buffer pool 2

Use for performance-critical, short-lived messages.

There is normally a high degree of buffer reuse, using few buffers. However, you should make this buffer pool large to allow for unexpected message accumulation, for example, when a server application fails.

Buffer pool 3

Use for all other (typically, performance noncritical) messages.

Queues such as the dead-letter queue, SYSTEM.COMMAND.* queues and SYSTEM.ADMIN.* queues can also be mapped to buffer pool 3.

Where virtual storage constraints exist, and buffer pools need to be smaller, buffer pool 3 is the first candidate for size reduction.

You might need to define additional buffer pools in the following circumstances:

- If a particular queue is known to require isolation, perhaps because it exhibits different behavior at various times.
  - Such a queue might either require the best performance possible under the varying circumstances, or need to be isolated so that it does not adversely affect the other queues in a buffer pool.
  - Each such queue can be isolated into its own buffer pool and page set.
- You want to isolate different sets of queues from each other for class-of-service reasons.
  - Each set of queues might then require one, or both, of the two types of buffer pools 1 or 2, as described in Suggested definitions for buffer pool settings, necessitating creation of several buffer pools of a specific type.

Decide on the settings for each buffer pool

If you are using the four buffer pools described in “Decide on the number of buffer pools to define” on page 142, then Suggested definitions for buffer pool settings gives two sets of values for the size of the buffer pools.

The first set is suitable for a test system, the other for a production system or a system that will become a production system eventually. In all cases define your buffer pools with the **LOCATION**(*ABOVE*) attribute

Table 21. Suggested definitions for buffer pool settings

Definition setting	Test system	Production system
BUFFPOOL 0	1 050 buffers	50 000 buffers
BUFFPOOL 1	1 050 buffers	20 000 buffers
BUFFPOOL 2	1 050 buffers	50 000 buffers
BUFFPOOL 3	1 050 buffers	20 000 buffers

If you need more than the four suggested buffer pools, select the buffer pool (1 or 2) that most accurately describes the expected behavior of the queues in the buffer pool, and size it using the information in Suggested definitions for buffer pool settings.

Ensure that your MEMLIMIT is set high enough, so that all the buffer pools can be located above the bar.

Monitor the performance of buffer pools under expected load

You can monitor the usage of buffer pools by analyzing buffer pool performance statistics. In particular, you should ensure that the buffer pools are large enough so that the values of QPSTSOS, QPSTSTLA, and QPSTDMC remain at zero.

For further information, see Buffer manager data records.

Adjust buffer pool characteristics

Use the following points to adjust the buffer pool settings from “Decide on the settings for each buffer pool” on page 143, if required.

Use the performance statistics from “Monitor the performance of buffer pools under expected load” on page 143 as guidance.

1. If you are migrating from an earlier version of IBM MQ, only change your existing settings if you have more real storage available.
2. In general, bigger buffer pools are better for performance, and buffer pools can be much bigger if they are above the bar.

   However, at all times you should have sufficient real storage available so that the buffer pools are resident in real storage. It is better to have smaller buffer pools that do not result in paging, than big ones that do.

Additionally, there is no point having a buffer pool that is bigger than the total size of the page sets that use it, although you should take into account page set expansion if it is likely to occur.

3. Aim for one page set per buffer pool, as this provides better application isolation.
4. If you have sufficient real storage, such that your buffer pools will never be paged out by the operating system, consider using page-fixed buffers in your buffer pool.

   This is particularly important if the buffer pool is likely to undergo much I/O, as it saves the CPU cost associated with page-fixing the buffers before the I/O, and page-unfixing them afterwards.
5. There are several benefits to locating buffer pools above the bar even if they are small enough to fit below the bar. These are:
   - 31 bit virtual storage constraint relief - for example more space for common storage.
   - If the size of a buffer pool needs to be increased unexpectedly while it is being heavily used, there is less impact and risk to the queue manager, and its workload, by adding more buffers to a buffer pool that is already above the bar, than moving the buffer pool to above the bar and then adding more buffers.
6. Tune buffer pool zero and the buffer pool for short-lived messages (buffer pool 2) so that the 15% free threshold is never exceeded (that is, QPSTCBSL divided by QPSTNBUF is always greater than 15%). If more than 15% of buffers remain free, I/O to the page sets using these buffer pools can be largely avoided during normal operation, although messages older than two checkpoints are written to page sets.

   Attention: The optimum value for these parameters is dependent on the characteristics of the individual system. The values given are intended only as a guideline and might not be appropriate for your system.
7. SYSTEM.* queues which get very deep, for example SYSTEM.CHANNEL.SYNCQ, might benefit from being placed in their own buffer pool, if sufficient storage is available.

IBM MQ SupportPac MP16 - z/OS 용 IBM MQ 용량 계획 및 튜닝 provides further information about tuning buffer pools.

z/OS Planning your logging environment

Use this topic to plan the number, size and placement of the logs, and log archives used by IBM MQ.

Logs are used to:

- Write recovery information about persistent messages
- Record information about units of work using persistent messages
- Record information about changes to objects, such as define queue
- Backup CF structures

and for other internal information.

The IBM MQ logging environment is established using the system parameter macros to specify options, such as: whether to have single or dual active logs, what media to use for the archive log volumes, and how many log buffers to have.

These macros are described in Create the bootstrap and log data sets and Tailor your system parameter module.

Note: If you are using queue sharing groups, ensure that you define the bootstrap and log data sets with SHAREOPTIONS(2 3).

This section contains information about the following topics:

z/OS *Log data set definitions*

Use this topic to decide on the most appropriate configuration for your log data sets.

This topic contains information to help you answer the following questions:

- Should your installation use single or dual logging?
- How many active log data sets do you need?
- "How large should the active logs be?" on page 146
- Active log placement
- "Active log encryption with z/OS data set encryption" on page 147

Should your installation use single or dual logging?

In general you should use dual logging for production, to minimize the risk of losing data. If you want your test system to reflect production, both should use dual logging, otherwise your test systems can use single logging.

With single logging data is written to one set of log data sets. With dual logging data is written to two sets of log data sets, so in the event of a problem with one log data set, such as the data set being accidentally deleted, the equivalent data set in the other set of logs can be used to recover the data.

With dual logging you require twice as much DASD as with single logging.

If you are using dual logging, then also use dual BSDSs and dual archiving to ensure adequate provision for data recovery.

Dual active logging adds a small performance cost.

Attention: Use of disk mirroring technologies, such as Metro Mirror, are not necessarily a replacement for dual logging and dual BSDS. If a mirrored data set is accidentally deleted, both copies are lost.

If you use persistent messages, single logging can increase maximum capacity by 10-30% and can also improve response times.

Single logging uses 2 - 310 active log data sets, whereas dual logging uses 4 - 620 active log data sets to provide the same number of active logs. Thus single logging reduces the amount of data logged, which might be important if your installation is I/O constrained.

How many active log data sets do you need?

The number of logs depends on the activities of your queue manager. For a test system with low throughput, three active log data sets might be suitable. For a high throughput production system you might want the maximum number of logs available, so, if there is a problem with offloading logs you have more time to resolve the problems.

You must have at least three active log data sets, but it is preferable to define more. For example, if the time taken to fill a log is likely to approach the time taken to archive a log during peak load, define more logs.

Note: Page sets and active log data sets are eligible to reside in the extended addressing space (EAS) part of an extended address volumes (EAV) and an archive log dataset can also reside in the EAS.

You should also define more logs to offset possible delays in log archiving. If you use archive logs on tape, allow for the time required to mount the tape.

Consider having enough active log space to keep a day's worth of data, in case the system is unable to archive because of lack of DASD or because it cannot write to tape. If all the active logs fill up, then IBM MQ is unable to process persistent messages or transactions. It is very important to have enough active log space.

It is possible to dynamically define new active log data sets as a way of minimizing the effect of archive delays or problems. New data sets can be brought online rapidly, using the **DEFINE LOG** command to avoid queue manager 'stall' due to lack of space in the active log.

If you want to define more than 31 active log data sets, you must configure your logging environment to use a version 2 format BSDS. Once a version 2 format BSDS is in use, up to 310 active log data sets can be defined for each log copy ring. See "Planning to increase the maximum addressable log range" on page 156 for information on how you convert to a version 2 format BSDS.

You can tell whether your queue manager is using a version 2 or higher BSDS, either by running the print log map utility (CSQJU004), or from the CSQJ034I message issued during queue manager initialization. An end of log RBA range of FFFFFFFFFFFFFFFF, in the CSQJ034I message, indicates that a version 2, or higher, format BSDS is in use. An end of log RBA range of 0000FFFFFFFFFFFF, in the CSQJ034I message, indicates that a version 1 format BSDS is in use.

When a queue manager is using a version 2, or higher, format BSDS, it is possible to use the **DEFINE LOG** command to dynamically add more than 31 active log data sets to a log copy ring.

How large should the active logs be?

The maximum supported active log size, when archiving to disk or to tape, is 4 GB.

You should create active logs of at least 1 GB in size for production and test systems.

Important: You need to be careful when allocating data sets, because IDCAMS rounds up the size you allocate.

To allocate a 3 GB log specify one of the following options:

- Cylinders(4369)
- Megabytes(3071)
- TRACKS(65535)
- RECORD(786420)

Any one of these allocates 2.99995 GB.

To allocate a 4GB log specify one of the following options:

- Cylinders(5825)
- Megabytes(4095)
- TRACKS(87375)
- RECORD(1048500)

Any one of these allocates 3.9997 GB.

When using striped data sets, where the data set is spread across multiple volumes, the specified size value is allocated on each DASD volume used for striping. So, if you want to use 4 GB logs and four volumes for striping, you should specify:

- CYLinders(1456)
- Megabytes(1023)

Setting these attributes allocates 4*1456 = 5824 Cylinders or 4 * 1023 = 4092 Megabytes.

Note: Striping is supported when using extended format data sets. This is usually set by the storage manager.

See Increasing the size of the active log for information on carrying out the procedure.

Active log placement

You should work with your storage management team to set up storage pools for the queue managers. You need to consider:

- A naming convention, so the queue managers use the correct SMS definitions.

- Space required for active and archive logs. Your storage pool should have enough space for the active logs from a whole day.
- Performance and resilience to failures.

For performance reasons you should consider striping your active log data sets. The I/O is spread across multiple volumes and reduces the I/O response times, leading to higher throughput. See the preceding text for information about allocating the size of the active logs when using striping.

You should review the I/O statistics using reports from RMF or a similar product. Perform the review of these statistics monthly (or more frequently) for the IBM MQ data sets, to ensure there are no delays due to the location of the data sets.

In some situations, there can be much IBM MQ page set I/O, and this can impact the IBM MQ log performance if they are located on the same DASD.

If you use dual logging, ensure that each set of active and archive logs is kept apart. For example, allocate them on separate DASD subsystems, or on different devices.

This reduces the risk of them both being lost if one of the volumes is corrupted or destroyed. If both copies of the log are lost, the probability of data loss is high.

When you create a new active log data, set you should preformat it using CSQJUFMT. If the log is not preformatted, the queue manager formats the log the first time it is used, which impacts the performance.

With older DASD with large spinning disks, you had to be careful which volumes were used to get the best performance.

With modern DASD, where data is spread over many PC sized disks, you do not need to worry so much about which volumes are used.

Your storage manager should be checking the enterprise DASD to review and resolve any performance problems. For availability, you might want to use one set of logs on one DASD subsystem, and the dual logs on a different DASD subsystem.

Active log encryption with z/OS data set encryption

You can apply the z/OS data set encryption feature to active log data sets for queue managers running at IBM MQ for z/OS 9.1.4 or later.

You must allocate these active log data sets with EXTENDED attributes, and a data set key label that ensures the data is AES encrypted.

See the section, confidentiality for data at rest on IBM MQ for z/OS with data set encryption. for more information.

z/OS *Using MetroMirror with IBM MQ*

IBM Metro Mirror, previously known as Synchronous Peer to Peer Remote Copy (PPRC), is a synchronous replication solution between two storage subsystems, where write operations are completed on both the primary and secondary volumes before the write operation is considered to be complete. Metro Mirror can be used in environments that require no data loss in the event of a storage subsystem failure.

Supported data set types

All of the following IBM MQ data set types can be replicated using Metro Mirror. However, exactly which ones are replicated depends on the availability requirements of your enterprise:

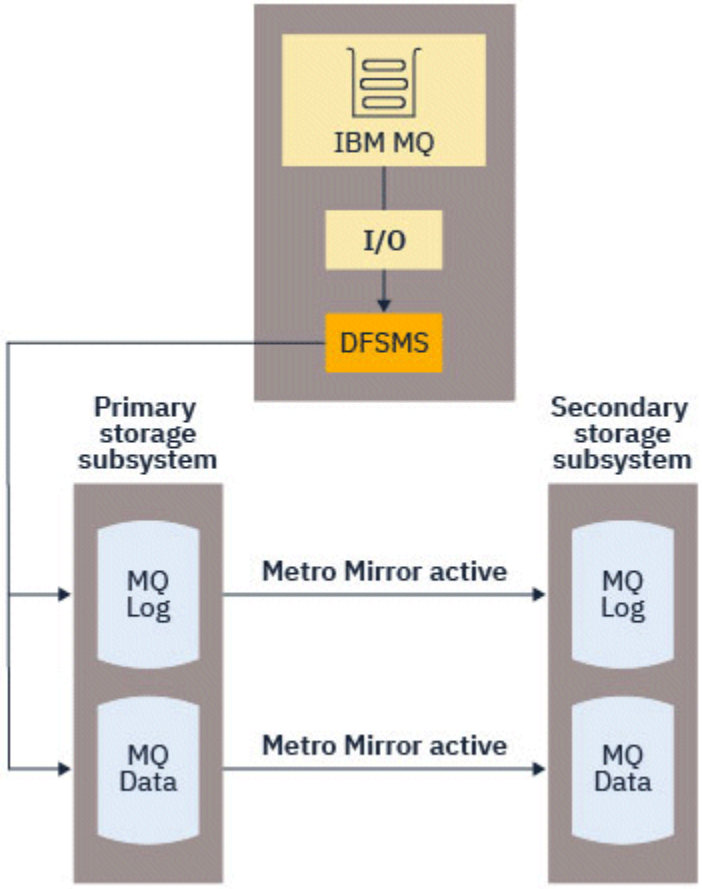
- Active logs
- Archive logs
- Bootstrap data set (BSDS)
- Page sets
- Shared message data set (SMDS)

- Data sets used for configuration, for example, in the CSQINP* DD cards on the MSTR JCL

Using zHyperWrite with IBM MQ active logs

When a write is made to a data set that is replicated using Metro Mirror, the write is first made to the primary volume, and then replicated to the secondary volume. This replication is done by the storage subsystem and is transparent to the application that issued the write, for example IBM MQ.

This process is illustrated in the following diagram.

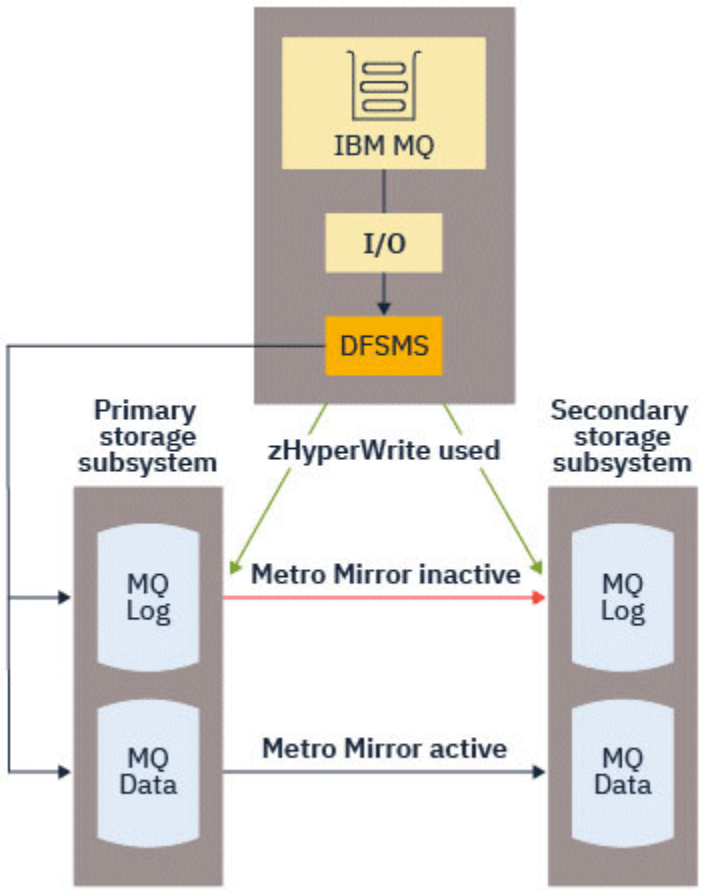


Because both writes to the primary and secondary storage subsystems need to complete before the write returns to IBM MQ, use of Metro Mirror can have a performance impact. You need to balance this performance impact against the availability benefits of using Metro Mirror.

