

9.3

IBM MQ 계획

IBM

참고

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, [189 페이지의 『주의사항』](#)에 있는 정보를 확인하십시오.

이 개정판은 새 개정판에 별도로 명시하지 않는 한, IBM® MQ의 버전 9릴리스 3 및 모든 후속 릴리스와 수정에 적용됩니다.

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

© Copyright International Business Machines Corporation 2007년, 2024.

목차

계획	5
IBM MQ 릴리스 유형: 계획 고려사항.....	5
GDPR 대비를 위한 IBM MQ 및 IBM MQ Appliance 온프레미스 고려사항.....	8
단일 큐 관리자에 기반한 아키텍처.....	17
다중 큐 관리자에 기반한 아키텍처.....	17
분산 큐와 클러스터 계획.....	18
분산 발행/구독 네트워크 계획.....	64
멀티플랫폼에서 스토리지 및 성능 요구사항 계획.....	98
멀티플랫폼에서 디스크 공간 요구사항.....	99
멀티플랫폼에서 파일 시스템 지원 계획.....	102
멀티플랫폼의 MFT 에 대한 파일 시스템 지원 계획.....	128
멀티플랫폼에서 순환 또는 선형 로깅 선택.....	128
AIX의 공유 메모리.....	129
IBM MQ 및 UNIX System V IPC 자원.....	129
IBM MQ 및 UNIX 프로세스 우선순위.....	129
z/OS에서 IBM MQ 환경 계획.....	130
큐 관리자에 대한 계획.....	130
채널 시작기 계획.....	155
큐 공유 그룹(QSG) 계획.....	159
백업 및 복구 계획.....	171
z/OS UNIX 환경 계획.....	179
Advanced Message Security 계획.....	179
Managed File Transfer 계획.....	180
z/OS 에서 IBM MQ Console 및 REST API 사용 계획.....	185
주의사항	189
프로그래밍 인터페이스 정보.....	190
상표.....	190

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.

- 17 페이지의 『[단일 큐 관리자에 기반한 아키텍처](#)』
- 17 페이지의 『[다중 큐 관리자에 기반한 아키텍처](#)』
 - 18 페이지의 『[분산 큐와 클러스터 계획](#)』
 - 64 페이지의 『[분산 발행/구독 네트워크 계획](#)』

 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 그룹](#)을 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 버전화](#)를 참조하십시오.

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

관련 개념

IBM MQ 릴리스 유형 및 버전화

130 페이지의 『[z/OS에서 IBM MQ 환경 계획](#)』

IBM MQ 환경을 계획할 때에는 데이터 세트, 페이지 세트, Db2, 커플링 기능, 로깅 및 백업 기능 필요에 대한 자원 요구사항을 고려해야 합니다. 이 주제를 사용하여 IBM MQ가 실행하는 환경을 계획하십시오.

가용성, 복구 및 재시작

관련 태스크

[요구사항 검사](#)

[메시지가 유실되지 않았는지 확인\(로깅\)](#)

IBM MQ 릴리스 유형: 계획 고려사항

IBM MQ의 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 버전화](#)를 참조하십시오.

IBM MQ for Multiplatforms에 대한 고려사항



주문

Passport Advantage® 내에는 IBM MQ 9.3에 대한 두 개의 별도 eAssemblies 가 있습니다. 하나는 IBM MQ 9.3.0 Long Term Support 릴리스에 대한 설치 이미지를 포함하고 다른 하나는 IBM MQ 9.3.x Continuous Delivery 릴리스에 대한 설치 이미지를 포함합니다. 선택한 릴리스에 따라 eAssembly에서 설치 이미지를 다운로드합니다.

모든 IBM MQ 버전(IBM MQ 9.3의 경우 LTS 릴리스와 CD 릴리스 둘 다)이 동일한 제품 ID에 속합니다.

IBM MQ 를 사용할 수 있는 인타이틀먼트는 라이선스가 부여된 구성요소 및 가격 책정 메트릭의 제한조건에 따라 전체 제품 (PID) 에서 확장됩니다. 이는 IBM MQ 9.3에 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 라이선스 정보](#)를 참조하십시오.

동일한 운영 체제 이미지에 IBM MQ 9.3.0 LTS 릴리스 및 IBM MQ 9.3.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 수정사항 및 보안 패치를 제공하는 누적 보안 업데이트 (CSU)를 제공하는 수정팩을 적용하여 서비스됩니다. 수정팩 및 CSU 는 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



주문

IBM MQ for z/OS 9.3을 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 라이선스가 있는 제품 ID이므로, 한 기능에 라이선스가 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 세 자리 VRM 레벨입니다. IBM MQ 프로그램은 MQIA_COMMAND_LEVEL 선택자를 전달하는 MQINQ를 호출하여 연결된 큐 관리자의 명령 레벨을 가져올 수 있습니다.

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.

LTS 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) 와 직접 정렬됩니다. 이러한 수정사항은 누적되며 동등한 멀티플랫폼 수정팩 또는 CSU와 동시에 사용 가능하게 됩니다.

CD 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
9.0.0 LTS	9.3.0 LTS 또는 9.3.0 CD	9.0.0 LTS가 표준 지원을 벗어나므로 역방향 마이그레이션이 지원되지 않습니다.
9.1.0 LTS	9.3.0 LTS 또는 9.3.0 CD	역방향 마이그레이션이 지원됩니다.
9.2.0 LTS	9.3.0 LTS 또는 9.3.0 CD	역방향 마이그레이션이 지원됩니다.
9.2.5 CD	9.3.0 LTS 또는 9.3.0 CD	소스 릴리스가 수정자가 0이 아닌 CD이므로 역방향 마이그레이션이 지원되지 않습니다.
9.3.0 LTS 또는 9.3.0 CD	9.3.1 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/OS에서 유지보수 적용 및 제거](#)

관련 정보

[다운로드 중IBM MQ 9.3](#)

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 **GDPR**을 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 콘텐츠에 설명된 제품, 서비스 및 기타 기능은 일부 고객의 상황에 적합하지 않거나 사용 가능성이 제한될 수 있습니다. **IBM**은 법률, 회계 또는 감사 관련 자문을 제공하지 않으며, **IBM**의 제품이나 서비스가 클라이언트의 관련 법령 및 규정을 준수한다는 진술이나 보증을 제공하지 않습니다.

목차

1. [GDPR](#)
2. [GDPR을 위한 제품 구성](#)
3. [데이터 수명 주기](#)
4. [데이터 수집](#)
5. [데이터 스토리지](#)
6. [데이터 액세스](#)
7. [데이터 처리](#)
8. [데이터 삭제](#)
9. [데이터 모니터링](#)
10. [개인정보 사용 제한 기능](#)
11. [파일 핸들링](#)

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 Developer](#)의 IBM MQ 영역에 있는 페이지의 공개 댓글 영역
- [IBM Documentation](#)의 IBM MQ 제품 정보 페이지에 있는 공개 댓글 영역
- [IBM 지원 센터 포럼](#)의 공개 댓글
- [IBM Integration Ideas](#)의 공개 댓글

일반적으로 사용자 문의사항에 대한 개별적인 답변에는 고객 이름 및 이메일 주소만 사용되며, 이러한 개인정보 사용은 [IBM 온라인 개인정보처리방침](#)을 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 강조 표시합니다.

- 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.

- [로그: 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 콘텐츠를 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 콘텐츠를 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 콘텐츠에 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 3rd 당사자 서비스(예: LDAP, Salesforce 등)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 콘텐츠를 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 Bridge to Salesforce[원격에만 해당]
- IBM MQ Bridge to Blockchain[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)
- **Deprecated** **Removed** IBM Cloud Pak® for Integration Operations Dashboard 통합-상위 레벨 추적 데이터를 중앙 소스에 제출합니다 (CP4I 만 해당). 이 기능은 IBM MQ Operator 2.3.0에서 더 이상 사용되지 않으며 IBM MQ Operator 2.4.0에서 제거되었습니다.
- 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 9.3.3** 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 매핑
- 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 셸을 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 및 IBM MQ 9.3.4 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 매핑:

역할 매핑 단계에서 인증 단계에 제공된 신임 정보는 대체 사용자 ID에 매핑될 수 있습니다. 매핑된 사용자 ID가 진행하도록 허용된 경우(예: 채널 인증 규칙으로 관리 사용자가 차단될 수 있음), 매핑된 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 콘텐츠를 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 라이선스가 부여된 IBM MQ의 사용자는 보안 정책의 정의 및 구성을 통해 애플리케이션이 메시지에 보유한 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 메시지가 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 보유한 발행 데이터를 삭제합니다.
 - 메시지가 비지속적임을 지정하고 큐 관리자를 재시작합니다.
 - 메시지 만기를 사용하거나 보유한 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.
- Salesforce 연결 데이터 구성 파일.
- 블록체인 연결 데이터 구성 파일.
- IBM MQ Internet Pass-Thru 구성 파일 및 키 저장소.

자세한 내용은 다음을 참조하십시오.

- IBM MQ Bridge to Salesforce 구성
- 블록체인에 사용할 IBM MQ 구성
- 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 개요](#)를 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 예](#)를 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 개요](#)를 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 MQ 분산 큐잉 기술](#)을 참조하십시오.