The IBM MQ active logs are most sensitive to the performance impact of using Metro Mirror. IBM MQ allows use of zHyperWrite with the active logs to help reduce this performance impact.

zHyperWrite is a storage subsystem technology that works with z/OS to reduce the performance impact of writes made to data sets that are replicated using Metro Mirror. When zHyperWrite is used, the write to the primary and secondary volumes are issued in parallel at the Data Facility Storage Management Subsystem (DFSMS) level, instead of sequentially at the storage subsystem level, thereby reducing the performance impact.

The following diagram illustrates zHyperWrite being used for the active logs, and Metro Mirror being used for the other IBM MQ data set types. Note that if a zHyperWrite write fails, DFSMS will transparently reissue the write using Metro Mirror.



zHyperWrite on IBM MQ, is supported only on the active log data sets.

In order to use zHyperWrite with the active logs, you need to:

- Configure IBM MQ to use zHyperWrite, and
- The active logs need to be on zHyperWrite capable volumes

You can configure IBM MQ to use zHyperWrite by using one of the following methods:

- Specify ZHYWRITE(YES) in the system parameter module.
- Issue the command SET LOG ZHYWRITE(YES).

Set the following conditions for active log data sets to be on zHyperWrite capable volumes:

- Enable the volumes for Metro Mirror, and the volumes support zHyperWrite
- Ensure that the volumes are HyperSwap enabled
- Specify HYPERWRITE=YES in the IECIOSxx parameter

V 9.4.0 Prior to IBM MQ 9.4.0, if all the preceding conditions are met, then writes to the active logs are enabled for zHyperWrite. If one, or more, of these conditions are not met, IBM MQ writes to the active logs as normal, and Metro Mirror replicates the writes if it is configured.

V 9.4.0 From IBM MQ 9.4.0, if ZHYWRITE(YES) is specified, then IBM MQ always attempts to use zHyperWrite when writing to the active logs, regardless of whether the logs are on zHyperWrite capable volumes. If the logs are not on zHyperWrite capable volumes then Metro Mirror replicates the writes if it is configured. There are no negative effects of attempting to use zHyperWrite if the logs are not on zHyperWrite capable volumes

Notes:

- IBM MQ does not require that all active log data sets are on zHyperWrite capable volumes.

If IBM MQ detects that some active log data sets are on zHyperWrite capable volumes, and others are not, it issues message CSQJ166E and carries on processing.

- IBM MQ checks whether active log data sets are zHyperWrite capable when the data sets are first opened.

  Log data sets are opened either at queue manager start up, or when dynamically adding using the DEFINE LOG command. If the log data sets are made zHyperWrite capable while a queue manager has them open, the queue manager will not detect this until it has been restarted.

You can use the output of the DISPLAY LOG command to indicate whether the current active log data sets are zHyperWrite capable. The following example shows that both of the data sets are zHyperWrite capable. If the queue manager has been configured with ZHYWRITE(YES), writes to these logs would be enabled for zHyperWrite:

```
Copy %Full zHyperWrite DSName
 1      4  CAPABLE     MQTST.SUBSYS.MQDL.LOGCOPY1.DS001
 2      4  CAPABLE     MQTST.SUBSYS.MQDL.LOGCOPY2.DS001
```

z/OS V 9.4.0 **zHyperLink를 사용한 빠른 로그 처리량**

zHyper 링크 기술은 CPU와 I/O 장치 간에 빠르고 안정적이며 직접적인 통신 경로를 제공하여 입출력(I/O) 대기 시간을 줄이도록 설계되었습니다.

개요zHyper 링크

zHyperLink는 활성 로그 처리량을 개선하고 IBM MQ 트랜잭션 시간을 최대 3.5 배까지 줄일 수 있습니다. 이 목표는 설치를 통해 달성됩니다.zHyper 링크 어댑터z/OS 호스트, 선택IBM 스토리지 하드웨어를 사용하여 연결하고zHyper 케이블을 연결하세요. 이는 CPU와 I/O 장치 사이에 지점 간 연결을 생성하여 I/O 응답 시간을 최대 10 배까지 단축시킵니다.IBM z 고성능 FICON®(zHPF). 이러한 낮은 응답 시간은 동기 I/O 요청을 사용하여 달성됩니다.

비동기 I/O에 비해 동기 I/O의 장점

그만큼IBM MQ 로거 작업은 로그에 기록되어야 하는 다음 데이터 조각을 기다리는 루프로 구성됩니다. 해당 데이터를 사용할 수 있게 되면 로거는 쓰기를 예약하고 완료될 때까지 기다린 후 다음 데이터 조각으로 이동합니다.

기존 I/O는 CPU보다 느리기 때문에 I/O를 비동기식으로 수행하여 다른 작업을 위해 CPU를 확보하는 것이 가장 효율적입니다. 따라서 기존 비동기 I/O에서는 쓰기가 완료될 때까지 로거 작업을 일시 중지해야 합니다. 쓰기가 완료되면 로거 작업은 CPU를 사용할 수 있을 때까지 기다려야 하며 짧은 재디스패치 지연과 CPU 캐시 다시 채우기로 인한 지연이 추가됩니다.

zHyper Link는 CPU 속도에 더 가까운 훨씬 빠른 I/O 시간을 제공합니다.zHyper 링크, I/O는 동기식으로 수행될 수 있습니다. 즉, 쓰기 작업 중에 로거 작업이 일시 중지되지 않아 재디스패치 및 캐시 관련 지연이 제거됩니다.

쓰기가 진행되는 동안 로거 작업은 여전히 CPU를 적극적으로 사용하므로 기존 I/O에 비해 CPU 사용량이 늘어납니다.

큐 관리자가 사용을 시도하는 경우zHyper 링크와,zHyper 예를 들어 구성 문제로 인해 링크 쓰기가 실패하면 큐 관리자는 투명하게 기존 I/O로 대체됩니다.

최소 하드웨어 요구사항

- IBM z14 이상
- DS8880 이상

필수 소프트웨어

- zHyperLink Express는 z/OS 2.3 이상에서 지원됩니다.
- z/OS 이미지는 IBM z/VM®에서 게스트로 실행되지 않고 LPAR에서 실행되어야 합니다.
- zHyper링크를 사용하려면 IBM z 고성능 FICON (zHPF) 이 사용 가능해야 합니다.

zHyperLink with IBM MQ 활성 로그 사용

사용하기 위해서는zHyper 큐 관리자의 활성 로그와 연결하려면 다음을 수행해야 합니다.

- 구성IBM MQ 사용zHyper 링크하고,
- 활성 로그가 켜져 있는지 확인하세요.zHyper 링크 가능 볼륨.

  보다 시작하기IBMzHyper 링크z/OS 자세한 내용은.

구성할 수 있습니다IBM MQ 사용zHyper 다음 방법 중 하나를 사용하여 연결합니다.

- 로그 매개변수에 ZHYLINK(YES) 를 지정하십시오.
- SET LOG ZHYLINK(YES) 명령을 실행하십시오.

참고:

- zHyper링크를 사용하려면 zHyper쓰기가 켜져 있어야 합니다. 이는 ZHYLINK를 사용하려면 로그 매개변수에서 ZHYWRITE도 켜져야 함을 의미합니다. 큐 관리자에서 ZHYWRITE (NO) 가 설정된 경우에만 ZHYLINK (YES) 를 지정하면 ZHYWRITE 매개변수가 자동으로 YES로 대체됩니다.
- ZHYWRITE (NO) 를 사용하여 ZHYLINK (YES) 를 명시적으로 설정하려고 하면 SET LOG 명령이 비정상적으로 완료됩니다.
- ZPRM에서 ZHYLINK=YES를 설정하면 ZHYWRITE가 YES로 대체됩니다.

문제점이 발생하는 경우 자세한 정보는 zHyper링크 문제점 해결 을 참조하십시오.

IBM MQ 에서는 모든 활성 로그 데이터 세트가 zHyperLink 가능 볼륨에 있을 필요는 없지만 이를 수행하는 것이 좋습니다. IBM MQ 가 일부 활성 로그 데이터 세트가 zHyper링크 가능 볼륨에 있고 다른 데이터 세트는 없음을 발견하는 경우 메시지 CSQJ601E 를 발행하고 처리를 계속합니다.

IBM MQ 는 데이터 세트가 처음 열릴 때 활성 로그 데이터 세트가 zHyper가능한지 여부를 확인합니다. 로그 데이터 세트는 큐 관리자 시작 시 또는 DEFINE LOG 명령을 사용하여 동적으로 추가할 때 열립니다. 큐 관리자가 로그 데이터 세트를 열어 놓은 상태에서 로그 데이터 세트가 zHyper링크를 사용할 수 있게 되면 큐 관리자가 재시작될 때까지 이를 감지하지 않습니다.

ZHYLINK (YES) 가 지정되면 IBM MQ 는 로그가 zHyperLink 가능 볼륨에 있는지 여부에 관계없이 활성 로그에 쓸 때 항상 zHyperLink를 사용하려고 시도합니다. 로그가 zHyperLink 가능 볼륨에 없는 경우 zHyperLink를 사용하려고 시도해도 부정적인 영향을 미치지 않습니다.

DISPLAY LOG 명령의 출력을 사용하여 현재 활성 로그 데이터 세트에 대한 zHyper링크의 상태를 표시할 수 있습니다.

```
Copy %Full zHyperWrite    Encrypted          DSName
  1     81 YES          NO                 MQTST.SUBSYS.MQDL.LOGCOPY1.DS001
  2     81 YES            NO                 MQTST.SUBSYS.MQDL.LOGCOPY2.DS001
Copy zHyperLink
  1  YES
  2  YES
```

zHyper링크 상태는 다음 중 하나입니다.

YES
zHyper링크가 큐 관리자에서 사용 가능하며 모든 쓰기에서 시도됩니다.

NO
zHyper링크가 큐 관리자에서 사용되지 않으며 데이터 세트가 zHyper링크 가능 볼륨에서 **사용되지 않습니다** .

CAPABLE
zHyperLink는큐 관리자에서 사용으로 설정되지 않으며 데이터 세트 **는 zHyperLink 가능 볼륨에서** 입니다.

모니터링 및 이해를 위한 여러 가지 추가 SMF 통계가 있습니다.zHyper 링크 성능; 보다zHyper 링크통계 자세한 내용은.

세션 쓰기

사용할 때zHyper 링크, 하나 이상zHyper 링크 쓰기 세션은 DASD를 통해 설정됩니다. 현재 DASD는 최대 64개의 동시 쓰기 세션을 지원하므로 활성화할 큐 관리자를 신중하게 고려해야 합니다.zHyper 링크 켜짐 및 기타 하위 시스템(예:Db2 또한 사용하고 있습니다zHyper 동일한 DASD에 쓰기 위한 링크입니다. 사용 가능한 쓰기 세션이 부족하면 큐 관리자는 자동으로 기존 비동기 I/O를 사용하도록 다시 전환합니다.

횟수를 계산할 수 있습니다.zHyper 다음과 같이 쓰기 세션을 연결합니다.

```
Number of log copies (either 1 or 2) * number of stripes per log copy * 2
if Metro Mirror (PPRC) is used.
```

따라서 하나의 스트라이프가 있고 스트라이프가 없는 단일 로깅 모드의 큐 관리자는Metro Mirror 단일 쓰기 세션을 사용합니다. 두 개의 스트라이프와 PPRC가 있는 이중 로깅 모드의 큐 관리자는 8개의 쓰기 세션을 사용합니다.

참고: 하는 동안Metro Mirror 결과적으로 두 배의 쓰기 세션이 사용되며 해당 쓰기 세션은 두 개의 미러링된 DASD 간에 균등하게 분할됩니다.

z/OS *Planning your log archive storage*

Use this topic to understand the different ways of maintaining your archive log data sets.

You can place archive log data sets on standard-label tapes, or DASD, and you can manage them by data facility hierarchical storage manager (DFHSM). Each z/OS logical record in an archive log data set is a VSAM control interval from the active log data set. The block size is a multiple of 4 KB.

Archive log data sets are dynamically allocated, with names chosen by IBM MQ. The data set name prefix, block size, unit name, and DASD sizes needed for such allocations are specified in the system parameter module. You can also choose, at installation time, to have IBM MQ add a date and time to the archive log data set name.

It is not possible to specify with IBM MQ, specific volumes for new archive logs, but you can use Storage Management routines to manage this. If allocation errors occur, offloading is postponed until the next time offloading is triggered.

If you specify dual archive logs at installation time, each log control interval retrieved from the active log is written to two archive log data sets. The log records that are contained in the pair of archive log data sets are identical, but the end-of-volume points are not synchronized for multivolume data sets.

Should your archive logs reside on tape or DASD?

When deciding whether to use tape or DASD for your archive logs, there are a number of factors that you should consider:

- Review your operating procedures before deciding about tape or disk. For example, if you choose to archive to tape, there must be enough tape drive when they are required. After a disaster, all subsystems might want tape drives and you might not have as many free tape drives as you expect.
- During recovery, archive logs on tape are available as soon as the tape is mounted. If DASD archives have been used, and the data sets migrated to tape using hierarchical storage manager (HSM), there is a delay while HSM recalls each data set to disk. You can recall the data sets before the archive log is used. However, it is not always possible to predict the correct order in which they are required.
- When using archive logs on DASD, if many logs are required (which might be the case when recovering a page set after restoring from a backup) you might require a significant quantity of DASD to hold all the archive logs.
- In a low-usage system or test system, it might be more convenient to have archive logs on DASD to eliminate the need for tape mounts.
- Both issuing a RECOVER CFSTRUCT command, and backing out a persistent unit of work, result in the log being read backwards. Tape drives with hardware compression perform badly on operations that read backwards. Plan sufficient log data on DASD to avoid reading backwards from tape.

Archiving to DASD offers faster recoverability but is more expensive than archiving to tape. If you use dual logging, you can specify that the primary copy of the archive log go to DASD and the secondary copy go to tape. This increases recovery speed without using as much DASD, and you can use the tape as a backup.

See "Changing the storage medium for archive logs" on page 154 for details of how you archive your logs from tape to DASD, and how you carry out the reverse process.

Archiving to tape

If you choose to archive to a tape device, IBM MQ can extend to a maximum of 20 volumes.

If you are considering changing the size of the active log data set so that the set fits on one tape volume, note that a copy of the BSDS is placed on the same tape volume as the copy of the active log data set. Adjust the size of the active log data set downward to offset the space required for the BSDS on the tape volume.

If you use dual archive logs on tape, it is typical for one copy to be held locally, and the other copy to be held off-site for use in disaster recovery.

Archiving to DASD volumes

IBM MQ requires that you catalog all archive log data sets allocated on non-tape devices (DASD). If you choose to archive to DASD, the CATALOG parameter of the CSQ6ARVP macro must be YES. If this parameter is NO, and you decide to place archive log data sets on DASD, you receive message CSQJ072E each time an archive log data set is allocated, although IBM MQ still catalogs the data set.

If the archive log data set is held on DASD, the archive log data sets can extend to another volume; multivolume is supported.

If you choose to use DASD, make sure that the primary space allocation (both quantity and block size) is large enough to contain either the data coming from the active log data set, or that from the corresponding BSDS, whichever is the larger of the two.

This minimizes the possibility of unwanted z/OS X'B37' or X'E37' abend codes during the offload process. The primary space allocation is set with the PRIQTY (primary quantity) parameter of the CSQ6ARVP macro.

Archive log data sets can exist on large or extended-format sequential data sets. SMS ACS routines now use DSNTYPE(LARGE) or DSNTYPE(EXT).

IBM MQ supports allocation of archive logs as extended format data sets. When extended format is used, the maximum archive log size is increased from 65535 tracks to the maximum active log size of 4GB. Archive logs are eligible for allocation in the extended addressing space (EAS) of extended address volumes (EAV).

Where the required hardware and software levels are available, allocating archive logs to a data class defined with COMPACTION using zEDC might reduce the disk storage required to hold archive logs. For more information, see IBM MQ for z/OS: Reducing storage occupancy with IBM zEnterprise Data Compression (zEDC) and zEnterprise Data Compression (zEDC) for more information.

The z/OS data set encryption feature can be applied to archive logs for queue managers running on IBM MQ. These archive logs must be allocated through Automatic Class Selection (ACS) routines to a data class defined with EXTENDED attributes, and a data set key label that ensures the data is AES encrypted.

Using SMS with archive log data sets

If you have MVS/DFP storage management subsystem (DFSMS) installed, you can write an Automatic Class Selection (ACS) user-exit filter for your archive log data sets, which helps you convert them for the SMS environment.

Such a filter, for example, can route your output to a DASD data set, which DFSMS can manage. You must exercise caution if you use an ACS filter in this manner. Because SMS requires DASD data sets to

be cataloged, you must make sure the CATALOG DATA field of the CSQ6ARVP macro contains YES. If it does not, message CSQJ072E is returned; however, the data set is still cataloged by IBM MQ.

For more information about ACS filters, see Data sets that DFSMShsm dynamically allocates during aggregate backup processing.

z/OS *Changing the storage medium for archive logs*

The procedure for changing the storage medium used by archive logs.

About this task

This task describes how to change the storage medium used for archive logs, for example moving from archiving to tape to archiving to DASD.

You have a choice of how to make the changes:

1. Make the changes only using the CSQ6ARVP macro so that they are applied from the next time the queue manager restarts.
2. Make the changes using the CSQ6ARVP macro, and dynamically using the SET ARCHIVE command. This means that the changes apply from the next time the queue manager archives a log file, and persist after the queue manager restarts.

Procedure

1. Changing so archive logs are stored on DASD instead of tape:
   a) Read the section "Archiving to DASD volumes" on page 153 and review the CSQ6ARVP parameters.
   b) Make changes to the following parameters in CSQ6ARVP
      - Update the UNIT and, if necessary, the UNIT2 parameters.
      - Update the BLKSIZE parameter, as the optimal setting for DASD differs from tape.
      - Set the PRIQTY and SECQTY parameters to be large enough to hold the largest of the active log or BSDS.
      - Set the CATALOG parameter to be YES.
      - Confirm the ALCUNIT setting is what you want. You should use BLK, because it is independent of the device type.
      - Set the ARCWTOR parameter to NO if it is not already.
2. Changing so archive logs are stored on tape instead of DASD:
   a) Read the section "Archiving to tape" on page 153, and review the CSQ6ARVP parameters.
   b) Make changes to the following parameters in CSQ6ARVP:
      - Update the UNIT and, if necessary, the UNIT2 parameters.
      - Update the BLKSIZE parameter, as the optimal setting for tape differs from DASD.
      - Confirm the ALCUNIT setting is what you want. You should use BLK, because it is independent of the device type.
      - Review the setting of the ARCWTOR parameter.

z/OS ***How long do I need to keep archive logs***

Use the information in this section to help you plan your backup strategy.

You specify how long archive logs are kept in days , using the ARCRETN parameter in USING CSQ6ARVP or the SET SYSTEM command. After this period the data sets can be deleted by z/OS.

You can manually delete archive log data sets when they are no longer needed.

- The queue manager might need the archive logs for recovery.

The queue manager can only keep the most recent 1000 archives in the BSDS, When the archive logs are not in the BSDS they cannot be used for recovery, and are only of use for audit, analysis, or replay type purposes.

- You might want to keep the archive logs so that you can extract information from the logs. For example, extracting messages from the log, and reviewing which user ID put or got the message.

The BSDS contains information on logs and other recovery information. This data set is a fixed size. When the number of archive logs reaches the value of MAXARCH in CSQ6LOGP, or when the BSDS fills up, the oldest archive log information is overwritten.

There are utilities to remove archive log entries from the BSDS, but in general, the BSDS wraps and overlays the oldest archive log record.

When is an archive log needed

You need to back up your page sets regularly. The frequency of backups determines which archive logs are needed in the event of losing a page set.

You need to back up your CF structures regularly. The frequency of backups determines which archive logs are needed in the event of losing data in the CF structure.

The archive log might be needed for recovery. The following information explains when the archive log might be needed, where there are problems with different IBM MQ resources.

Loss of a page set
You must recover your system from your backup and restart the queue manager.

You need the logs from when the backup was taken, as well as up to three log data sets prior to the backup being taken.

All LPARs lose connectivity to a CF structure, or the structure is unavailable
Use the RECOVER CFSTRUCT command to recover the structure.

Structure recovery requires the logs from all queue managers that have accessed the structure since the last backup (back to the time when the backup was taken) plus the structure backup itself in the log of the queue manager that took the backup.

If you have been doing frequent backups of the CF structures, the data should be in active logs, and you should not need archive logs.

If there is no recent backup of the CF structure, you might need archive logs.

Note: All non persistent messages will be lost; all persistent messages will be re-created by performing the following tasks:

1. Reading the last CF structure backup from the log
2. Reading the logs from all queue managers that have used the structure
3. Merging updates since the backup

Administration structure rebuild
If you need to rebuild the administration structure, the information is read from the last checkpoint of the log for each queue manager in the QSG.

If a queue manager is not active, another queue manager in the QSG reads the log.

You should not need archive logs.

Loss of an SMDS data set
If you lose an SMDS data set, or the data set gets corrupted, the data set becomes unusable and the status for it is set to FAILED. The CF structure is unchanged.

In order to restore the SMDS data set, you need to:

1. Redefine the SMDS data set, and
2. Recover the CF structure by issuing the RECOVER CFSTRUCT command.

Note: All non persistent messages on the CF structure will be lost; all persistent messages will be restored.

The requirement for queue manager logs is the same as for recovering from a structure that is unavailable.

z/OS *Planning to increase the maximum addressable log range*

You can increase the maximum addressable log range by configuring your queue manager to use a larger log relative byte address (RBA).

The log RBA size was increased from IBM MQ for z/OS 8.0. For an overview of this change, see Larger log Relative Byte Address.

Queue managers created at IBM MQ 9.3.0 or later, have 8 byte log RBA enabled by default and, therefore, do not require conversion.

You can convert your queue managers to use 8 byte log RBA values at any time. A queue sharing group can contain some queue managers with 8 byte log RBA enabled, and some queue managers with 6 byte log RBA enabled.

Undoing the change

The change cannot be backed out.

How long does it take?

The change requires a queue manager restart. Stop the queue manager, run the CSQJUCNV utility against the bootstrap data set (BSDS), or data sets, to create new data sets, rename these bootstrap data sets, and restart the queue manager. The CSQJUCNV utility usually takes a few seconds to run.

What impact does this have?

- With 8 byte log RBA in use, every write of data to the log data sets has additional bytes. Therefore, for a workload consisting of persistent messages there is a small increase in the amount of data written to the logs.
- Data written to a page set, or coupling facility (CF) structure, is not affected.

Related tasks

Implementing the larger log Relative Byte Address

z/OS Planning your channel initiator

The channel initiator provides communications between queue managers, and runs in its own address space.

There are two types of connections:

1. Application connections to a queue manager over a network. These are known as client channels.
2. Queue manager to queue manager connections. These are known as MCA channels.

Listeners

A channel listener program listens for incoming network requests and starts the appropriate channel when that channel is needed. To process inbound connections the channel initiator needs at least one IBM MQ listener task configured. A listener can either be a TCP listener, or a LU 6.2 listener.

Each listener requires a TCP port or LU name.

Note that you can have more than one listener for each channel initiator.

TCP/IP

A channel initiator can operate with more than one TCP stack on the same z/OS image. For example, one TCP stack could be for internal connections, and another TCP stack for external connections.

When you define an output channel:

1. You set the destination host and port of the connection. This can be either:
   - an IP address, for example `10.20.4.6`
   - a host name, for example `mvs-prod.myorg.com`

   If you use a host name to specify the destination, IBM MQ uses the Domain Name System (DNS) to resolve the IP address of the destination.
2. If you are using multiple TCP stacks you can specify the **LOCLADDR** parameter on the channel definition, which specifies the IP Stack address to be used.

You should plan to have a highly available DNS server, or DNS servers. If the DNS is not available, outbound channels might not be able to start, and channel authentication rules that map an incoming connection using a host name cannot be processed.

APPC and LU 6.2

If you are using APPC, the channel initiator needs an LU name, and configuration in APPC.

Queue sharing groups

To provide a single system image, and allow an incoming IBM MQ connection request to go to any queue manager in the queue sharing group, you need to do some configuration. For example:

1. A hardware network router. This router has one IP address seen by the enterprise, and can route the initial request to any queue manager connected to this hardware.
2. A Virtual IP address (VIPA). An enterprise wide IP address is specified, and that address can be routed to any one of the TCP stacks in a sysplex. The TCP stack can then route it to any listening queue manager in the sysplex.

Protecting IBM MQ traffic

You can configure IBM MQ to use TLS connections to protect data on the wire. To use TLS you need to use digital certificates and key rings.

You also need to work with the personnel at the remote end of the channel, to ensure that you have compatible IBM MQ definitions and compatible certificates.

You can control which connections can connect to IBM MQ and the user ID, based on

- IP address
- Client user ID
- Remote queue manager, or
- Digital certificate (see Channel Authentication Records)

It is also possible to restrict client applications by ensuring that they supply a valid user ID and password (see Connection Authentication).

You can get the channel initiator working, and then configure each channel to use TLS, one at a time.

Monitoring the channel initiator

There are MQSC commands that give information about the channel initiator and channels:

- The DISPLAY CHINIT command gives information about the channel initiator, and active listeners.

- The DISPLAY CHSTATUS command displays the activity and status of a channel.

The channel initiator can also produce SMF records with information about the channel initiator tasks and channel activity. See "Planning for channel initiator SMF data" on page 159 for more information.

The channel initiator emits messages to the job log when channels start and stop. Automation in your enterprise can use these messages to capture status. As some channels are active for only a few seconds, many messages can be produced. You can suppress these messages either by using the z/OS message processing facility, or by setting **EXCLMSG** with the SET SYSTEM command.

Configuring your IBM MQ channel definitions

When you have many queue managers connected together it can be hard to manage all the object definitions. Using IBM MQ clustering can simplify this.

You specify two queue managers as full repositories. Other queue managers need one connection to, and one connection from, one of the repositories. When connections to other queue managers are needed, the queue manager creates and starts channels automatically.

If you are planning to have a large number of queue managers in a cluster, you should plan to have queue managers that act as dedicated repositories and have no application traffic.

See "분산 큐와 클러스터 계획" on page 18 for more information.

Actions before you configure the channel initiator

1. Decide if you are using TCP/IP or APPC.
2. If you are using TCP, allocate at least one port for IBM MQ.
3. If you need a a DNS server, configure the server to be highly available if required.
4. If you are using APPC, allocate an LU name, and configure APPC.

Actions after you have configured the channel initiator, before you go into production

1. Plan what connections you will have:
   a. Client connections from remote applications.
   b. MCA channels to and from other queue managers. Typically you have a channel to and from each remote queue manager.
2. Set up clustering, or join an existing clustering environment.
3. Consider whether you need to use multiple TCP stacks, VIPA, or an external router for availability in front of the channel initiator.
4. If you are planning on using TLS:
   a. Set up the key ring
   b. Set up certificates
5. If you are planning on using channel authentication:
   a. Decide the criteria for mapping inbound sessions to MCA user IDs
   b. Enable reverse DNS lookup by setting the queue manager parameter **REVDNS**
   c. Review security. For example, delete the default channels, and specify user IDs with only the necessary authority in the **MCAUSER** attribute for a channel.
6. Capture the accounting and statistics SMF records produced by the channel initiator and post process them.
7. Automate the monitoring of job log messages.
8. If necessary, tune your network environment to improve throughput. With TCP, large send and receive buffers improve throughput. You can force MQ to use specific TCP buffer sizes using the commands:

```
RECOVER QMGR(TUNE CHINTCPRBDYNSZ nnnnn)
RECOVER QMGR(TUNE CHINTCPSBDYNSZ nnnnn)
```

which sets the SO_RCVBUF, and SO_SNDBUF, for the channels to the size in bytes specified in nnnnn.

Related concepts

"Planning for your queue manager" on page 128

When you are setting up a queue manager, your planning should allow for the queue manager to grow, so that the queue manager meets the needs of your enterprise.

z/OS Planning for channel initiator SMF data

You need to plan the implementation of collecting SMF data for the channel initiator.

The channel initiator produces two types of record:

- Statistics data with information about the channel initiator and the tasks within it.
- Channel accounting data with information similar to the DISPLAY CHSTATUS command.

You start collecting statistics data using the command:

```
START TRACE(STAT) CLASS(4)
```

and stop it using the command:

```
STOP TRACE(STAT) CLASS(4)
```

You start collecting accounting data using the command:

```
START TRACE(ACCTG) CLASS(4)
```

and stop it using the command:

```
STOP TRACE(ACCTG) CLASS(4)
```

You can control which channels have accounting data collected for using the **STATCHL** attribute on the channel definition or the queue manager.

- For client channels, you must set **STATCHL** at the queue manager level.
- For automatically defined cluster sender channels, you can control the collection of accounting data with the **STATACLS** queue manager attribute.

The default value of **STATCHL** for the queue manager is OFF. In order to collect channel accounting data you must change the value of **STATCHL** from the default on either the queue manager or channel definition, in addition to starting class 4 accounting trace.

The SMF records are produced when:

- From IBM MQ for z/OS 9.3.0 onwards, the time interval indicated by the CSQ6SYSP **STATIME** or **ACCTIME** parameters has elapsed; or, if **STATIME** or **ACCTIME** is zero on the SMF data collection broadcast. The requests to collect SMF data for the channel initiator and the queue manager are synchronized.
- A `STOP TRACE(ACCTG) CLASS(4)` or `STOP TRACE(STAT) CLASS(4)` command is issued, or
- The channel initiator is shut down. At this point any SMF data is written out.

If a channel stops during the SMF interval, accounting data is written to SMF the next time the SMF processing runs. If a client connects, does some work and disconnects, then reconnects and disconnects, there are two sets of channel accounting data produced.

The statistics data normally fits into one SMF record, however, multiple SMF records might be created if a large number of tasks are in use.

Accounting data is gathered for each channel for which it is enabled, and normally fits into one SMF record. However, multiple SMF records might be created if a large number of channels are active.

The cost of collecting the channel initiator SMF data is small. Typically the increase in CPU usage is under a few percent, and often within measurement error.

Before you use this function you need to work with your z/OS systems programmer to ensure that SMF has the capacity for the additional records, and that they change their processes for extracting SMF records to include the new SMF data.

For channel initiator statistics data, the SMF record type is 115 and sub-type 231.

For channel initiator accounting data, the SMF record type is 116 and sub-type 10.

You can write your own programs to process this data, or use the SupportPac MP1B that contains a program, MQSMF, for printing the data, and creating data in Comma Separated Values (CSV) format suitable for importing into a spread sheet.

If you are experiencing issues with capturing channel initiator SMF data, see Dealing with issues when capturing SMF data for the channel initiator (CHINIT) for further information.

Related tasks

Interpreting IBM MQ performance statistics

Troubleshooting channel accounting data

z/OS Planning your z/OS TCP/IP environment

To get the best throughput through your network, you must use TCP/IP send and receive buffers with a size of 64 KB, or greater. With this size, the system optimizes its buffer sizes.

See What is Dynamic Right Sizing for High Latency Networks? for more information.