분산 큐 관리를 통해 큐 관리자 간의 통신을 작성하고 모니터링할 수 있습니다. 분산 큐 관리에 대한 자세한 정보는 [분산 큐 관리 소개](#)를 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 클러스터](#)를 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 큼니까?
- 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 1](#)에 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-Thru](#)의 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 אלי어스 이름과 동일한 이름을 제공할 수 있습니다. 이는 사용하는 라우트와 전송 큐를 연관시켜서 중간(멀티홉핑된) 큐 관리자를 통해 작성된 병렬 라우트의 명확한 개요를 제공합니다.

이는 채널 이름에는 명확하지 않습니다. 예를 들어, QM2에 대한 [20 페이지의 그림 1](#)의 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 1](#)의 제안된 채널 이름 표는 [20 페이지의 표 1](#)에 제공됩니다.

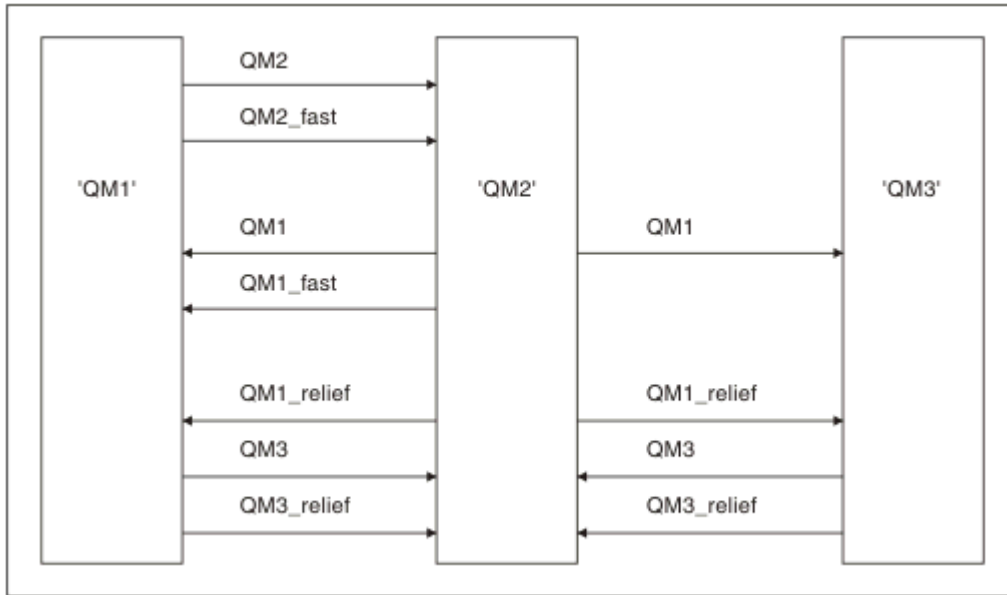



그림 1. 모든 채널이 표시된 네트워크 다이어그램

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.  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.

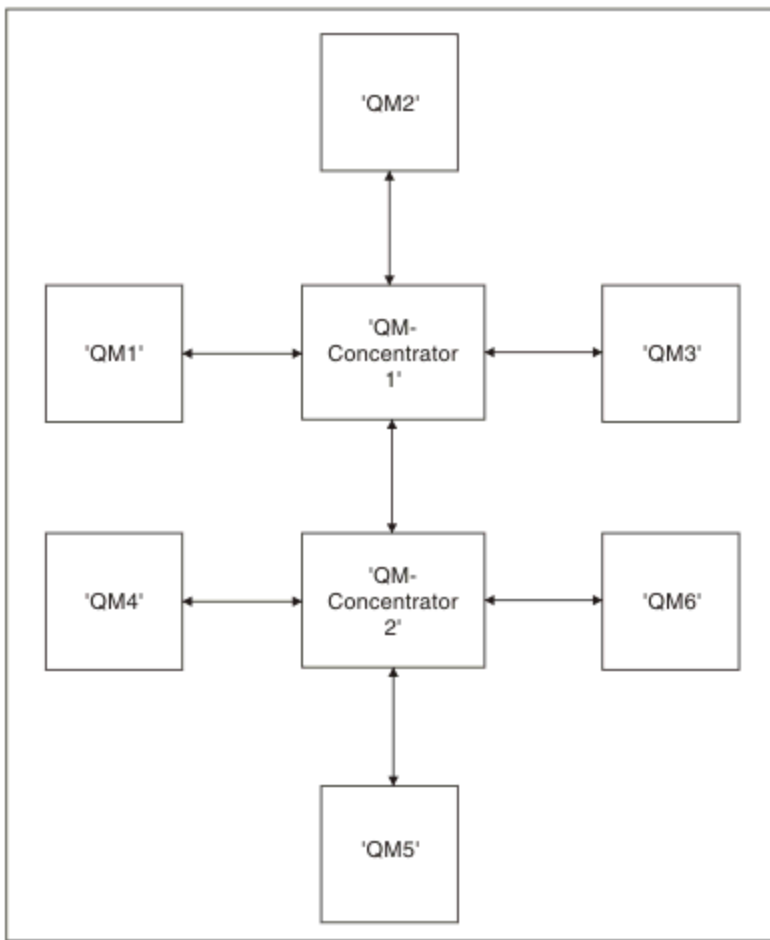


그림 2. QM 집선기가 표시된 네트워크 다이어그램

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.

- 35 페이지의 『클러스터 예』
- 31 페이지의 『클러스터 조직화』
- 31 페이지의 『클러스터 이름 지정 규칙』
-  32 페이지의 『큐 공유 그룹 및 클러스터』
- 33 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 34 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/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/OS에서 전송 큐를 수동으로 정의하는 경우 자세한 정보는 25 페이지의 『수동으로 정의된 클러스터 전송 큐 계획』의 내용을 참조하십시오.

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.

runswch1 명령  또는 z/OS의 [CSQUTIL의 SWITCH CHANNEL \(*\) STATUS](#) 명령 를 사용하여 클러스터-송신자 채널의 상태 및 전송 큐 구성에 미해결된 보류 중인 변경사항을 조회할 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 구현](#)을 참조하십시오.


변경 실행 취소



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/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: 

큐 관리자가 IBM MQ 8.0로 마이그레이션되면 큐 관리자에 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE입니다.

이 큐를 먼저 정의하여 명령 ALTER QMGR DEFCLXQ(CHANNEL)이 적용되도록 하십시오.

다음 JCL은 모델 큐 정의에 사용할 수 있는 코드 예입니다.

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

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

* MODEL QUEUE ATTRIBUTES
DEFTYPE( PERMDYN ) +

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

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

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

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) 

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』

IBM MQ for z/OS에서 전송 큐를 직접 정의하는 경우 정의 및 메시지가 보유되는 페이지 세트를 더 많이 제어할 수 있습니다.

관련 참조

[ALTER QMGR](#)

[DISPLAY CLUSQMGR](#)



수동으로 정의된 클러스터 전송 큐 계획

IBM MQ for z/OS에서 전송 큐를 직접 정의하는 경우 정의 및 메시지가 보유되는 페이지 세트를 더 많이 제어할 수 있습니다.

시작하기 전에

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

이 태스크 정보

관리자가 수동으로 전송 큐를 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)를 계획해야 합니다. 서로 다른 큐에 대해 서로 다른 페이지 세트를 가질 수 있으며, 이들 사이에 격리를 제공할 수 있으므로, 하나의 페이지 세트 채우기가 다른 페이지 세트의 전송 큐에 영향을 주지 않습니다.

각 채널이 적절한 큐를 선택하는 방법에 대한 정보는 [클러스터 전송 큐 및 클러스터 송신자 채널에 대한 작업](#)을 참조하십시오.

채널이 시작되면 이는 해당 연관을 새 전송 큐로 스위치합니다. 메시지가 유실되지 않도록 하기 위해 큐 관리자는 자동으로 이전 클러스터 전송 큐에서 새 전송 큐로 순서대로 메시지를 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 프로세스는 CSQUTIL SWITCH 함수를 사용합니다. 자세한 정보는 [클러스터-송신자 채널 \(SWITCH\) 과 연관된 전송 큐 전환](#)을 참조하십시오.

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 모니터링하고 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

z/OS에서 클러스터 전송 큐를 설정했습니다.

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.QUEUE(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.  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

43 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

28 페이지의 그림 3에서는 분산 큐잉에 필요한 컴포넌트를 보여줍니다.

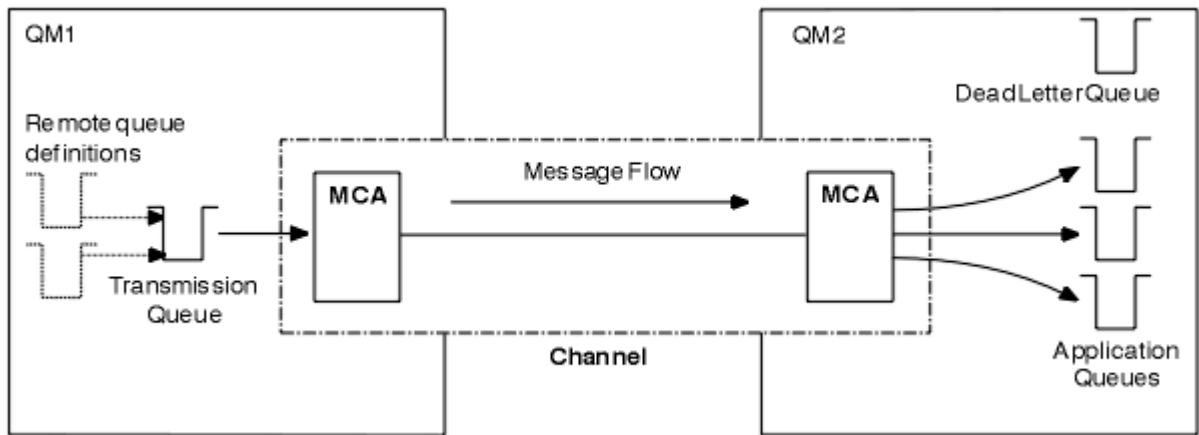


그림 3. 분산 큐잉

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.

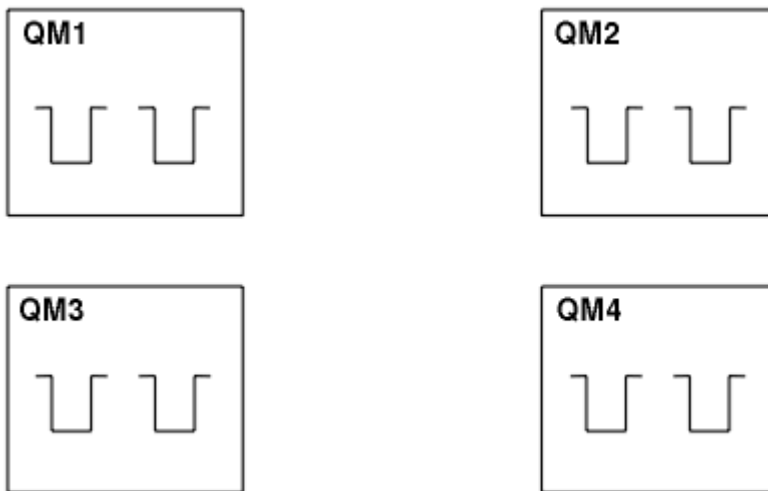


그림 4. 네 개의 큐 관리자 네트워크

분산 큐잉을 사용하여 네트워크를 설정하기 위한 정의

분산 큐잉을 사용하여 28 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 관리자로부터 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 [28 페이지의 그림 3](#)에 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.

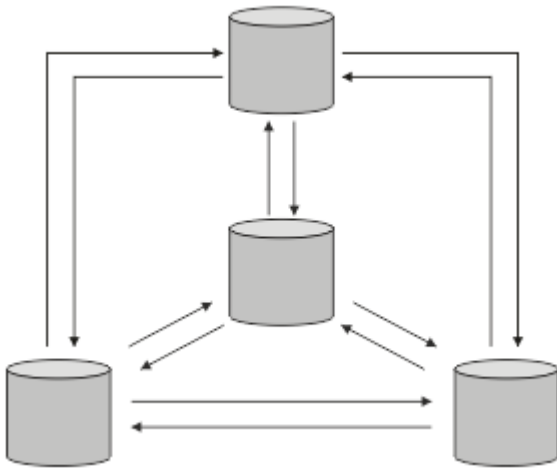


그림 6. 연결된 셋 이상의 전체 저장소

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 선택했다면 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 사용하려고 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. 32 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 기억하는데 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.QM2입니다.

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 S1는 첫 번째 SNA 채널일 수 있고, N3는 네트워크 우선순위가 3인 NetBIOS 채널일 수 있으며, T4는 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 관리에 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/OS의 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 CONNAME로 CLUSRCVR를 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 큐 공유 그룹 및 클러스터

공유 큐는 클러스터 큐일 수 있으며 큐 공유 그룹의 큐 관리자도 클러스터 큐 관리자일 수 있습니다.

IBM MQ for z/OS에서는 큐 관리자를 큐 공유 그룹으로 그룹화할 수 있습니다. 큐 공유 그룹의 큐 관리자는 최대 32개의 큐 관리자에서 공유할 로컬 큐를 정의할 수 있습니다.

공유 큐는 클러스터 큐일 수도 있습니다. 큐 공유 그룹의 큐 관리자도 하나 이상의 클러스터에 있을 수 있습니다.

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 CONNAME로 CLUSRCVR를 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. 채널

이 연결을 다시 시도하기 시작하면 동일한 일반 이름을 가진 다른 큐 관리자에 다시 연결하여 메시지 플로우를 방해할 수 있습니다.

그룹 리스너 포트를 사용하는 CLUSRCVR 채널을 시작할 수 없습니다. 이 경우 CLUSRCVR가 매번 연결하는 큐 관리자를 알 수 없기 때문입니다. 클러스터에 대한 정보가 보관되는 클러스터 시스템 큐는 공유되지 않습니다. 각 큐 관리자에 고유한 클러스터 시스템 큐가 있습니다.

클러스터 채널은 애플리케이션 메시지뿐만 아니라 클러스터 설정에 대한 내부 시스템 메시지를 전송하는 데 사용됩니다. 클러스터의 각 큐 관리자가 클러스터링에 제대로 참여하려면 이러한 내부 시스템 메시지를 수신해야 하므로 이 메시지를 수신할 고유한 CLUSRCVR 채널이 필요합니다.

공유 CLUSRCVR은 큐 공유 그룹(QSG)의 모든 큐 관리자에서 시작될 수 있으므로 내부 시스템 메시지가 QSG 큐 관리자에 일관되지 않게 공급될 수 있습니다. 즉, 어떤 큐 관리자도 클러스터에 제대로 참여할 수 없습니다. 공유 CLUSRCVR 채널을 사용할 수 없도록 하기 위해 모든 시도가 [CSQX502E](#) 메시지와 함께 실패합니다.

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

34 페이지의 그림 7에서 STF2 큐 관리자는 두 클러스터 모두의 멤버입니다. 한 큐 관리자가 둘 이상의 클러스터의 멤버이면 이름 목록을 사용하여 필요한 정의의 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 34 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.

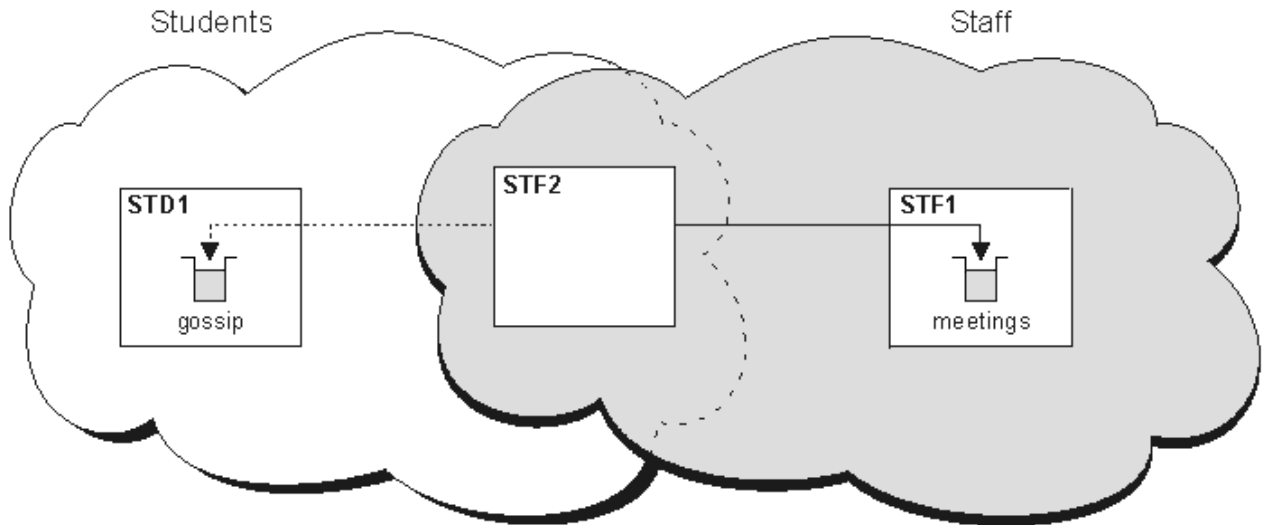


그림 7. 서비스 클래스

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 위해 서입니다. 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 임의로 서버스되지 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

62 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 안 됩니다. 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 자용도가 아직은 높습니다.

DEMOCLSTR

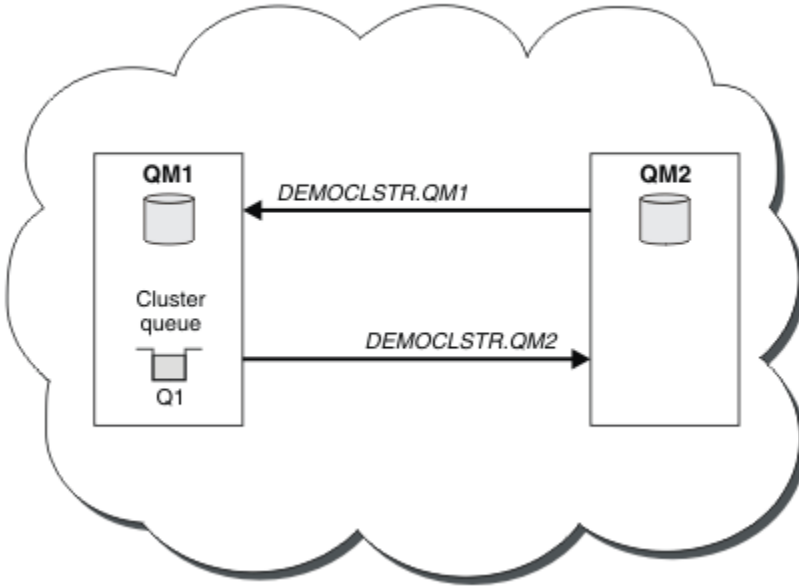


그림 8. 두 큐 관리자의 작은 클러스터

- 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 설정](#)을 참조하십시오.