You can check your system buffer size by using the following Netstat command, for example:

```
TSO NETSTAT ALL (CLIENT csq1CHIN
```

The results display much information, including the following two values:

```
ReceiveBufferSize: 0000065536
SendBufferSize: 0000065536
```

65536 is 64 KB. If your buffer sizes are less than 65536, you must work with your network team to increase the **TCPSENDBFRSIZE** and **TCPRCVBUFRSIZE** values in the `PROFILE DDName` in the TCPIP procedure. For example, you might use the following command:

```
TCPCONFIG TCPSENDBFRSZE 65536 TCPRCVBUFRSIZE 65536
```

If you are unable to change your system-wide **TCPSENDBFRSIZE** or **TCPRCVBUFRSIZE** settings, contact your IBM Software Support center.

z/OS Planning your queue sharing group (QSG)

The easiest way to implement a shared queuing environment, is to configure a queue manager, add that queue manager to a QSG, then add other queue managers to the QSG.

A queue sharing group uses Db2 tables to store configuration information. There is one set of tables used by all QSGs that share the same Db2 data sharing group.

Shared queue messages are stored in a structure in a coupling facility (CF). Each QSG has its own set of CF structures. You need to configure the structures to meet your needs.

Messages over 63KB in size cannot be stored in the CF. You need to use either Shared Message Data Sets (SMDS) or Db2 for these messages.

Message profiles and capacity planning

You should understand the message profile of your shared queue messages. The following are examples of factors that you need to consider:

- Average, and maximum message size
- The typical queue depth, and exception queue depth. For example, you might need to have enough capacity to hold messages for a whole day, and the typical queue depth is under 100 messages.

If the message profile changes, you can increase the size of the structures, or implement SMDS, at a later date.

If you want to be able to handle a large peak volume of messages, you can configure IBM MQ to offload messages to SMDS when the usage of the structure reaches user specified thresholds.

You need to decide if you want to duplex the CF structures. This is controlled by the CF structure definition in the CFRM policy:

1. A duplexed structure uses two coupling facilities. If there is a problem with one CF, there is no interruption to the service, and the structure can be rebuilt on a third CF, if one is available. Duplexed structures can significantly impact the performance of operations on shared queues.
2. If the structure is not duplexed, then a problem with the CF means that shared queues on structures in that CF will become unavailable until the structure can be rebuilt in another CF.

   IBM MQ can be configured to automatically rebuild structures in another CF in this case. Persistent messages will be recovered from the logs of the queue managers.

Note that it is easy to change the CF definitions.

You can define a structure so that it can hold nonpersistent messages only, or so that it can hold persistent and nonpersistent messages.

Structures that can hold persistent messages need to be backed up periodically. Back up your CF structures at least every hour to minimize the time needed to recover the structure in the event of a failure. The backup is stored in the log data set of the queue manager performing the backup.

If you are expecting to have a high throughput of messages on your shared queues, it is best practice to have a dedicated queue manager for backing up the CF structures. This reduces the time needed to recover the structures, as a less data needs to be read from queue manager logs.

Channels

To provide a single system image for applications connecting into an IBM MQ QSG, you can define shared input channels. If these are set up, then a connection coming into the queue sharing group environment, can go to any queue manager in the QSG.

You might need to set up a network router, or Virtual IP address (VIPA) for these channels.

You can define shared output channels. A shared output channel instance can be started from any queue manager in the QSG.

See Shared channels for more information.

Security

You protect IBM MQ resources using an external security manager. If you are using RACF®, the RACF profiles are prefixed with the queue manager name. For example, a queue named APPLICATION.INPUT would be protected using a profile in the MQQUEUE class named `qmgrName.APPLICATION.INPUT`.

When using a queue sharing group you can continue to protect resources with profiles prefixed with the queue manager name, or you can prefix profiles with the queue sharing group name. For example `qsgName.APPLICATION.INPUT`.

You should aim to use profiles prefix with the queue sharing group name because this means there is a single definition for all queue managers, saving you work, and preventing a mismatch in definitions between queue managers.

Related concepts

"Planning for your queue manager" on page 128
When you are setting up a queue manager, your planning should allow for the queue manager to grow, so that the queue manager meets the needs of your enterprise.

z/OS Planning your coupling facility and offload storage environment

Use this topic when planning the initial sizes, and formats of your coupling facility (CF) structures, and shared message data set (SMDS) environment or Db2 environment.

This section contains information about the following topics:

Defining coupling facility resources

If you intend to use shared queues, you must define the coupling facility structures that IBM MQ will use in your CFRM policy. To do this you must first update your CFRM policy with information about the structures, and then activate the policy.

Your installation probably has an existing CFRM policy that describes the coupling facilities available. The Administrative data utility is used to modify the contents of the policy based on textual statements you provide. You must add statements to the policy that defines the names of the new structures, the coupling facilities that they are defined in, and what size the structures are.

The CFRM policy also determines whether IBM MQ structures are duplexed and how they are reallocated in failure scenarios. Shared queue recovery contains recommendations for configuring CFRM for resilience to failures that affect the coupling facility.

Deciding your offload storage environment

The message data for shared queues can be offloaded from the coupling facility and stored in either a Db2 table or in an IBM MQ managed data set called a *shared message data set* (SMDS). Messages which are too large to store in the coupling facility (that is, larger than 63 KB) must always be offloaded, and smaller messages can optionally be offloaded to reduce coupling facility space usage.

For more information, see Specifying offload options for shared messages.

Planning your structures

A queue sharing group (QSG) requires a minimum of two structures to be defined. The first structure, known as the administrative structure, is used to coordinate IBM MQ internal activity across the queue sharing group. No user data is held in this structure. It has a fixed name of *qsg-name*CSQ_ADMIN (where *qsg-name* is the name of your queue sharing group). Subsequent structures are known as application structures, and are used to hold the messages on IBM MQ shared queues. Each structure can hold up to 512 shared queues.

An application structure named *qsg-name*CSQSYSAPPL is used for system queues. Defining this structure is optional, but it is required in order to use certain features. By default, the SYSTEM.QSG.CHANNEL.SYNCQ and SYSTEM.QSG.UR.RESOLUTION.QUEUE queues are defined on the *qsg-name*CSQSYAPPL structure.

Using multiple structures

A queue sharing group can connect to up to 64 coupling facility structures. One of these structures must be the administration structure. If it is defined, another of these structures might be the *qsg-name*CSQSYSAPPL structure. You can use up to 63 (62 if *qsg-name*CSQSYSAPPL is defined) structures for message data. You might choose to use multiple application structures for any of the following reasons:

- You have some queues that are likely to hold a large number of messages and so require all the resources of an entire coupling facility.
- You have a requirement for a large number of shared queues, so they must be split across multiple structures because each structure can contain only 512 queues.
- RMF reports on the usage characteristic of a structure suggest that you should distribute the queues it contains across a number of coupling facilities.
- You want some queue data to held in a physically different coupling facility from other queue data for data isolation reasons.
- Recovery of persistent shared messages is performed using structure level attributes and commands, for example BACKUP CFSTRUCT. To simplify backup and recovery, you could assign queues that hold nonpersistent messages to different structures from those structures that hold persistent messages.

When choosing which coupling facilities to allocate the structures in, consider the following points:

- Your data isolation requirements.
- The volatility of the coupling facility (that is, its ability to preserve data through a power outage).
- Failure independence between the accessing systems and the coupling facility, or between coupling facilities.
- The level of coupling facility control code (CFCC) installed on the coupling facility (IBM MQ requires Level 9 or higher).

Planning the size of your structures

The administrative structure

The administrative structure (*qsg-name*CSQ_ADMIN) must be large enough to contain 1000 list entries for each queue manager in the queue sharing group. When a queue manager starts, the structure is checked to see if it is large enough for the number of queue managers currently *defined* to the queue sharing group. Queue managers are considered as being defined to the queue sharing group if they have been added by the CSQ5PQSG utility. You can check which queue managers are defined to the group with the MQSC DISPLAY GROUP command.

Note: When calculating the size of the structure, you should allow for the size of large units of work, in addition to the number of queue managers in the queue sharing group.

Table 22 on page 164 shows the minimum required size for the administrative structure for various numbers of queue managers defined in the queue sharing group. These sizes were established for a CFCC level 14 coupling facility structure; for higher levels of CFCC, they probably need to be larger.

Table 22. Minimum administrative structure sizes

Number of queue managers defined in queue sharing group	Required storage
1	6144 KB
2	6912 KB
3	7976 KB
4	8704 KB
5	9728 KB
6	10496 KB
7	11520 KB
8	12288 KB
9	13056 KB
10	14080 KB
11	14848 KB
12	15616 KB
13	16640 KB
14	17408 KB
15	18176 KB
16	19200 KB
17	19968 KB
18	20736 KB
19	21760 KB
20	22528 KB
21	23296 KB
22	24320 KB
23	25088 KB
24	25856 KB
25	27136 KB
26	27904 KB
27	28672 KB
28	29696 KB
29	30464 KB
30	31232 KB
31	32256 KB

When you add a queue manager to an existing queue sharing group, the storage requirement might have increased beyond the size recommended in Table 22 on page 164. If so, use the following procedure to estimate the required storage for the *qsg-name*CSQ_ADMIN structure:

1. Issue MQSC command **DISPLAY CFSTATUS(CSQ_ADMIN)** on an existing member of the queue sharing group.
2. Extract the ENTSMAX information for the CSQ_ADMIN structure.
3. If this number is less than 1000 times the total number of queue managers you want to define in the queue sharing group, increase the structure size.

Application structures

The size of the application structures required to hold IBM MQ messages depends on the likely number and size of the messages to be held on a structure concurrently.

The graph in Figure 42 on page 165 shows how large you should make your CF structures to hold the messages on your shared queues. To calculate the allocation size you need the following information:

- The average size of messages on your queues.
- The total number of messages likely to be stored in the structure.

Find the number of messages along the horizontal axis. Select the curve that corresponds to your message size and determine the required value from the vertical axis. For example, for 200 000 messages of length 1 KB gives a value in the range 256 through 512 MB.

Table 23 on page 166 provides the same information in tabular form.

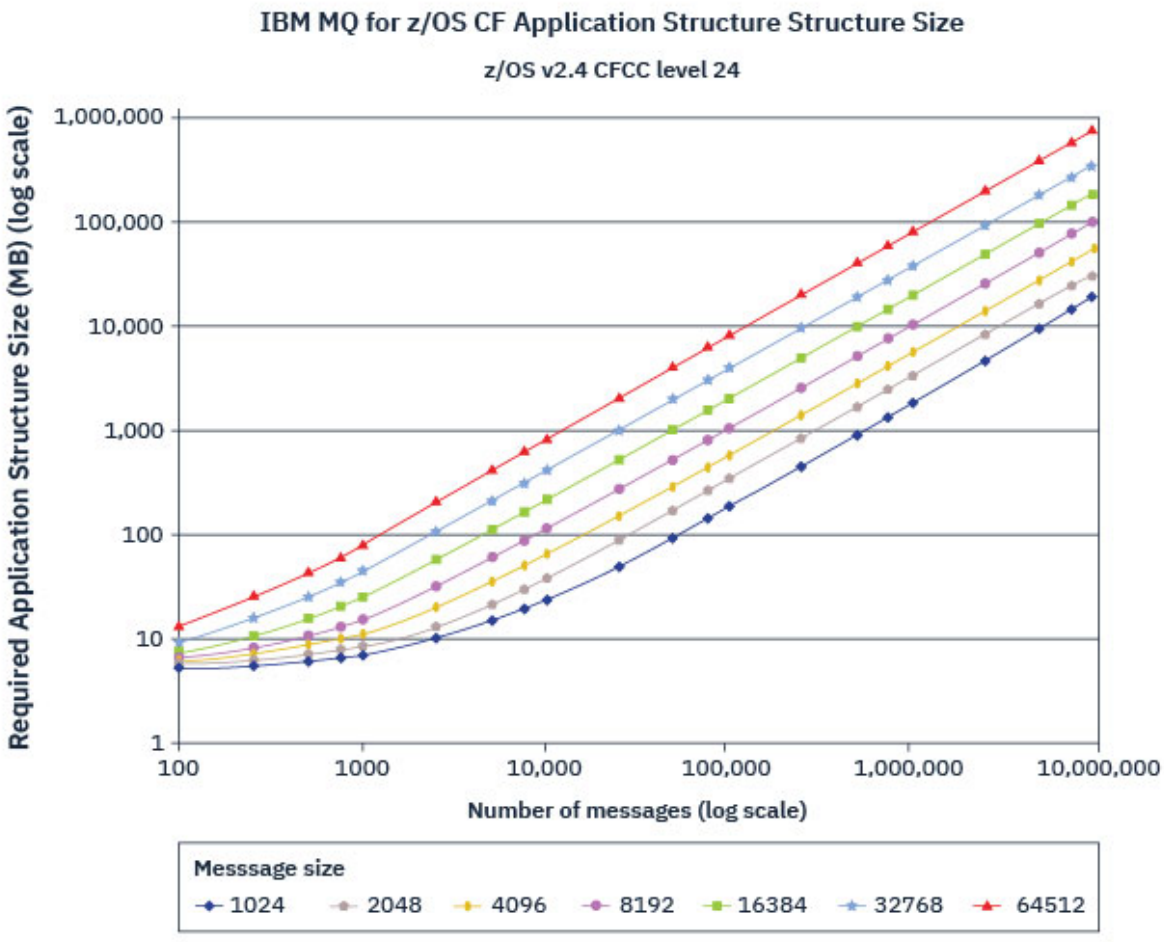


Figure 42. Calculating the size of a coupling facility structure

Use this table to help calculate how large to make your coupling facility structures:

Table 23. Calculating the size of a coupling facility structure

Number of messages	1 KB	2 KB	4 KB	8 KB	16 KB	32 KB	63 KB
100	6 MB	6 MB	7 MB	7 MB	8 MB	10 MB	14 MB
1000	8 MB	9 MB	12 MB	17 MB	27 MB	48 MB	88 MB
10000	25 MB	38 MB	64 MB	115 MB	218 MB	423 MB	821 MB
100000	199 MB	327 MB	584 MB	1097 MB	2124 MB	4177 MB	8156 MB

Your CFRM policy should include the following statements:

- INITSIZE is the size in KB that the structure is allocated with when the first queue manager connects to it.
- SIZE is the maximum size that the structure can attain.
- FULLTHRESHOLD sets the percentage value of the threshold at which z/OS issues message IXC585E to indicate that the structure is getting full.

A best practice is to ensure that INITSIZE and SIZE are within a factor of 2. For example, with the figures determined previously, you might include the following statements:

```
STRUCTURE NAME(structure-name)
INITSIZE(value from graph in KB, that is, multiplied by 1024)
SIZE(something larger)
FULLTHRESHOLD(85)
```

```
STRUCTURE NAME(QSG1APPLICATION1)
INITSIZE(262144) /* 256 MB */
SIZE(524288)     /* 512 MB */
FULLTHRESHOLD(85)
```

If the structure use reaches the threshold where warning messages are issued, intervention is required. You might use IBM MQ to inhibit MQPUT operations to some of the queues in the structure to prevent applications from writing more messages, start more applications to get messages from the queues, or quiesce some of the applications that are putting messages to the queue.

Alternatively, you can use z/OS facilities to alter the structure size in place. The following z/OS command:

```
SETXCF START,ALTER,STRNAME=structure-name,SIZE=newsize
```

alters the size of the structure to *newsize*, where *newsize* is a value that is less than the value of SIZE specified on the CFRM policy for the structure, but greater than the current coupling facility size.

You can monitor the use of a coupling facility structure with the MQSC DISPLAY CFSTATUS command.

If no action is taken and a queue structure fills up, an MQRC_STORAGE_MEDIUM_FULL return code is returned to the application. If the administration structure becomes full, the exact symptoms depend on which processes experience the error, but they might include the following problems:

- No responses to commands.
- Queue manager failure as a result of problems during commit processing.

The CSQSYSAPPL structure

The *qsg-name*CSQSYSAPPL structure is an application structure for system queues. Table 3 demonstrates an example of how to estimate the message data sizes for the default queues defined on the *qsg-name*CSQSYSAPPL structure.