37 페이지의 [그림 9](#)에서는 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.

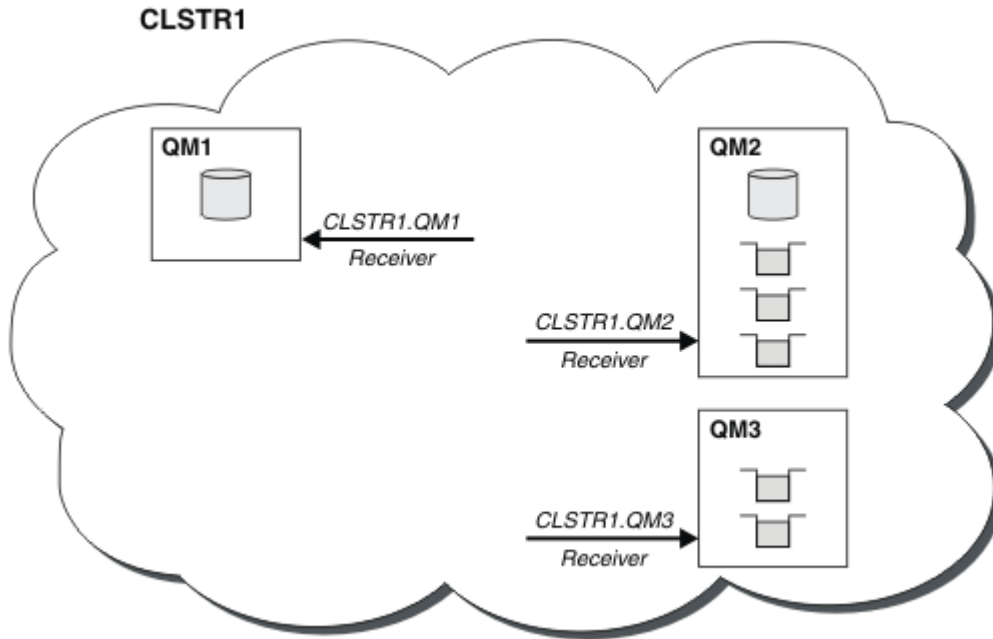


그림 9. 큐 관리자의 클러스터

- 37 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.

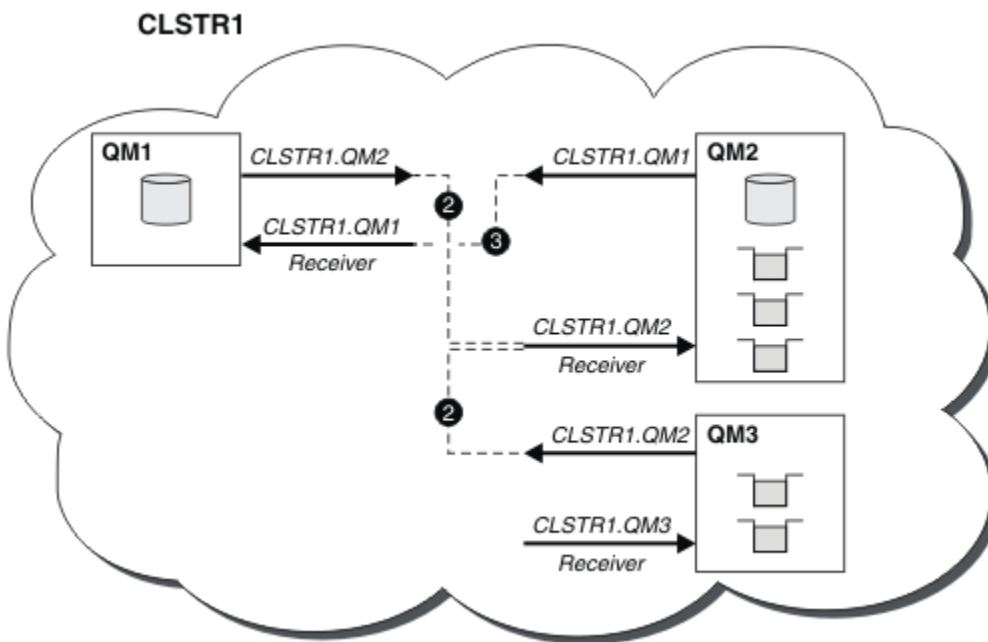


그림 10. 송신자 채널이 있는 큐 관리자의 클러스터

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.

37 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.

38 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.

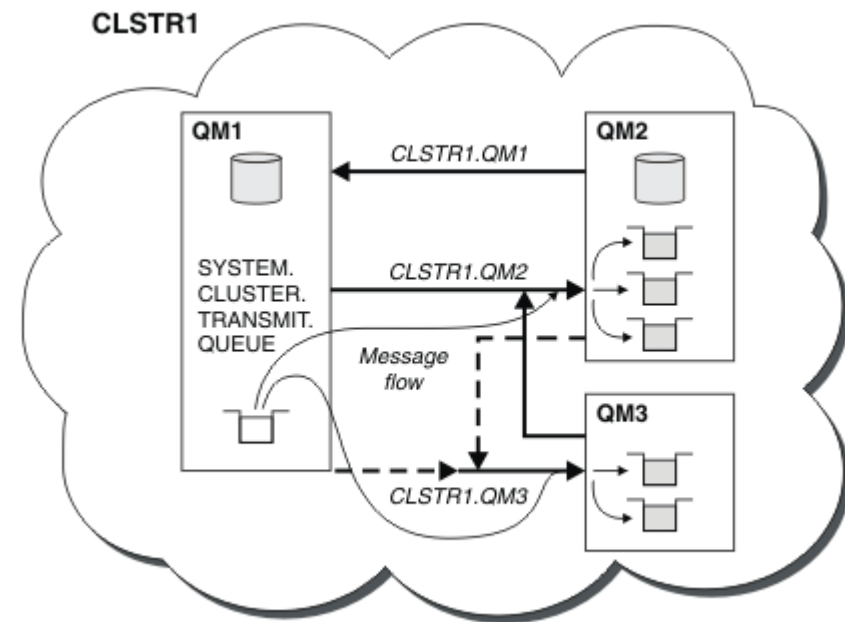


그림 11. 자동 정의된 채널이 표시된 큐 관리자의 클러스터

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 [33 페이지의 『중첩 클러스터』](#) 및 [클러스터 간 메시지 경로 구성](#)을 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. [31 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 방법](#)을 참조하십시오.
- 일부 큐 관리자를 다른 팀에서 관리하는 경우. 자세한 정보는 이전 [39 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 라우팅 하기 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 **QMNAME** 특성이 있는 큐 관리자 알리어스는 원하는 경우 워크로드 밸런싱을 다시 구동합니다.

관련 개념

31 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 배열로 단순화할 수 있는 설계 의사결정에 대한 정보를 포함합니다.

- [40 페이지의 『클러스터링 인프라의 성능』](#)
- [41 페이지의 『전체 저장소』](#)
- [42 페이지의 『애플리케이션에 전체 저장소의 큐를 사용해야 하나?』](#)
- [42 페이지의 『채널 정의 관리』](#)
- [42 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 컬렉션입니다. 클러스터에서 모든 큐 관리자에 대한 전체 정보 세트를 호스팅하는 큐 관리자에 전체 저장소가 있습니다. 전체 저장소 및 부분 저장소에 대한 자세한 정보는 [클러스터 저장소](#)를 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 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 하나?

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 하나?

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 31 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 안 됩니다. 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.

44 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.

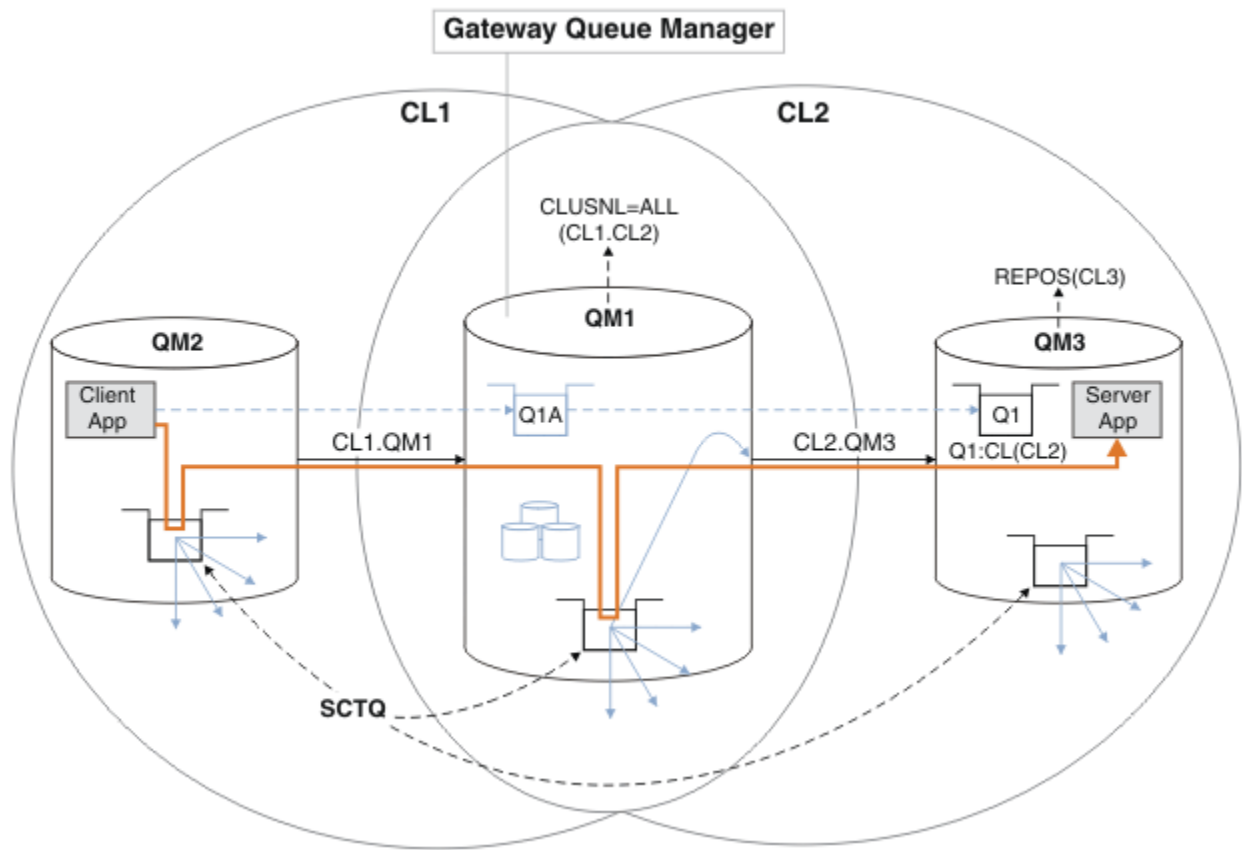
성능면에서 보면 단일 큐는 문제가 되지 않습니다. 공용 전송 큐는 일반적으로 성능 병목을 나타내지 않습니다. 게이트웨이의 메시지 처리량이 대부분 연결된 채널의 성능에 따라 판별됩니다. 처리량은 대개 큐의 수나 채널을 사용하는 큐의 메시지 수에 영향을 받지 않습니다.

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 모니터링하기 쉽지 않습니다. 모든 메시지가 단일 전송 큐에 있습니다. 전송 큐 용량을 표시해도 메시지가 모든 목적지로 전송되고 있는지 거의 표시되지 않습니다.



참고: 44 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

44 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. 44 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 접근법](#)입니다. 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](#)

33 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

48 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(46 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.

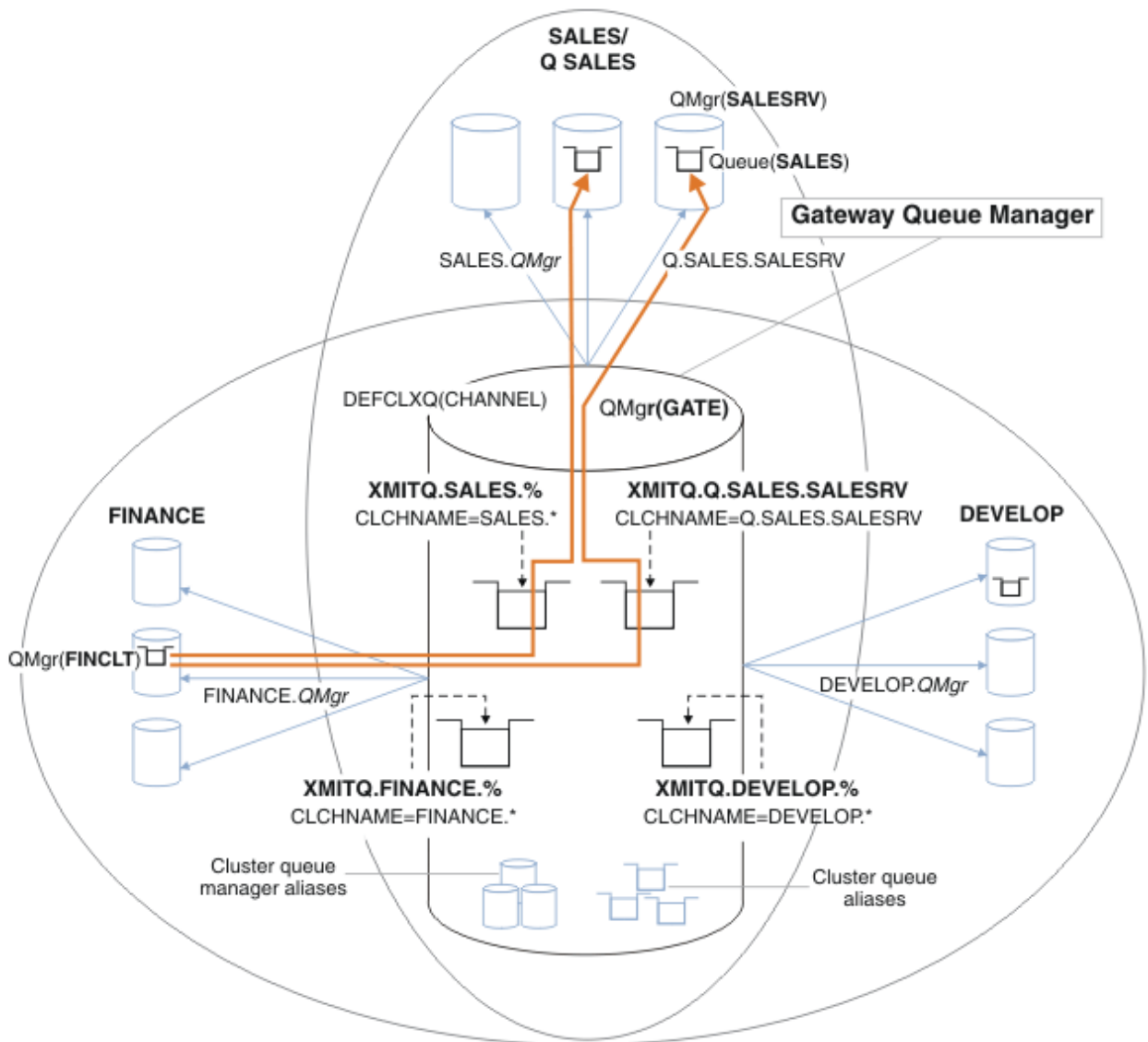


그림 13. 다른 부서 IBM MQ 클러스터의 특정 전송 큐 예

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.

- 49 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.

- 각 대상 큐가 해당 큐 관리자의 특정 클러스터에 있는 유일한 큐가 되도록 플로우 목적지를 구성하십시오.
 - 체계적인 이름 지정 규칙을 따라 작성한 새 클러스터에 대해 클러스터 송신자 및 클러스터 수신자 채널을 작성하십시오.
 - 39 페이지의 『클러스터링: 중첩 클러스터에 대한 특수 고려사항』의 내용을 참조하십시오.
 - 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. 49 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.

33 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.QUEUE`와 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.QUEUE`의 큐 관리자가 작성하고 이름은 `SYSTEM.CLUSTER.TRANSMIT.ChannelName`입니다. 큐 관리자 속성 `DEFCLXQ`이(가) `CHANNEL`로 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. 39 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.QUEUE`에 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.QUEUE`에 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.QUEUE`에 저장된 메시지 수가 증가할 수 있습니다. 한 대상에 대한 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`의 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.QUEUE`을(를) 공유합니다. 게이트웨이 큐 관리자의 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`에 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 메시지가 있고 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 플로우와 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**(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. 목표한 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 모니터링할 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 모니터링한 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 45 페이지의 『클러스터링: 클러스터 전송 큐를 구성하는 방법을 계획』의 프로시저를 적용할 방법 및 50 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 52 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.