Table 24. Table showing CSQSYSAPPL usage against sizing.	
***qsg-name*CSQSYSAPPL usage**	**Sizing**
SYSTEM.QSG.CHANNEL.SYNCQ	2 messages of 500 bytes per active instance of a shared channel
SYSTEM.QSG.UR.RESOLUTION.QUEUE	1000 messages of 2 KB

The suggested initial structure definition values are as follows:

```
STRUCTURE NAME(qsg-nameCSQSYSAPPL)
INITSIZE(20480)        /* 20 MB */
SIZE(30720)            /* 30 MB */
FULLTHRESHOLD(85)
```

These values can be adjusted depending on your use of shared channels and group units of recovery.

Mapping shared queues to structures

To define an application structure to IBM MQ, use the DEFINE CFSTRUCT command. When you define a structure to IBM MQ, do not include the QSG name prefix in the structure name. For example, to define an application structure to IBM MQ that has the name *qsg-name*APPLICATION1 in the CFRM policy, issue the following command:

```
DEFINE CFSTRUCT(APPLICATION1)
```

The CFSTRUCT attribute of the queue definition is used to map the queue to a structure. Specify the name of the CF structure without the QSG name prefix in this attribute. For example, the following command defines a shared queue on the APPLICATION1 structure:

```
DEFINE QLOCAL(myqueue) QSGDISP(SHARED) CFSTRUCT(APPLICATION1)
```

z/OS *Planning your shared message data set (SMDS) environment*

If you are using queue sharing groups with SMDS offloading, IBM MQ needs to connect to a group of shared message data sets. Use this topic to help understand the data set requirements, and configuration required to store IBM MQ message data.

A *shared message data set* (described by the keyword SMDS) is a data set used by a queue manager to store offloaded message data for shared messages stored in a coupling facility structure.

Note: When defining SMDS data sets for a structure, you must have one for each queue manager.

When this form of data offloading is enabled, the **CFSTRUCT** requires an associated group of shared message data sets, one data set for each queue manager in the queue sharing group. The group of shared message data sets is defined to IBM MQ using the **DSGROUP** parameter on the **CFSTRUCT** definition. Additional parameters can be used to supply further optional information, such as the number of buffers to use and expansion attributes for the data sets.

Each queue manager can write to the data set which it owns, to store shared message data for messages written through that queue manager, and can read all of the data sets in the group.

A list describing the status and attributes for each data set associated with the structure is maintained internally as part of the **CFSTRUCT** definition, so each queue manager can check the definition to find out which data sets are currently available.

This data set information can be displayed using the **DISPLAY CFSTATUS TYPE(SMDS)** command to display current status and availability, and the **DISPLAY SMDS** command to display the parameter settings for the data sets associated with a specified **CFSTRUCT**.

Individual shared message data sets are effectively identified by the combination of the owning queue manager name (usually specified using the **SMDS** keyword) and the **CFSTRUCT** structure name.

This section describes the following topics:

- The DSGROUP parameter
- The DSBLOCK parameter
- Shared message data set characteristics
- Shared message data set space management
- Access to shared message data sets
- Creating a shared message data set
- Shared message data set performance and capacity considerations
- Activating a shared message data set

See DEFINE CFSTRUCT for details of these parameters.

For information on managing your shared message data sets, see Managing shared message data sets for further details.

The DSGROUP parameter

The **DSGROUP** parameter on the **CFSTRUCT** definition identifies the group of data sets in which large messages for that structure are to be stored. Additional parameters may be used to specify the logical block size to be used for space allocation purposes and values for the buffer pool size and automatic data set expansion options.

The **DSGROUP** parameter must be set up before offloading to data sets can be enabled.

- If a new **CFSTRUCT** is being defined at **CFLEVEL(5)** and the option **OFFLOAD(SMDS)** is specified or assumed, then the **DSGROUP** parameter must be specified on the same command.
- If an existing **CFSTRUCT** is being altered to increase the **CFLEVEL** to **CFLEVEL(5)** and the option **OFFLOAD(SMDS)** is specified or assumed, then the **DSGROUP** parameter must be specified on the same command if it is not already set.

The DSBLOCK parameter

Space within each data set is allocated to queues as logical blocks of a fixed size (usually 256 KB) specified using the **DSBLOCK** parameter on the **CFSTRUCT** definition, then allocated to individual messages as ranges of pages of 4 KB (corresponding to the physical block size and control interval size) within each logical block. The logical block size also determines the maximum amount of message data that can be read or written in a single I/O operation, which is the same as the buffer size for the SMDS buffer pool.

A larger value of the **DSBLOCK** parameter can improve performance for very large messages by reducing the number of separate I/O operations. However, a smaller value decreases the amount of buffer storage required for each active request. The default value for the **DSBLOCK** parameter is 256 KB, which provides a reasonable balance between these requirements, so specifying this parameter might not normally be necessary.

Shared message data set characteristics

A shared message data set is defined as a VSAM linear data set (LDS). Each offloaded message is stored in one or more blocks in the data set. The stored data is addressed directly by information in the coupling facility entries, like an extended form of virtual storage. There is no separate index or similar control information stored in the data set itself.

The direct addressing scheme means that for messages which fit into one block, only a single I/O operation is needed to read or write the block. When a message spans more than one block, the I/O

operations for each block can be fully overlapped to minimize elapsed time, provided that sufficient buffers are available.

The shared message data set also contains a small amount of general control information, consisting of a header in the first page, which includes recovery and restart status information, and a space map checkpoint area which is used to save the free block space map at queue manager normal termination.

Shared message data set space management

As background information for capacity, performance and operational considerations, it might be useful to understand the concepts of how space in shared message data sets is managed by the queue managers.

Free space in each shared message data set is tracked by its owning queue manager using a space map which indicates the number of pages in use within each logical block. The space map is maintained in main storage while the data set is open and saved in the data set when it is closed normally. (In recovery situations the space map is automatically rebuilt by scanning the messages in the coupling facility structure to find out which data set pages are currently in use).

When a shared message with offloaded message data is being written, the queue manager allocates a range of pages for each message block. If there is a partly used current logical block for the specified queue, the queue manager allocates space starting at the next free page in that block, otherwise it allocates a new logical block. If the whole message does not fit within the current logical block, the queue manager splits the message data at the end of the logical block and allocates a new logical block for the next message block. This is repeated until space has been allocated for the whole message. Any unused space in the last logical block is saved as the new current logical block for the queue. When the data set is closed normally, any unused pages in current logical blocks are returned to the space map before it is saved.

When a shared message with offloaded message data has been read and is ready to be deleted, the queue manager processes the delete request by transferring the coupling facility entry for the message to a clean-up list monitored by the owning queue manager (which may be the same queue manager). When entries arrive on this list, the owning queue manager reads and deletes the entries and returns the freed ranges of pages to the space map. When all used pages in a logical block have been freed the block becomes available for reuse.

Access to shared message data sets

Each shared message data set must be on shared direct access storage which is accessible to all queue managers in the queue sharing group.

During normal running, each queue manager opens its own shared message data set for read/write access, and opens any active shared message data sets for other queue managers for read-only access, so it can read messages stored by those queue managers. This means that each queue manager userid requires at least UPDATE access to its own shared message data set and READ access to all other shared message data sets for the structure.

If it is necessary to recover shared message data sets using **RECOVER CFSTRUCT**, the recovery process can be executed from any queue manager in the queue sharing group. A queue manager which may be used to perform recovery processing requires UPDATE access to all data sets that it may need to recover

Creating a shared message data set

Each shared message data set should normally be created before the corresponding **CFSTRUCT** definition is created or altered to enable the use of this form of message offloading, as the **CFSTRUCT** definition changes will normally take effect immediately, and the data set will be required as soon as a queue manager attempts to access a shared queue which has been assigned to that structure. A sample job to allocate and pre-format a shared message data set is provided in SCSQPROC(CSQ4SMDS). The job must be customized and run to allocate a shared message data set for each queue manager which uses a CFSTRUCT with OFFLOAD(SMDS).

If the queue manager finds that offload support has been enabled and tries to open its shared message data set but it has not yet been created, the shared message data set will be flagged as unavailable. The queue manager will then be unable to store any large messages until the data set has been created and the queue manager has been notified to try again, for example using the **START SMDSCONN** command.

A shared message data set is created as a VSAM linear data set using an Access Method Services **DEFINE CLUSTER** command. The definition must specify **SHAREOPTIONS(2 3)** to allow one queue manager to open it for write access and any number of queue managers to read it at the same time. The default control interval size of 4 KB must be used. If the data set may need to expand beyond 4 GB, it must be defined using an SMS data class which has the VSAM extended addressability attribute. A shared message data set is eligible to reside in the extended addressing space (EAS) part of an extended address volumes (EAV).

Each shared message data set can either be empty or pre-formatted to binary zeros (using **CSQJUFMT** or a similar utility such as the sample job SCSQPROC(CSQ4SMDS)), before its initial use. If it is empty or only partly formatted when it is opened, the queue manager automatically formats the remaining space to binary zeros.

Shared message data set performance and capacity considerations

Each shared message data set is used to store offloaded data for shared messages written to the associated **CFSTRUCT** by the owning queue manager, from regions within the same system. Each message that is offloaded takes up to 768 bytes of CF storage, made up of 256 bytes for the entry and 512 bytes for the two elements of header and descriptor. Each offloaded message is stored in one or more pages (physical blocks of size 4 KB) in the data set.

The data set space required for a given number of offloaded messages can therefore be estimated by rounding up the overall message size (including the descriptor) to the next multiple of 4 KB and then multiplying by the number of messages.

As for a page set, when a shared message data set is almost full, it can optionally be expanded automatically. The default behavior for this automatic expansion can be set using the **DSEXPAND** parameter on the **CFSTRUCT** definition. This setting can be overridden for each queue manager using the **DSEXPAND** parameter on the **ALTER SMDS** command. Automatic expansion is triggered when the data set reaches 90% full and more space is required. If expansion is allowed but an expansion attempt is rejected by VSAM because no secondary space allocation was specified when the data set was defined, expansion is retried using a secondary allocation of 20% of the current size of the data set.

Provided that the shared message data set is defined with the extended addressability attribute, the maximum size is only limited by VSAM considerations to a maximum of 16 TB or 59 volumes. This is significantly larger than the 64 GB maximum size of a local page set.

Activating a shared message data set

When a queue manager has successfully connected to an application coupling facility structure, it checks whether that structure definition specifies offloading using an associated **DSGROUP** parameter. If so, the queue manager allocates and opens its own shared message data set for write access, then it opens for read access any existing shared message data sets owned by other queue managers.

When a shared message data set is opened for the first time (before it has been recorded as active within the queue sharing group), the first page will not yet contain a valid header. The queue manager fills in header information to identify the queue sharing group, the structure name and the owning queue manager.

After the header has been completed, the queue manager registers the new shared message data set as active and broadcasts an event to notify any other active queue managers about the new data set.

Every time a queue manager opens a shared message data set it validates the header information to ensure that the correct data set is still being used and that it has not been damaged.

z/OS *Planning your Db2 environment*

If you are using queue sharing groups, IBM MQ needs to attach to a Db2 subsystem that is a member of a data sharing group. Use this topic to help understand the Db2 requirements used to hold IBM MQ data.

IBM MQ needs to know the name of the data sharing group that it is to connect to, and the name of a Db2 subsystem (or Db2 group) to connect to, to reach this data sharing group. These names are specified in the QSGDATA parameter of the CSQ6SYSP system parameter macro (described in Using CSQ6SYSP).

Within the data sharing group, shared Db2 tables are used to hold:

- Configuration information for the queue sharing group.
- Properties of IBM MQ shared and group objects.
- Optionally, data relating to offloaded IBM MQ messages.

IBM MQ provides a single set of sample jobs for defining the necessary Db2 table spaces, tables, and indexes. These jobs make use of Universal Table Spaces (UTS). Earlier versions of the product had two sets of jobs, one for UTS, and one for older types of table space, which have been deprecated by the most recent versions of Db2.

IBM MQ can still be used with older types of table space, and this might be appropriate if you already have an existing queue sharing group. However, if you are creating a new queue sharing group, it should use UTS.

Db2 V12 Function level 508 provides a non disruptive migration process for migrating multi-table table spaces to universal table spaces. You can use this approach to migrate the multi-table table spaces, used by existing queue sharing groups, to universal table spaces without taking an outage of the whole queue sharing group.

In Db2 V13, use the MOVE TABLE option of the ALTER TABLESPACE statement. See Moving tables from multi-table table spaces to partition-by-growth table spaces for more information.

By default Db2 uses the user ID of the person running the jobs as the owner of the Db2 resources. If this user ID is deleted then the resources associated with it are deleted, and so the table is deleted. Consider using a group ID to own the tables, rather than an individual user ID. You can do this by adding `GROUP=groupname` onto the JOB card, and specifying `SET CURRENT SQLID='groupname'` before any SQL statements.

IBM MQ uses the RRS Attach facility of Db2. This means that you can specify the name of a Db2 group that you want to connect to. The advantage of connecting to a Db2 group attach name (rather than a specific Db2 subsystem), is that IBM MQ can connect (or reconnect) to any available Db2 subsystem on the z/OS image that is a member of that group. There must be a Db2 subsystem that is a member of the data sharing group active on each z/OS image where you are going to run a queue-sharing IBM MQ subsystem, and RRS must be active.

Db2 storage

For most installations, the amount of Db2 storage required is about 20 or 30 cylinders on a 3390 device. However, if you want to calculate your storage requirement, the following table gives some information to help you determine how much storage Db2 requires for the IBM MQ data. The table describes the length of each Db2 row, and when each row is added to or deleted from the relevant Db2 table. Use this information together with the information about calculating the space requirements for the Db2 tables and their indexes in the *Db2 for z/OS Installation Guide*.

Table 25. Planning your Db2 storage requirements

Db2 table name	Length of row	A row is added when:	A row is deleted when:
CSQ.ADMIN_B_QSG	252 bytes	A queue sharing group is added to the table with the ADD QSG function of the CSQ5PQSG utility.	A queue sharing group is removed from the table with the REMOVE QSG function of the CSQ5PQSG utility. (All rows relating to this queue sharing group are deleted automatically from all the other Db2 tables when the queue sharing group record is deleted.)
CSQ.ADMIN_B_QMGR	Up to 3828 bytes	A queue manager is added to the table with the ADD QMGR function of the CSQ5PQSG utility.	A queue manager is removed from the table with the REMOVE QMGR function of the CSQ5PQSG utility.
CSQ.ADMIN_B_STRUCTURE	1454 bytes	The first local queue definition, specifying the QSGDISP(SHARED) attribute, that names a previously unknown structure within the queue sharing group is defined.	The last local queue definition, specifying the QSGDISP(SHARED) attribute, that names a structure within the queue sharing group is deleted.
CSQ.ADMIN_B_SCST	342 bytes	A shared channel is started.	A shared channel becomes inactive.
CSQ.ADMIN_B_SSKT	254 bytes	A shared channel that has the NPMSPEED(NORMAL) attribute is started.	A shared channel that has the NPMSPEED(NORMAL) attribute becomes inactive.
CSQ.ADMIN_B_STRBACKUP	514 bytes	A new row is added to the CSQ.ADMIN_B_STRUCTURE table. Each entry is a dummy entry until the BACKUP CFSTRUCT command is run, which overwrites the dummy entries.	A row is deleted from the CSQ.ADMIN_B_STRUCTURE table.
CSQ.OBJ_B_AUTHINFO	3400 bytes	An authentication information object with QSGDISP(GROUP) is defined.	An authentication information object with QSGDISP(GROUP) is deleted.
CSQ.OBJ_B_QUEUE	Up to 3707 bytes	• A queue with the QSGDISP(GROUP) attribute is defined. • A queue with the QSGDISP(SHARED) attribute is defined. • A model queue with the DEFTYPE(SHAREDYN) attribute is opened.	• A queue with the QSGDISP(GROUP) attribute is deleted. • A queue with the QSGDISP(SHARED) attribute is deleted. • A dynamic queue with the DEFTYPE(SHAREDYN) attribute is closed with the DELETE option.
CSQ.OBJ_B_NAMELIST	Up to 15127 bytes	A namelist with the QSGDISP(GROUP) attribute is defined.	A namelist with the QSGDISP(GROUP) attribute is deleted.