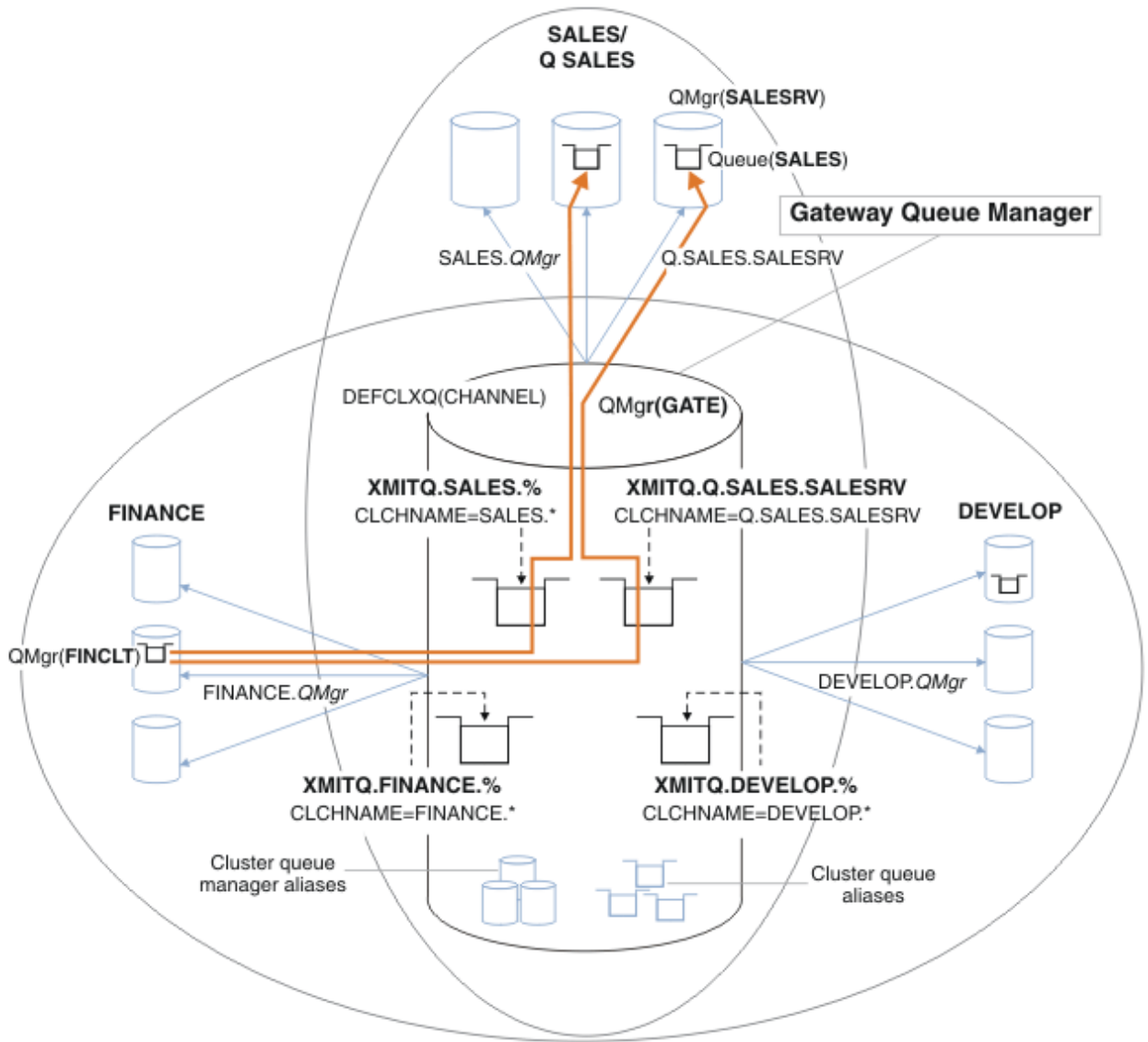


그림 14. 다른 부서 IBM MQ 클러스터의 특정 전송 큐 예

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 SALESRV(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. SALES(를) 저장소 큐 관리자의 클러스터 이름 목록에 추가하십시오. 이름 목록은 큐 관리자 REPOSNL 매개변수에서 참조됩니다.
- ii) 게이트웨이 큐 관리자의 클러스터 이름 목록에 Q. SALES(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. 55 페이지의 그림 15(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*(를) 각 저장소 큐 관리자에 추가
- ii) 클러스터 전송자 채널 Q. SALES. *OtherRepositoryQMGr*(를) 각 저장소 큐 관리자에 추가하여 다른 저장소 관리자에 연결하십시오. 이 채널을 시작하십시오.
- iii) 클러스터 수신자 채널 Q. SALES. SALESRV 및 Q. SALES. GATE(를) 실행 중인 저장소 큐 관리자 중 하나에 추가하십시오.
- iv) 클러스터 송신자 채널 Q. SALES. SALESRV 및 Q. SALES. GATE(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. SALESRV(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. [55 페이지의 그림 15](#)의 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, SALES 및 Q.SALES 클러스터를 작성하기 위한 전체 **MQSC** 명령은 기본 클러스터에 대한 정의, 새 클러스터에서 판매 큐를 분리하고 게이트웨이 클러스터 전송 큐를 분리하기 위한 변경사항 및 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.

[52 페이지의 표 4](#)의 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.

52 페이지의 표 4의 각 큐 관리자 이름에 대해 `stimqm 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)
```

chgate.txt

```
ALTER NAMELIST(ALL) NAMES(SALES, FINANCE, Q.SALES)
```

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

4. SALES 클러스터에서 SALES 큐를 제거하십시오.

55 페이지의 그림 15의 **MQSC** 명령을 실행하십시오.

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

그림 15. 판매 클러스터에서 큐 관리자 SALESRV의 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(54 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(58 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(59 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 복시 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, 49 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 모니터링하십시오. 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 중창에 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 58 페이지의 『활성 클러스터 송신자 채널을 다른 클러스터 전송 큐 세트로 전환』의 58 페이지의 『2』 단계와 대조시키십시오. 이 단계에서는 클러스터 송신자 채널이 실행 중이므로 새 전송 큐에 메시지가 반드시 빌드되지는 않습니다.

4. 옵션: 전환하는 채널을 모니터링하십시오.

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 ChannelName에 대한 전송 큐가 큐 QueueName에서 QueueName으로 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.

- 60 페이지의 『클러스터의 오브젝트 이동』 (IBM MQ의 새 버전이나 수정팩을 설치하지 않은 채, 클러스터 내에서 오브젝트를 이동시키는 우수 사례).
- 62 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 **CLWL RANK** 특성을 높은 값(예: 5)으로 설정하십시오.
2. 큐의 새 인스턴스를 작성하고 해당 **CLWL RANK** 특성을 0으로 설정하십시오.
3. 큐의 새 인스턴스에 대해 이용 애플리케이션을 배치하고 시작하는 것과 같은 새 시스템의 추가 구성을 완료하십시오.
4. 새 큐 인스턴스의 **CLWL RANK** 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 CLUSDR 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 올바르게 작동하는 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 올바르게 작동하는 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 것이라 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 9.3 큐 관리자에 대한 정보를 수신하는 IBM MQ 9.2 큐 관리자는 IBM MQ 9.2 정보만 저장합니다. IBM MQ 9.2 레코드를 수신하는 IBM MQ 9.3 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 마이그레이션](#)을 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 안 됩니다. 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 안 되며 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 62 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

94 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 보유하고 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 중단은 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.
 -  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. 62 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 네트워크](#)를 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. [69 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 관리자에서 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. [73 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. [94 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 결합](#)을 참조하십시오.

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

- 네트워크는 얼마나 큼니까?
- 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.

[82 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 계층을 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 계층을 사용하여 이를 수행할 수 있습니다.

모든 토픽로지에서, 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](#).

82 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 토픽로지의 다른 큐 관리자에게 통신됩니다. 직접 라우트 클러스터 및 계층에서는, 토픽에 발행자가 있는지 여부와 상관 없이 이 구독 정보가 토픽로지의 모든 큐 관리자에 전파됩니다. 토픽로지가 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.

82 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) 토픽로지의 큐 관리자가 얼마나 동적인지 고려하십시오.

계층은 토픽로지의 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. 92 페이지의 『[발행/구독 클러스터에서 보유된 발행에 대한 디자인 고려사항](#)』의 내용을 참조하십시오.

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

82 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 64 페이지의 『[분산 발행/구독 네트워크 계획](#)』 및 82 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 호스팅하는 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 성능](#)을 참조하십시오.

클러스터가 직접 라우트 클러스터되는 발행/구독의 오버헤드를 수용할 수 없음이 파악되면 대신에 [토픽 호스트 라우트 발행/구독](#)을 사용할 수 있습니다. 또는 극단적인 상황에서, 클러스터의 모든 큐 관리자에서 큐 관리자 속성 **PSCLUS**를 **DISABLED**로 설정하여 클러스터된 발행/구독 기능을 완전히 사용 불가능하게 할 수 있습니다. [91 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 호스팅하는 클러스터 큐 관리자는 구독의 토픽 문자열로 맵핑할 클러스터 토픽을 호스팅하는 큐 관리자에만 자동으로 클러스터 송신자 채널을 작성합니다.
- 클러스터된 토픽 하의 토픽 문자열에 대한 큐 관리자의 첫 번째 구독으로 인해 클러스터에서 클러스터된 토픽을 호스팅하는 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 "흡"이 발생합니다. 추가 "흡"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](#)

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 네트워크](#)를 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.

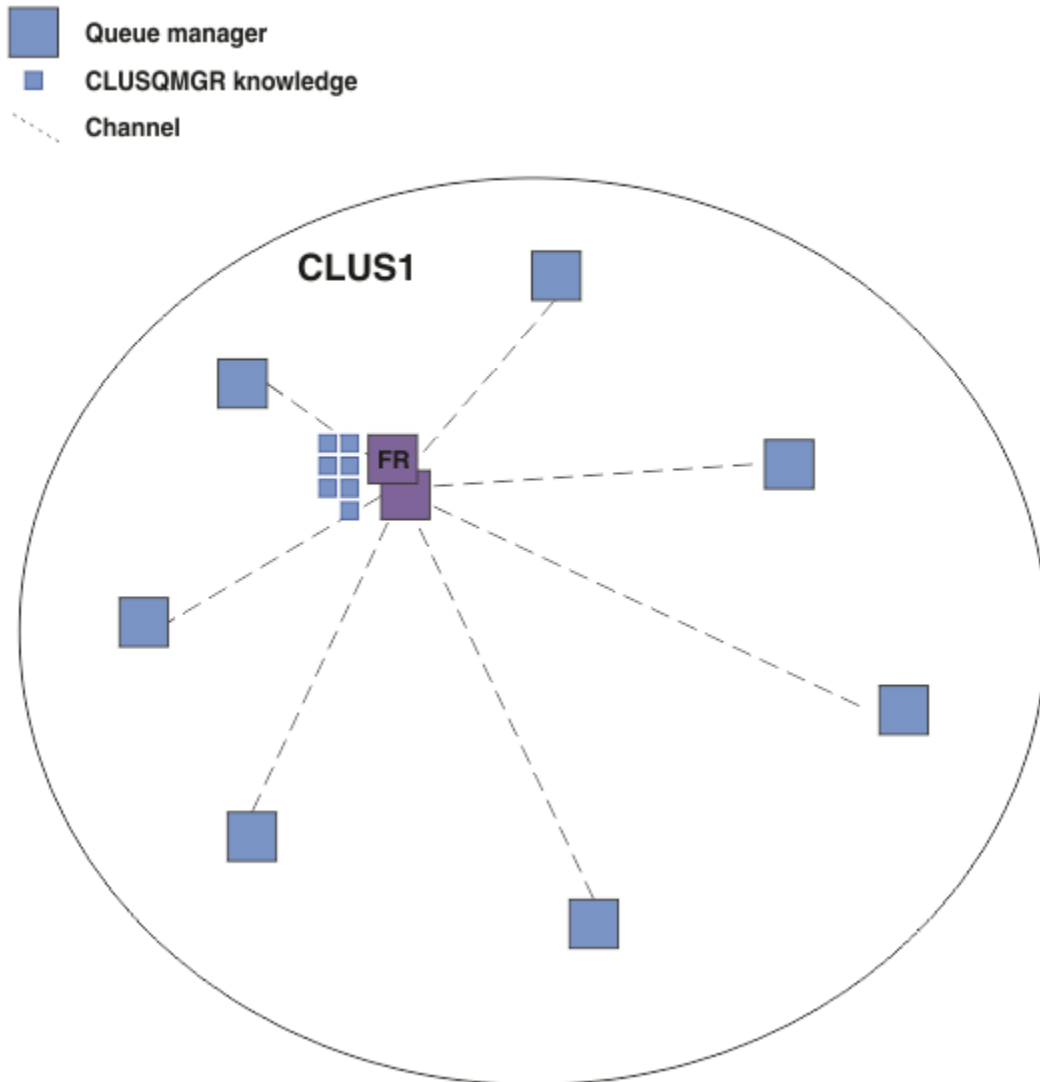


그림 16. 큐 관리자 클러스터

직접 라우트 클러스터의 큐 관리자 간에 플로우되는 발행물의 경우 [발행/구독 클러스터 구성](#)에 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.

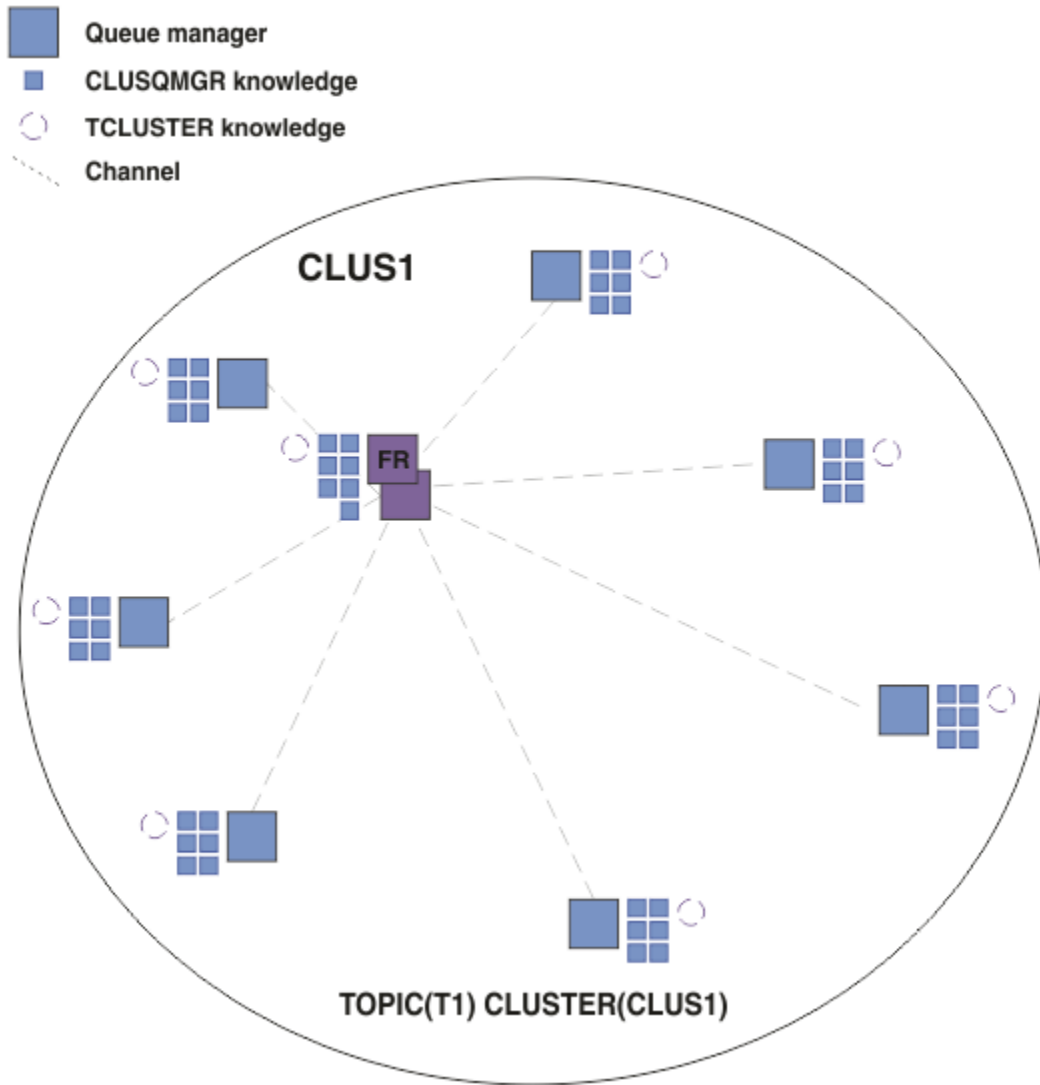


그림 17. 직접 라우트 발행/구독 클러스터

구독이 작성될 때 구독을 호스트하는 큐 관리자는 클러스터의 모든 큐 관리자에 대한 채널을 설정하고 구독 세부 사항을 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.

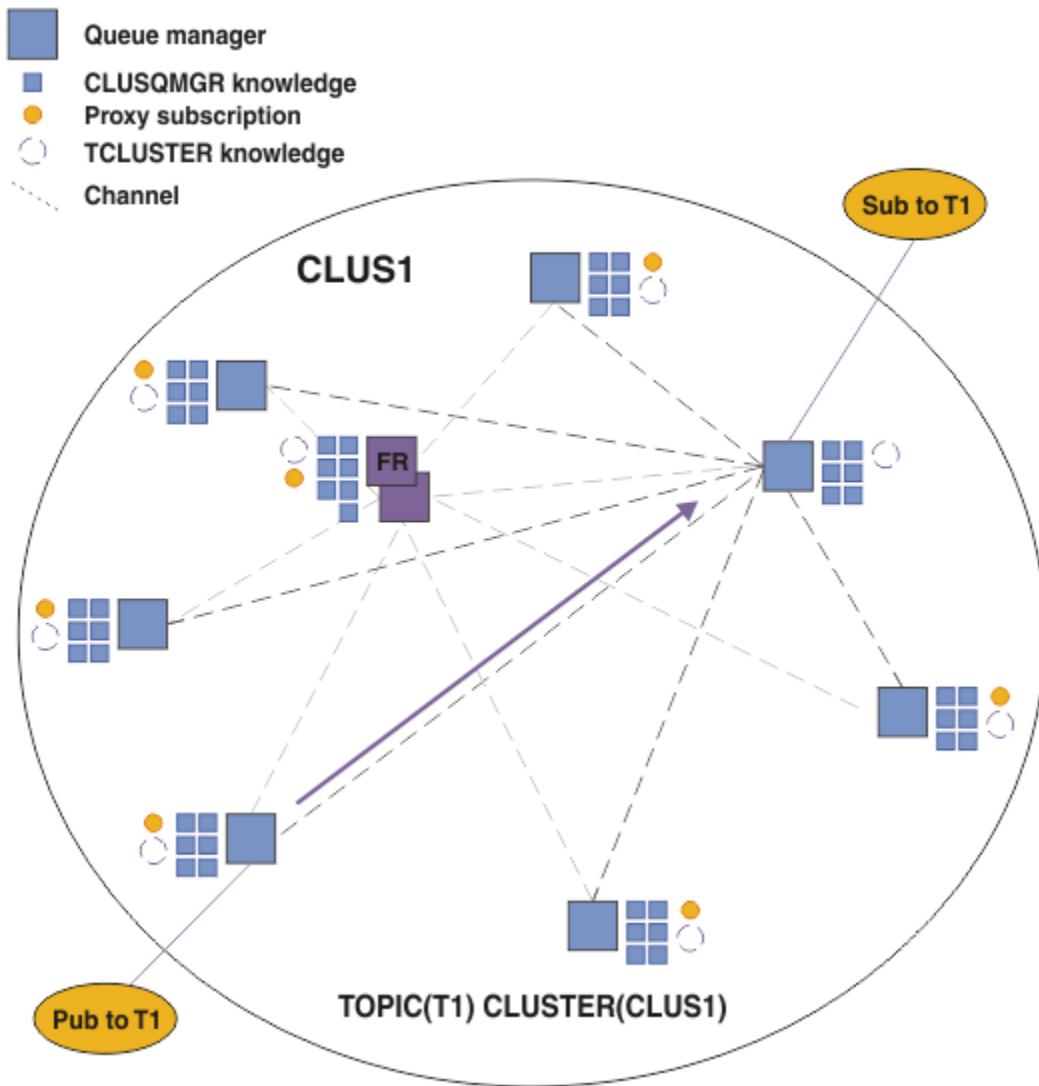


그림 18. 클러스터된 토픽에 대한 발행자 및 구독자가 있는 직접 라우트 발행/구독 클러스터

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.

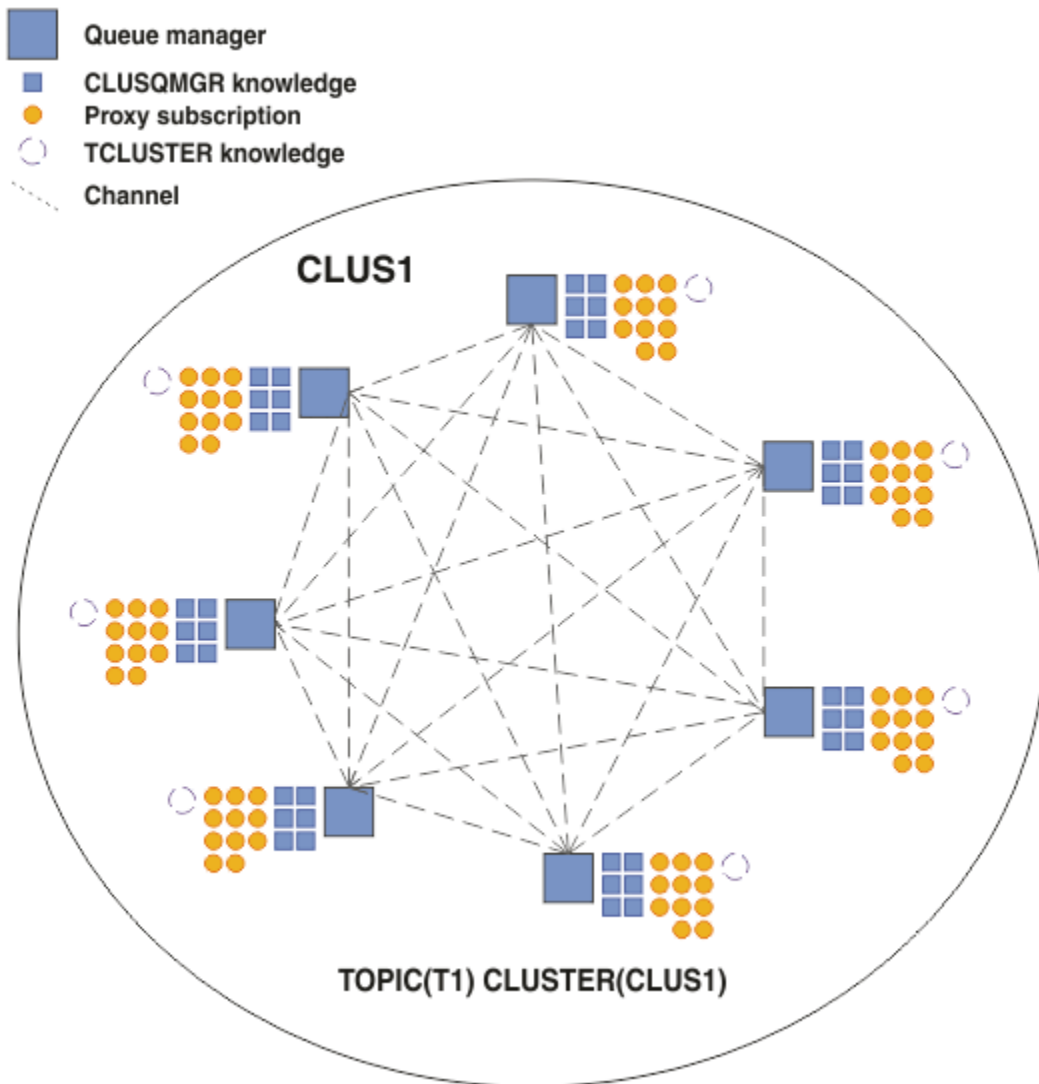


그림 19. 완전히 상호 연결된 직접 라우트 발행/구독 클러스터

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 73 페이지의 『[발행/구독 클러스터의 토픽 호스트 라우팅](#)』 및 94 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 네트워크](#)를 참조하십시오.

토픽 호스트 라우팅의 작동 및 혜택을 이해하려면 먼저 69 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 구성](#)에 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 관리자로서만 연결됨을 유의하십시오.

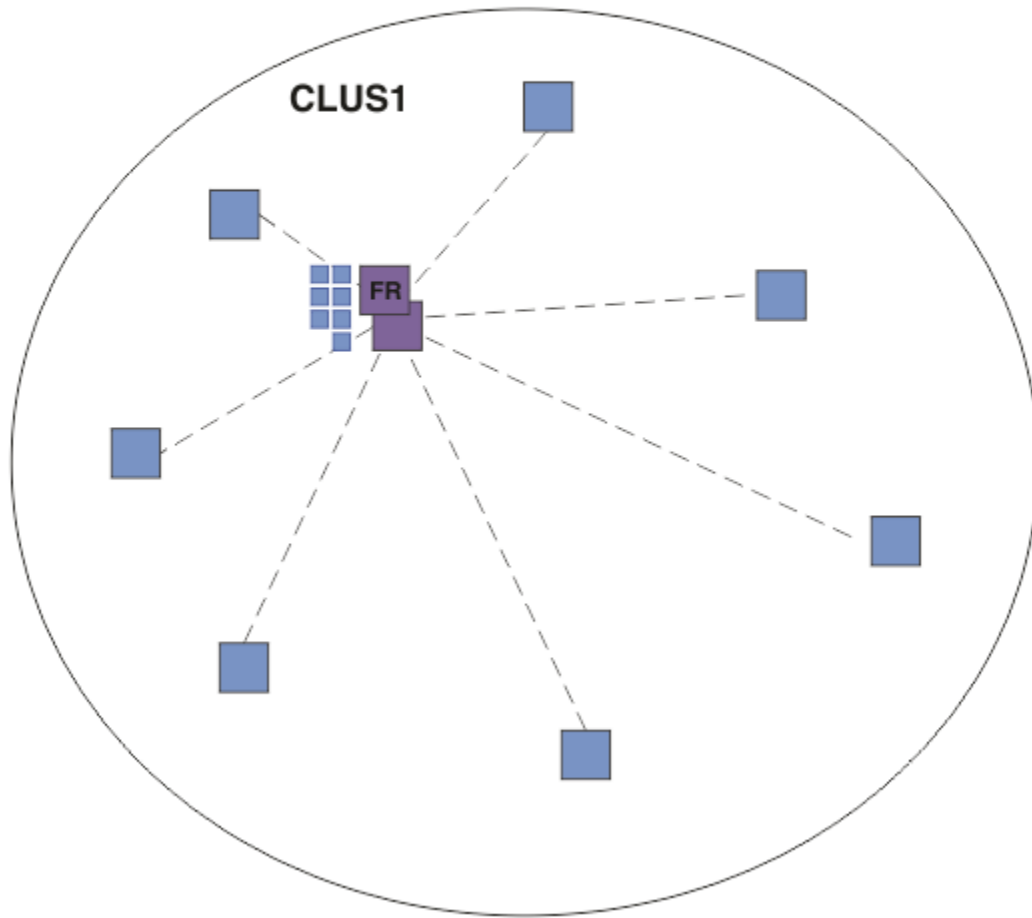
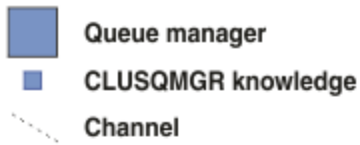


그림 20. 큐 관리자 클러스터

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.