Table 25. Planning your Db2 storage requirements (continued)			
Db2 table name	**Length of row**	**A row is added when:**	**A row is deleted when:**
CSQ.OBJ_B_CHANNEL	Up to 14127 bytes	A channel with the QSGDISP(GROUP) attribute is defined.	A channel with the QSGDISP(GROUP) attribute is deleted.
CSQ.OBJ_B_STGCLASS	Up to 2865 bytes	A storage class with the QSGDISP(GROUP) attribute is defined.	A storage class with the QSGDISP(GROUP) attribute class is deleted.
CSQ.OBJ_B_PROCESS	Up to 3347 bytes	A process with the QSGDISP(GROUP) attribute is defined.	A process with the QSGDISP(GROUP) attribute is deleted.
CSQ.OBJ_B_TOPIC	Up to 14520 bytes	A topic object with QSGDISP(GROUP) attribute is defined.	A topic object with QSGDISP(GROUP) attribute is deleted.
CSQ.EXTEND_B_QMGR	Less than 430 bytes	A queue manager is added to the table with the ADD QMGR function of the CSQ5PQSG utility.	A queue manager is removed from the table with the REMOVE QMGR function of the CSQ5PQSG utility.
CSQ.ADMIN_B_MESSAGES	87 bytes	For large message PUT (1 per BLOB).	For large message GET (1 per BLOB).
CSQ.ADMIN_MSGS_BAUX1 CSQ.ADMIN_MSGS_BAUX2 CSQ.ADMIN_MSGS_BAUX3 CSQ.ADMIN_MSGS_BAUX4		These 4 tables contain message payload for large messages added into one of these 4 tables for each BLOB of the message. BLOBS are up to 511 KB in length, so if the message size is > 711 KB, there will be multiple BLOBs for this message.	

The use of large numbers of shared queue messages of size greater than 63 KB can have significant performance implications on your IBM MQ system. For more information, see SupportPac MP16, Capacity Planning and Tuning for IBM MQ for z/OS, at: SupportPacs for IBM MQ and other project areas.

z/OS Planning for backup and recovery

Developing backup and recovery procedures at your site is vital to avoid costly and time-consuming losses of data. IBM MQ provides means for recovering both queues and messages to their current state after a system failure.

This topic contains the following sections:

- "Example of queue manager backup activity" on page 180

Recovery procedures

Develop the following procedures for IBM MQ:

- Creating a point of recovery.
- Backing up page sets.
- Backing up CF structures.
- Recovering page sets.
- Recovering from out-of-space conditions (IBM MQ logs and page sets).
- Recovering CF structures.

See IBM MQ for z/OS 관리 for information about these.

Become familiar with the procedures used at your site for the following:

- Recovering from a hardware or power failure.
- Recovering from a z/OS component failure.
- Recovering from a site interruption, using off-site recovery.

z/OS Tips for backup and recovery

Use this topic to understand some backup and recovery tasks.

The queue manager restart process recovers your data to a consistent state by applying log information to the page sets. If your page sets are damaged or unavailable, you can resolve the problem using your backup copies of your page sets (if all the logs are available). If your log data sets are damaged or unavailable, it might not be possible to recover completely.

Consider the following points:

- Periodically take backup copies
- Do not discard archive logs you might need
- Do not change the DDname to page set association

Periodically take backup copies

A *point of recovery* is the term used to describe a set of backup copies of IBM MQ page sets and the corresponding log data sets required to recover these page sets. These backup copies provide a potential restart point in the event of page set loss (for example, page set I/O error). If you restart the queue manager using these backup copies, the data in IBM MQ is consistent up to the point that these copies were taken. Provided that all logs are available from this point, IBM MQ can be recovered to the point of failure.

The more recent your backup copies, the quicker IBM MQ can recover the data in the page sets. The recovery of the page sets is dependent on all the necessary log data sets being available.

In planning for recovery, you need to determine how often to take backup copies and how many complete backup cycles to keep. These values tell you how long you must keep your log data sets and backup copies of page sets for IBM MQ recovery.

When deciding how often to take backup copies, consider the time needed to recover a page set. The time needed is determined by the following:

- The amount of log to traverse.
- The time it takes an operator to mount and remove archive tape volumes.

- The time it takes to read the part of the log needed for recovery.
- The time needed to reprocess changed pages.
- The storage medium used for the backup copies.
- The method used to make and restore backup copies.

In general, the more frequently you make backup copies, the less time recovery takes, but the more time is spent making copies.

For each queue manager, you should take backup copies of the following:

- The archive log data sets
- The BSDS copies created at the time of the archive
- The page sets
- Your object definitions
- Your CF structures

To reduce the risk of your backup copies being lost or damaged, consider:

- Storing the backup copies on different storage volumes to the original copies.
- Storing the backup copies at a different site to the original copies.
- Making at least two copies of each backup of your page sets and, if you are using single logging or a single BSDS, two copies of your archive logs and BSDS. If you are using dual logging or BSDS, make a single copy of both archive logs or BSDS.

Before moving IBM MQ to a production environment, fully test and document your backup procedures.

Backing up your page sets

You need to back up page sets regularly. Some enterprises back up the page sets twice a day.

You need the active and archive logs since a backup to be able to recover using the backup. You need enough log data to go back four checkpoints if the backup was taken when the queue manager was running.

You can use ADRDSSU FastReplication to back up page sets, and you can do this while the queue manager is active. Note that you need to ensure there is enough space in the storage pool.

Backing up your object definitions

Create backup copies of your object definitions. To do this, use the MAKEDEF feature of the COMMAND function of the utility program (described in Using the COMMAND function of CSQUTIL).

You should do this whenever you take backup copies of your queue manager data sets, and keep the most current version.

Backing up your coupling facility structures

If you have set up any queue sharing groups, even if you are not using them, you must take periodic backups of your CF structures. To do this, use the IBM MQ BACKUP CFSTRUCT command. You can use this command only on CF structures that are defined with the RECOVER(YES) attribute. If any CF entries for persistent shared messages refer to offloaded message data stored in a shared message data set (SMDS) or Db2, the offloaded data is retrieved and backed up together with the CF entries. Shared message data sets should not be backed up separately.

It is recommended that you take a backup of all your CF structures about every hour, to minimize the time it takes to restore a CF structure.

You could perform all your CF structure backups on a single queue manager, which has the advantage of limiting the increase in log use to a single queue manager. Alternatively, you could perform backups on all the queue managers in the queue sharing group, which has the advantage of spreading the workload across the queue sharing group. Whichever strategy you use, IBM MQ can locate the backup

and perform a RECOVER CFSTRUCT from any queue manager in the queue sharing group. The logs of all the queue managers in the queue sharing group need to be accessed to recover the CF structure.

Backing up your message security policies

If you are using Advanced Message Security to create a backup of your message security policies, create a backup using the message security policy utility (CSQ0UTIL) to run **dspmqspl** with the -export parameter, then save the policy definitions that are output to the EXPORT DD.

You should create a backup of your message security policies whenever you take backup copies of your queue manager data sets, and keep the most current version.

Do not discard archive logs you might need

IBM MQ might need to use archive logs during restart. You must keep sufficient archive logs so that the system can be fully restored. IBM MQ might use an archive log to recover a page set from a restored backup copy. If you have discarded that archive log, IBM MQ cannot restore the page set to its current state. When and how you discard archive logs is described in Discarding archive log data sets.

You can use the `/cpf DIS USAGE TYPE(ALL)` command to display the log RBA, and log range sequence number (LRSN) that you need to recover your queue manager's page sets and the queue sharing group's structures. You should then use the print log map utility (CSQJU004) to print bootstrap data set (BSDS) information for the queue manager to locate the logs containing the log RBA.

For CF structures, you need to run the CSQJU004 utility on each queue manager in the queue sharing group to locate the logs containing the LRSN. You need these logs and any later logs to be able to recover the page sets and structures.

Do not change the DDname to page set association

IBM MQ associates page set number 00 with DDname CSQP0000, page set number 01 with DDname CSQP0001, and so on, up to CSQP0099. IBM MQ writes recovery log records for a page set based on the DDname that the page set is associated with. For this reason, you must not move page sets that have already been associated with a PSID DDname.

z/OS Recovering page sets

Use this topic to understand the factors involved when recovering pages sets, and how to minimize restart times.

A key factor in recovery strategy concerns the time for which you can tolerate a queue manager outage. The total outage time might include the time taken to recover a page set from a backup, or to restart the queue manager after an abnormal termination. Factors affecting restart time include how frequently you back up your page sets, and how much data is written to the log between checkpoints.

To minimize the restart time after an abnormal termination, keep units of work short so that, at most, two active logs are used when the system restarts. For example, if you are designing an IBM MQ application, avoid placing an MQGET call that has a long wait interval between the first in-syncpoint MQI call and the commit point because this might result in a unit of work that has a long duration. Another common cause of long units of work is batch intervals of more than 5 minutes for the channel initiator.

You can use the DISPLAY THREAD command to display the RBA of units of work and to help resolve the old ones.

How often must you back up a page set?

Frequent page set backup is essential if a reasonably short recovery time is required. This applies even when a page set is very small or there is a small amount of activity on queues in that page set.

If you use persistent messages in a page set, the backup frequency should be in hours rather than days. This is also the case for page set zero.

To calculate an approximate backup frequency, start by determining the target total recovery time. This consists of the following:

1. The time taken to react to the problem.
2. The time taken to restore the page set backup copy.

   If you use SnapShot backup/restore, the time taken to perform this task is a few seconds. For information about SnapShot, see the *DFSMSdss Storage Administration Guide*.
3. The time the queue manager requires to restart, including the additional time needed to recover the page set.

   This depends most significantly on the amount of log data that must be read from active and archive logs since that page set was last backed up. All such log data must be read, in addition to that directly associated with the damaged page set.

   Note: When using *fuzzy backup* (where a snapshot is taken of the logs and page sets while a unit of work is active), it might be necessary to read up to three additional checkpoints, and this might result in the need to read one or more additional logs.

When deciding on how long to allow for the recovery of the page set, the factors that you need to consider are:

- The rate at which data is written to the active logs during normal processing depends on how messages arrive in your system, in addition to the message rate.

  Messages received or sent over a channel result in more data logging than messages generated and retrieved locally.
- The rate at which data can be read from the archive and active logs.

  When reading the logs, the achievable data rate depends on the devices used and the total load on your particular DASD subsystem.

  With most tape units, it is possible to achieve higher data rates for archived logs with a large block size. However, if an archive log is required for recovery, all the data on the active logs must be read also.

z/OS Recovering CF structures

Use this topic to understand the recovery process for CF structures.

At least one queue manager in the queue sharing group must be active to process a RECOVER CFSTRUCT command. CF structure recovery does not affect queue manager restart time, because recovery is performed by an already active queue manager.

The recovery process consists of two logical steps that are managed by the RECOVER CFSTRUCT command:

1. Locating and restoring the backup.
2. Merging all the logged updates to persistent messages that are held on the CF structure from the logs of all the queue managers in the queue sharing group that have used the CF structure, and applying the changes to the backup.

The second step is likely to take much longer because a lot of log data might need to be read. You can reduce the time taken if you take frequent backups, or if you recover multiple CF structures at the same time, or both.

The queue manager performing the recovery locates the relevant backups on all the other queue managers' logs using the data in Db2 and the bootstrap data sets. The queue manager replays these backups in the correct time sequence across the queue sharing group, from just before the last backup through to the point of failure.

The time it takes to recover a CF structure depends on the amount of recovery log data that must be replayed, which in turn depends on the frequency of the backups. In the worst case, it takes as long to

read a queue manager's log as it did to write it. So if, for example, you have a queue sharing group containing six queue managers, an hour's worth of log activity could take six hours to replay. In general it takes less time than this, because reading can be done in bulk, and because the different queue manager's logs can be read in parallel. As a starting point, we recommend that you back up your CF structures every hour.

All queue managers can continue working with non-shared queues and queues in other CF structures while there is a failed CF structure. If the administration structure has also failed, at least one of the queue managers in the queue sharing group must be started before you can issue the RECOVER CFSTRUCT command.

Backing up CF structures can require considerable log writing capacity, and can therefore impose a large load on the queue manager doing the backup. Choose a lightly loaded queue manager for doing backups; for busy systems, add an additional queue manager to the queue sharing group and dedicate it exclusively for doing backups.

z/OS Achieving specific recovery targets

Use this topic for guidance on how you can achieve specific recovery target times by adjusting backup frequency.

If you have specific recovery targets to achieve, for example, completion of the queue manager recovery and restart processing in addition to the normal startup time within *xx* seconds, you can use the following calculation to estimate your backup frequency (in hours):

```
Formula (A)
                    Required restart time * System recovery log read rate
                          (in secs)                  (in MB/sec)
Backup frequency  = ------------------------------------------------------
   (in hours)              Application log write rate (in MB/hour)
```

Note: The examples given next are intended to highlight the need to back up your page sets frequently. The calculations assume that most log activity is derived from a large number of persistent messages. However, there are situations where the amount of log activity is not easily calculated. For example, in a queue sharing group environment, a unit of work in which shared queues are updated in addition to other resources might result in UOW records being written to the IBM MQ log. For this reason, the Application log write rate in Formula (A) can be derived accurately only from the observed rate at which the IBM MQ logs fill.

For example, consider a system in which IBM MQ MQI clients generate a total load of 100 persistent messages a second. In this case, all messages are generated locally.

If each message is of user length 1 KB, the amount of data logged each hour is approximately:

```
100 * (1 + 1.3) KB * 3600 = approximately 800 MB

where
      100           = the message rate a second
      (1 + 1.3) KB  = the amount of data logged for
                      each 1 KB of persistent messages
```

Consider an overall target recovery time of 75 minutes. If you have allowed 15 minutes to react to the problem and restore the page set backup copy, queue manager recovery and restart must then complete within 60 minutes (3600 seconds) applying formula (A). Assuming that all required log data is on RVA2-T82 DASD, which has a recovery rate of approximately 2.7 MB a second, this necessitates a page set backup frequency of at least every:

```
3600 seconds * 2.7 MB a second / 800 MB an hour = 12.15 hours
```

If your IBM MQ application day lasts approximately 12 hours, one backup each day is appropriate. However, if the application day lasts 24 hours, two backups each day is more appropriate.

Another example might be a production system in which all the messages are for request-reply applications (that is, a persistent message is received on a receiver channel and a persistent reply message is generated and sent down a sender channel).

In this example, the achieved batch size is one, and so there is one batch for every message. If there are 50 request replies a second, the total load is 100 persistent messages a second. If each message is 1 KB in length, the amount of data logged each hour is approximately:

```
50((2 * (1+1.3) KB) + 1.4 KB + 2.5 KB) * 3600 = approximately 1500 MB

where:
 50                 = the message pair rate a second
 (2 * (1 + 1.3) KB) = the amount of data logged for each message pair
 1.4 KB             = the overhead for each batch of messages
                      received by each channel
 2.5 KB             = the overhead for each batch of messages sent
                      by each channel
```

To achieve the queue manager recovery and restart within 30 minutes (1800 seconds), again assuming that all required log data is on RVA2-T82 DASD, this requires that page set backup is carried out at least every:

```
1800 seconds * 2.7 MB a second / 1500 MB an hour = 3.24 hours
```

Periodic review of backup frequency

Monitor your IBM MQ log usage in terms of MB an hour. Periodically perform this check and amend your page set backup frequency if necessary.

z/OS Backup considerations for other products

If you are using IBM MQ with CICS or IMS then you must also consider the implications for your backup strategy with those products. The data facility hierarchical storage manager (DFHSM) manages data storage, and can interact with the storage used by IBM MQ.

Backup and recovery with DFHSM

The data facility hierarchical storage manager (DFHSM) does automatic space-availability and data-availability management among storage devices in your system. If you use it, you need to know that it moves data to and from the IBM MQ storage automatically.

DFHSM manages your DASD space efficiently by moving data sets that have not been used recently to alternative storage. It also makes your data available for recovery by automatically copying new or changed data sets to tape or DASD backup volumes. It can delete data sets, or move them to another device. Its operations occur daily, at a specified time, and allow for keeping a data set for a predetermined period before deleting or moving it.

You can also perform all DFHSM operations manually. For more information on DFHSM, see the z/OS DFSMS product documentation. If you use DFHSM with IBM MQ, note that DFHSM does the following:

- Uses cataloged data sets.
- Operates on page sets and logs.
- Supports VSAM data sets.

Recovery and CICS

The recovery of CICS resources is not affected by the presence of IBM MQ. CICS recognizes IBM MQ as a non-CICS resource (or external resource manager), and includes IBM MQ as a participant in any syncpoint coordination requests using the CICS resource manager interface (RMI). For more information about CICS recovery and the CICS resource manager interface, see the CICS product documentation.

Recovery and IMS

IMS recognizes IBM MQ as an external subsystem and as a participant in syncpoint coordination. IMS recovery for external subsystem resources is described in the IMS product documentation.

z/OS Preparing for recovery on an alternative site

If a total loss of an IBM MQ computing center, you can recover on another IBM MQ system at a recovery site.

To recover an IBM MQ system at a recovery site, you must regularly back up the page sets and the logs. As with all data recovery operations, the objectives of disaster recovery are to lose as little data, workload processing (updates), and time as possible.

At the recovery site:

- The recovery IBM MQ queue manager **must** have the same name as the lost queue manager.
- Ensure the system parameter module used on the recovery queue manager contains the same parameters as the lost queue manager.

See Administering IBM MQ for z/OS and Troubleshooting IBM MQ for z/OS problems for more information.

z/OS Example of queue manager backup activity

This topic shows as an example of queue manager backup activity.

When you plan your queue manager backup strategy, a key consideration is retention of the correct amount of log data. Managing the logs describes how to determine which log data sets are required, by reference to the system recovery RBA of the queue manager. IBM MQ determines the system recovery RBA using information about the following:

- Currently active units of work.
- Page set updates that have not yet been flushed from the buffer pools to disk.
- CF structure backups, and whether this queue manager's log contains information required in any recovery operation using them.

You must retain sufficient log data to be able to perform media recovery. While the system recovery RBA increases over time, the amount of log data that must be retained only decreases when subsequent backups are taken. CF structure backups are managed by IBM MQ, and so are taken into account when reporting the system recovery RBA. This means that in practice, the amount of log data that must be retained only reduces when page set backups are taken.

Figure 43 on page 181 shows an example of the backup activity on a queue manager that is a member of a queue sharing group, how the recovery RBA varies with each backup, and how that affects the amount of log data that must be retained. In the example the queue manager uses local and shared resources: page sets, and two CF structures, STRUCTURE1 and STRUCTURE2.

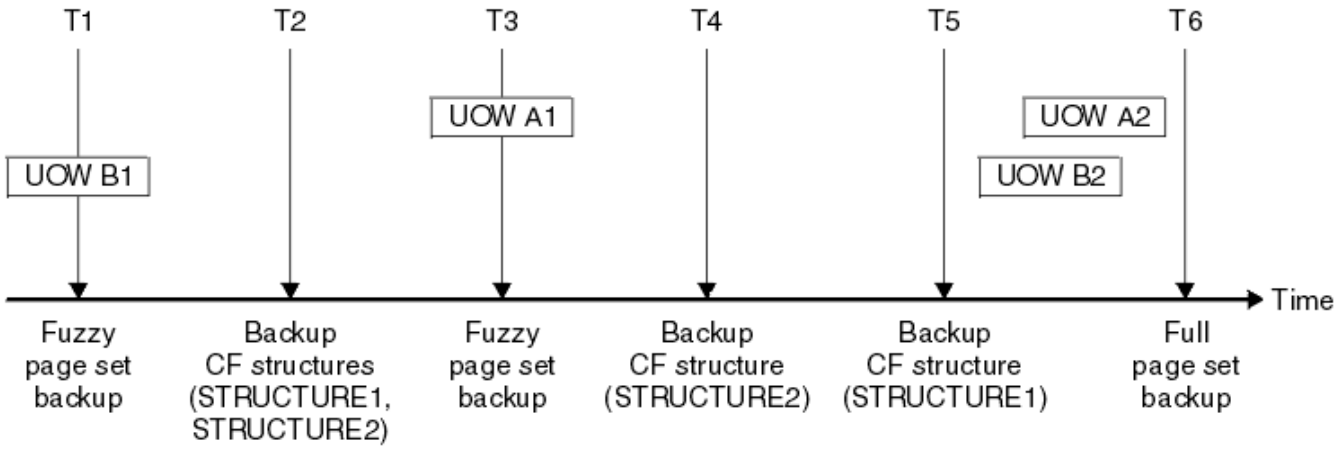


Figure 43. Example of queue manager backup activity

This is what happens at each point in time:

Point in time T1

A fuzzy backup is created of your page sets, as described in How to back up and recover page sets.

The system recovery RBA of the queue manager is the lowest of the following:

- The recovery RBAs of the page sets being backed up at this point.
- The lowest recovery RBA required to recover the CF application structures. This relates to the recovery of backups of STRUCTURE1 and STRUCTURE2 created earlier.
- The recovery RBA for the oldest currently active unit of work within the queue manager (UOWB1).

The system recovery RBA for this point in time is given by messages issued by the DISPLAY USAGE command, which is part of the fuzzy backup process.

Point in time T2

Backups of the CF structures are created. CF structure STRUCTURE1 is backed up first, followed by STRUCTURE2.

The amount of log data that must be retained is unchanged, because the same data as determined from the system recovery RBA at T1 is still required to recover using the page set backups taken at T1.

Point in time T3

Another fuzzy backup is created.

The system recovery RBA of the queue manager is the lowest of the following:

- The recovery RBAs of the page sets being backed up at this point.
- The lowest recovery RBA required to recover CF structure STRUCTURE1, because STRUCTURE1 was backed up before STRUCTURE2.
- The recovery RBA for the oldest currently active unit of work within the queue manager (UOWA1).

The system recovery RBA for this point in time is given by messages issued by the DISPLAY USAGE command, which is part of the fuzzy backup process.

You can now reduce the log data retained, as determined by this new system recovery RBA.

Point in time T4

A backup is taken of CF structure STRUCTURE2. The recovery RBA for the recovery of the oldest required CF structure backup relates to the backup of CF structure STRUCTURE1, which was backed up at time T2.

The creation of this CF structure backup has no effect on the amount of log data that must be retained.

Point in time T5

A backup is taken of CF structure STRUCTURE1. The recovery RBA for recovery of the oldest required CF structure backup now relates to recovery of CF structure STRUCTURE2, which was backed up at time T4.

The creation of this CF structure backup has no effect on amount of log data that must be retained.

Point in time T6

A full backup is taken of your page sets as described in How to back up and recover page sets.

The system recovery RBA of the queue manager is the lowest of the following:

- The recovery RBAs of the page sets being backed up at this point.
- The lowest recovery RBA required to recover the CF structures. This relates to recovery of CF structure STRUCTURE2.
- The recovery RBA for the oldest currently active unit of work within the queue manager. In this case, there are no current units of work.

The system recovery RBA for this point in time is given by messages issued by the DISPLAY USAGE command, which is part of the full backup process.

Again, the log data retained can be reduced, because the system recovery RBA associated with the full backup is more recent.

z/OS Planning your z/OS UNIX environment

Certain processes within the IBM MQ queue manager, channel initiator, and mqweb server use z/OS UNIX System Services (z/OS UNIX) for their normal processing.

The queue manager and channel initiator started task user IDs need an OMVS segment with a UID defined in order to be able to access z/OS UNIX. The user IDs require no special permissions in z/OS UNIX.

Note: Although the queue manager and channel initiator make use of z/OS UNIX facilities (for example, to interface with TCP/IP services), they do not need to access any of the content of the IBM MQ installation directory in the z/OS UNIX file system. As a result, the queue manager and channel initiator do not require any configuration to specify the path for the z/OS UNIX file system.

The mqweb server, which hosts the IBM MQ Console and REST API, makes use of files in the IBM MQ installation directory in the z/OS UNIX file system. It also needs access to another file system which is used to store data such as configuration and log files. The mqweb started task JCL needs to be customized to reference these z/OS UNIX file systems.

The content of the IBM MQ directory in the z/OS UNIX file system is also used by applications connecting to IBM MQ. For example, applications using the IBM MQ classes for Java or IBM MQ classes for JMS interfaces.

See the following topics for the relevant configuration instructions:

- Environment variables relevant to IBM MQ classes for Java
- IBM MQ classes for Java libraries
- Setting environment variables
- Configuring the Java Native Interface (JNI) libraries

z/OS Planning for Advanced Message Security

TLS (or SSL) can be used to encrypt and protect messages flowing on a network, but this does not protect messages when they are on a queue ("at rest"). Advanced Message Security (AMS) protects the messages from the time that they are first put to a queue, until they are got, so that only the intended recipients of the message can read that message. The messages are encrypted and signed during put processing, and unprotected during get processing.

AMS can be configured to protect messages in different ways:

1. A message can be signed. The message is in clear text, but there is a checksum, which is signed. This allows any changes in the message content to be detected. From the signed content, you can identify who signed the data.
2. A message can be encrypted. The contents are not visible to anyone without the decryption key. The decryption key is encrypted for each recipient.
3. A message can be encrypted and signed. They decryption key is encrypted for each recipient, and from the signing you can identify who sent the message.

The encryption and signing use digital certificates and key rings.

You can set up a client to use AMS, so the data is protected before the data is put on the client channel. Protected messages can be sent to a remote queue manager, and you need to configure the remote queue manager to process these messages.

Setting up AMS

An AMS address space is used for doing the AMS work. This has additional security set up, to give access to and protect the use of key rings and certificates.

You configure which queues are to be protected by using a utility program (CSQ0UTIL) to define the security policies for queues.

Once AMS is set up

You need to set up a digital certificate and a key ring for people who put messages, and the people who get messages.

If a user, Alice, on z/OS needs to send a message to Bob, AMS needs a copy of the public certificate for Bob.

If Bob wants to process a message from Alice, AMS needs the public certificate for Alice, or the same certificate authority certificate used by Alice.

Attention: You need to:

- Carefully plan who can put to, or get from, queues
- Identify the people and their certificate names.

It is easy to make mistakes, and problems can be hard to resolve.

Related concepts

"Planning for your queue manager" on page 128
When you are setting up a queue manager, your planning should allow for the queue manager to grow, so that the queue manager meets the needs of your enterprise.

z/OS Planning for Managed File Transfer

Use this section as guidance on how you need to set up your system to run Managed File Transfer (MFT) on z/OS.

z/OS Planning for Managed File Transfer - hardware and software requirements

Use this topic as guidance on how you need to set up hardware and software requirements on your system to run Managed File Transfer (MFT) on z/OS.

Software requirements

Managed File Transfer is written in Java, with some shell scripts and JCL to configure and operate the program.

Important: You must be familiar with z/OS UNIX System Services (z/OS UNIX) in order to configure Managed File Transfer. For example:

- The file directory structure, with names such as `/u/userID/myfile.txt`
- z/OS UNIX commands, for example:
  - `cd` (change directory)
  - `ls` (list)
  - `chmod` (change the file permissions)
  - `chown` (change file ownership or groups which can access the file or directory)

You require the following products in z/OS UNIX to be able to configure and run MFT:

1. Java, for example, in directory `/java/java80_bit64_GA/J8.0_64/`
2. IBM MQ 9.4.0, for example, in directory `/mqm/V9R3M0`
3. If you want to use Db2 for status and history, you need to install Db2 JDBC libraries, for example, in directory `/db2/db2v10/jdbc/libs`.

Product registration

At startup Managed File Transfer checks the registration in `sys1.parmlib`(IFAPRDxx) concatenation. The following code is an example of how you register MFT:

```
PRODUCT OWNER('IBM CORP')
NAME('WS MQ FILE TRANS')
ID(5655-MFT)
VERSION(*) RELEASE(*) MOD(*)
FEATURENAME('WS MQ FILE TRANS')
STATE(ENABLED)
```

Disk space

The IBM MQ for z/OS Program Directory states the DASD and zFS storage requirements for Managed File Transfer. For download links for the Program Directory for IBM MQ for z/OS, see IBM MQ 9.4 PDF files for product documentation and Program Directories.

z/OS Planning for Managed File Transfer - topologies

Use this topic as guidance on what topology you need on your system to run Managed File Transfer (MFT) on z/OS.

Managed File Transfer queue managers

IBM MQ Managed File Transfer topologies consist of:

Agents, and their associated queue managers
The agent uses system queues hosted on their agent queue manager to maintain state information and receive requests for work.

A command queue manager
This acts as a gateway into an MFT topology. It is connected to the agent queue managers through either sender and receiver channels, or clustering. When certain commands are run, they connect directly to the command queue manager, and send a message to the specified agent. This message is routed through the IBM MQ network to the agent queue manager, where it is picked up by the agent and processed.

A coordination queue manager
This is a central hub that has knowledge of the entire topology. The coordination queue manager is connected to all of the agent queue managers in a topology through either sender and receiver

channels, or using clustering. Agents regularly publish status information to the coordination queue manager, and store their transfer templates there.

It is possible for a single queue manager to perform multiple roles within a topology. For example, the same queue manager can be configured as both the coordination queue manager and the command queue manager for a topology.

If you are using multiple queue managers you need to set up channels between the queue managers. You can either do this by using clustering or by using point-to-point connections.

When using IBM MQ Managed File Transfer for z/OS, there are a number of things to consider when determining which queue managers to use for the different roles within a topology.

Agent queue managers

The agent queue manager for an IBM MQ Managed File Transfer for z/OS agent must be running on z/OS.

If:

- The agent is running Managed File Transfer for z/OS on IBM MQ 9.1 or later
- And, the agent queue manager is licensed for IBM MQ Advanced for z/OS Value Unit Edition (Advanced VUE)

the agent can connect to the queue manager using the CLIENT transport.

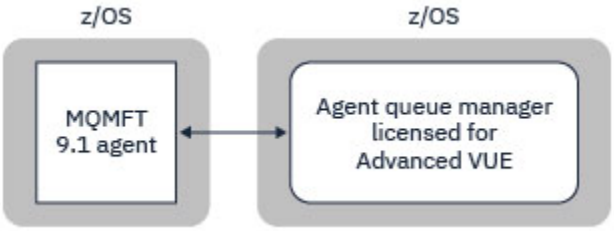


Figure 44. MFT 9.1 agents on z/OS can connect to a queue manager using the CLIENT transport, assuming the queue manager is licensed for Advanced VUE.

If:

- The agent is running Managed File Transfer for z/OS on IBM MQ 9.0 or earlier
- Or, the agent queue manager is running Managed File Transfer for z/OS on IBM MQ 9.0 or later, and the agent queue manager is licensed for either MFT, IBM MQ Advanced for z/OS, or Advanced VUE

the agent must connect to the queue manager using the BINDINGS transport.

Figure 45. MFT 9.0 agents on z/OS and 9.1 agents that have an agent queue manager licensed for either MFT or IBM MQ Advanced, must connect using the BINDINGS transport.

Command queue managers

The Which MFT commands and processes connect to which queue manager topic shows all of the commands that connect to the command queue manager for a Managed File Transfer topology.

Note: When running these commands on z/OS, the command queue manager must also be on z/OS.

If the command queue manager is licensed for Advanced VUE, the commands can connect to the queue manger using the CLIENT transport. Otherwise, the commands must connect to the command queue manager using the BINDINGS transport.

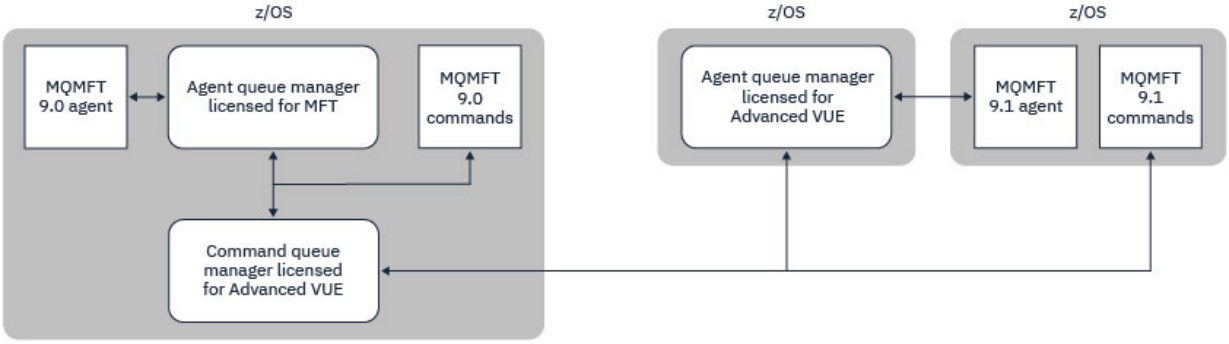


Figure 46. Commands connect to the command queue manager for an MFT topology. When running these commands on z/OS, the command queue manager must also be on z/OS

Coordination queue managers

IBM MQ Managed File Transfer for z/OS agents can be part of a topology where the coordination queue manager is either running on z/OS, or is running on a multiplatform.

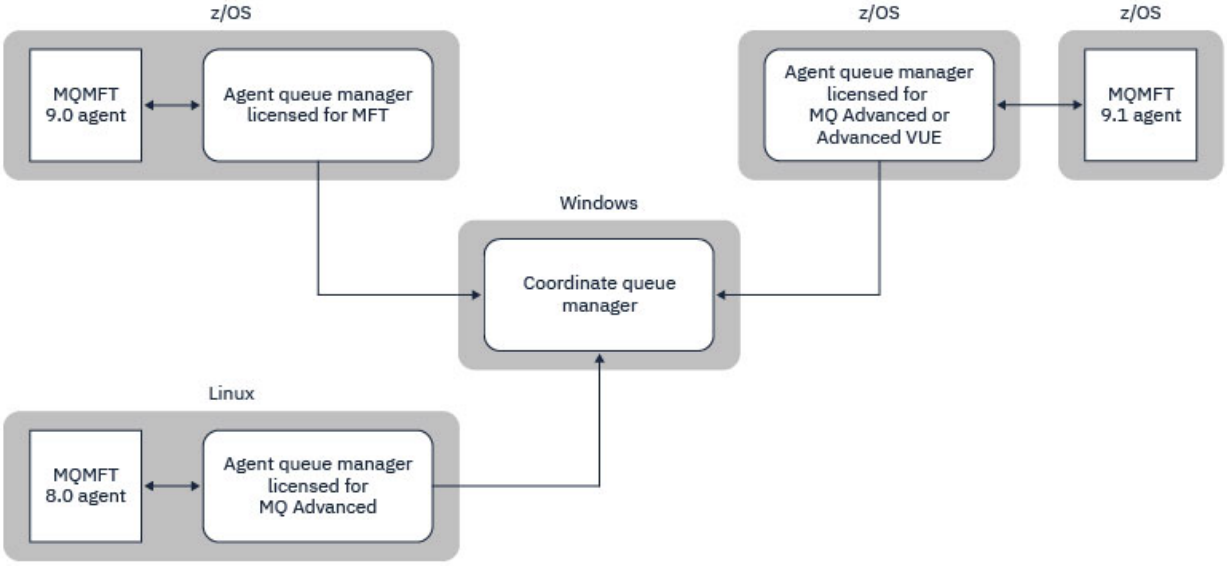


Figure 47. MFT agents running on z/OS can be part of an MFT topology where the coordination queue manager is running on an IBM MQ multiplatform.

The Which MFT commands and processes connect to which queue manager topic shows the commands that connect to the coordination queue manager for a Managed File Transfer topology. It is possible to run these commands on z/OS and have then connect to the coordination queue manager running on a different platform.

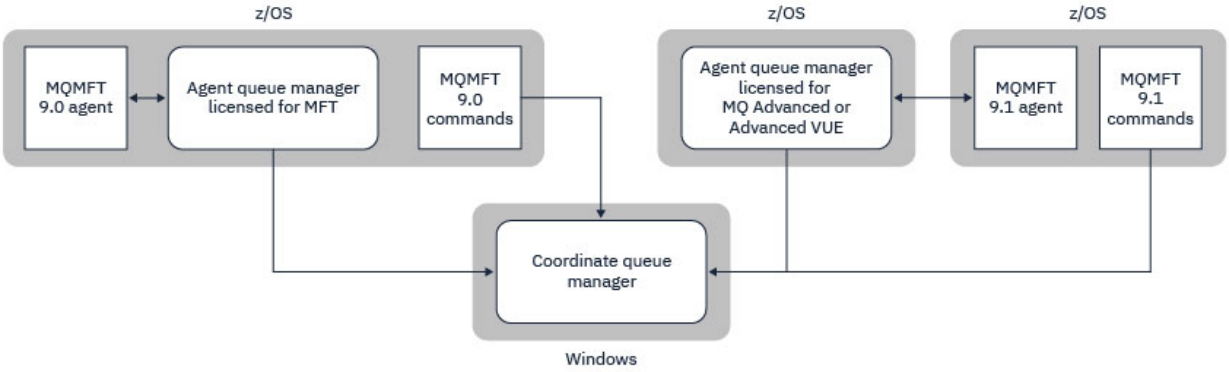


*Figure 48. Certain commands, such as **fteListAgents**, connect directly to the coordination queue manager for an MFT topology.*

How many agents do I need?

The agents do the work in transferring data, and when you make a request to transfer data you specify the name of an agent.

By default an agent can process 25 send and 25 receive requests concurrently. You can configure these processes. See Managed File Transfer configuration options on z/OS for more information.

If the agent is busy then work is queued. The time taken to process a request depends on multiple factors, for example, the amount of data to be sent, the network bandwidth, and the delay on the network.

You might want to have multiple agents to process work in parallel.

You can also control which resources an agent can access, so you might want some agents to work with a limited subset of data.

If you want to process requests with different priority you can use multiple agents and use workload manager to set the priority of the jobs.

Running the agents

Typically the agents are long running processes. The processes can be submitted as jobs that run in batch, or as started tasks.

z/OS Planning for Managed File Transfer - security considerations

Use this topic as guidance on what security considerations you need on your system to run Managed File Transfer (MFT) on z/OS.

Security

You need to identify which user IDs are going to be used for MFT configuration and for MFT operation.

You need to identify the files or queues you transfer, and which user IDs are going to be submitting transfer requests to MFT.

When you customize the agents and logger, you specify the group of users that is allowed to run MFT services, or do MFT administration.

You should set up this group before you start customizing MFT. As MFT uses IBM MQ queues, if you have security enabled in the queue manager, MFT requires access to the following resources:

Table 26. MQADMIN resource class

Name	Access required
QUEUE.SYSTEM.FTE.EVENT.agent_name	Update

Table 26. MQADMIN resource class (continued)	
Name	**Access required**
QUEUE.SYSTEM.FTE.COMMAND.agent_name	Update
CONTEXT.SYSTEM.FTE.COMMAND.agent_name	Update
QUEUE.SYSTEM.FTE.STATE.agent_name	Update
QUEUE.SYSTEM.FTE.DATA.agent_name	Update
QUEUE.SYSTEM.FTE.REPLY.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHAGT1.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHTRN1.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHOPS1.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHSCH1.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHMON1.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHADM1.agent_name	Update

Table 27. MQQUEUE resource class	
Name	**Access required**
SYSTEM.FTE.AUTHAGT1.agent_name	Update
SYSTEM.FTE.AUTHTRN1.agent_name	Update
SYSTEM.FTE.AUTHOPS1.agent_name	Update
SYSTEM.FTE.AUTHSCH1.agent_name	Update
SYSTEM.FTE.AUTHMON1.agent_name	Update

You can use user sandboxing to determine which parts of the file system the user who requests the transfer can access.

To enable user sandboxing, add the `userSandboxes=true` statement to the *agent.properties* file for the agent that you want to restrict, and add appropriate values to the `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name/UserSandboxes.xml` file.

See Working with user sandboxes for further information.

This user ID is configured in `UserSandboxes.xml` files.

This XML file has information like user ID, or user ID* and a list of resource that can be used (included), or cannot be used (excluded). You need to define specific user IDs that can access which resources: for example:

Table 28. Example user ID together with access to specific resources			
User ID	**Access**	**Include or Exclude**	**Resource**
Admin*	Read	Include	`/home/user/**`
Admin*	Read	Exclude	`/home/user/private/**`
Sysprog	Read	Include	`/home/user/**`
Admin*	Read	Include	Application.reply.queue

Notes:

1. If `type=queue` is specified, the resource is either a queue name, or `queue@qmgr`.
2. If the resource begins with `//`, the resource is a data set; otherwise the resource is a file in z/OS UNIX.
3. The user ID is the user ID from the MQMD structure, so this might not reflect the user ID that actually puts the message.
4. For requests on the local queue manager you can use MQADMIN CONTEXT.* to limit which users can set this value.
5. For requests coming in over a remote queue manager, you have to assume that the distributed queue managers have security enabled to prevent unauthorized setting of the user ID in the MQMD structure.
6. A user ID of SYSPROG1 on a Linux machine, is the same user ID SYSPROG1 for the security checking on z/OS.

z/OS Planning to use the IBM MQ Console and REST API on z/OS

The IBM MQ Console and REST API are applications that run in a WebSphere Liberty (Liberty) server known as mqweb. The mqweb server runs as a started task. The IBM MQ Console allows a web browser to be used to administer queue managers. The REST API provides a simple programmatic interface for applications to do queue manager administration, and to perform messaging.

Installation and configuration files

You need to install the IBM MQ for z/OS UNIX System Services Web Components feature, which will install the files needed to run the mqweb server in z/OS UNIX System Services (z/OS UNIX). You need to be familiar with z/OS UNIX to be able to configure and manage the mqweb server.

See IBM MQ for z/OS Program Directory PDF files for information on installing IBM MQ for z/OS UNIX System Services Components.

The IBM MQ files in z/OS UNIX are installed with various attributes set that are required for the correct operation of the mqweb server. If you need to copy the IBM MQ z/OS UNIX installation files, for example if you have installed IBM MQ on one system, and run IBM MQ on a different system, you should copy the IBM MQ ZFS created during the installation, and mount it read only at the destination. Copying the files in other ways might cause some file attributes to be lost.

You need to decide upon the location for, and create, a Liberty user directory when you create the mqweb server. This directory contains configuration and log files, and the location can be something similar to `/var/mqm/mqweb`.

Using the IBM MQ Console and REST API with queue managers at different levels

The REST API can directly interact only with queue managers that run at the same Version, Release, and Modification (VRM) as the mqweb server which runs the REST API. For example, the IBM MQ 9.4.0 REST API can directly interact only with local queue managers at IBM MQ 9.4.0, and the IBM MQ 9.3.5 REST API can directly interact only with local queue managers at IBM MQ 9.3.5.

You can use the REST API to administer a queue manager at a different version from the mqweb server by configuring a gateway queue manager. However, you need at least one queue manager at the same version as the mqweb server to act as the gateway queue manager. For more information, see Remote administration using the REST API.

The IBM MQ Console can be used to manage local queue managers that run at the same version as the IBM MQ Console. From IBM MQ 9.3.0, you can also use the IBM MQ Console to administer a queue manager running on a remote system, or at a different version to the IBM MQ Console. For more information, see IBM MQ Console: Adding a remote queue manager.

Migration

If you have only one queue manager, you can run the mqweb server as a single started task, and change the libraries it uses when you migrate your queue manager.

If you have more than one queue manager, during migration you can start mqweb servers at different versions by using started tasks with different names. These names can be any name you want. For example, you can start an IBM MQ 9.3.0 mqweb server using a started task named MQWB0930, and an IBM MQ 9.3.5 mqweb server using a started task named MQWB0935.

Then, when you migrate the queue managers from one version to a later version, the queue managers become available in the mqweb server for the later version, and are no longer available in the mqweb server for the earlier version.

After you have migrated all the queue managers to the later version, you can delete the mqweb server for the earlier version.

HTTP ports

The mqweb server uses up to two ports for HTTP:

- One for HTTPS, with a default value of 9443.
- One for HTTP. HTTP is not enabled by default, but if enabled, has a default value of 9080.

If the default port values are in use, you must allocate other ports. If you have more than one mqweb server running simultaneously for more than one version of IBM MQ, you must allocate separate ports for each version. For more information on setting the ports that the mqweb server uses, see Configuring the HTTP and HTTPS ports.

You can use the following TSO command to display information about a port:

```
NETSTAT TCP tcpip (PORT portNumber)
```

where *tcpip* is the name of the TCP/IP address space, and *portNumber* specifies the number of the port to display information about.

Security - starting the mqweb server

The mqweb server user ID needs certain authorities. For more information, see Authority required by the mqweb server started task user ID.

Security - using the IBM MQ Console and REST API

When you use the IBM MQ Console and REST API, you must authenticate as a user that is included in a configured registry. These users are assigned specific roles that determine the actions the users can perform. For example, to use the messaging REST API, a user must be assigned the `MQWebUser` role. For more information about the available roles for the IBM MQ Console and REST API, and the access that these roles grant, see Roles on the IBM MQ Console and REST API.

For more information about configuring security for the IBM MQ Console and REST API, see IBM MQ Console and REST API security.

주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다.

IBM은 다른 국가에서 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산권을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이센스까지 부여하는 것은 아닙니다. 라이센스에 대한 의문사항은 다음으로 문의하십시오.

07326
서울특별시 영등포구
국제금융로 10, 3IFC
한국 아이.비.엠 주식회사
U.S.A.

2바이트(DBCS) 정보에 관한 라이센스 문의는 한국 IBM에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

Intellectual Property Licensing
2-31 Roppongi 3-chome, Minato-Ku
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다. IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 명시적 또는 묵시적인 일체의 보증 없이 이 책을 "현상태대로" 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함) 간의 정보 교환 및 (ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이센스 사용자는 다음 주소로 문의하십시오.

서울특별시 영등포구
서울특별시 강남구 도곡동 467-12,
군인공제회관빌딩
한국 아이.비.엠 주식회사
U.S.A.

이러한 정보는 해당 조건(예를 들면, 사용료 지불 등)하에서 사용될 수 있습니다.

이 정보에 기술된 라이센스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이센스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이센스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

본 문서에 포함된 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 성능은 개발 단계의 시스템에서 측정되었을 수 있으므로 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한 일부 성능은 추정을

통해 추측되었을 수도 있으므로 실제 결과는 다를 수 있습니다. 이 책의 사용자는 해당 데이터를 본인의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 제품들을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 기타 청구에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지 없이 변경될 수 있습니다.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

저작권 라이센스:

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 된 운영 플랫폼의 애플리케이션 프로그래밍 인터페이스(API)에 부합하는 애플리케이션을 개발, 사용, 판매 또는 배포할 목적으로 IBM에 추가 비용을 지불하지 않고 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이들 샘플 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 진술하지 않습니다.

이 정보를 소프트카피로 확인하는 경우에는 사진과 컬러 삽화가 제대로 나타나지 않을 수도 있습니다.

프로그래밍 인터페이스 정보

프로그래밍 인터페이스 정보는 본 프로그램과 함께 사용하기 위한 응용프로그램 소프트웨어 작성을 돕기 위해 제공됩니다.

이 책에는 고객이 IBM MQ의 서비스를 얻기 위해 프로그램을 작성할 수 있도록 하는 의도된 프로그래밍 인터페이스에 대한 정보가 들어 있습니다.

그러나 본 정보에는 진단, 수정 및 성능 조정 정보도 포함되어 있습니다. 진단, 수정 및 성능 조정 정보는 응용프로그램 소프트웨어의 디버그를 돕기 위해 제공된 것입니다.

중요사항: 이 진단, 수정 및 튜닝 정보는 변경될 수 있으므로 프로그래밍 인터페이스로 사용하지 마십시오.

상표

IBM, IBM 로고, ibm.com®는 전세계 여러 국가에 등록된 IBM Corporation의 상표입니다. 현재 IBM 상표 목록은 웹 "저작권 및 상표 정보"(www.ibm.com/legal/copytrade.shtml) 에 있습니다. 기타 제품 및 서비스 이름은 IBM 또는 타사의 상표입니다.

Microsoft 및 Windows는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

UNIX는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.

Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 등록상표입니다.

이 제품에는 Eclipse 프로젝트 (https://www.eclipse.org/) 에서 개발한 소프트웨어가 포함되어 있습니다.

Java 및 모든 Java 기반 상표와 로고는 Oracle 및/또는 그 계열사의 상표 또는 등록상표입니다.

부품 번호:

(1P) P/N: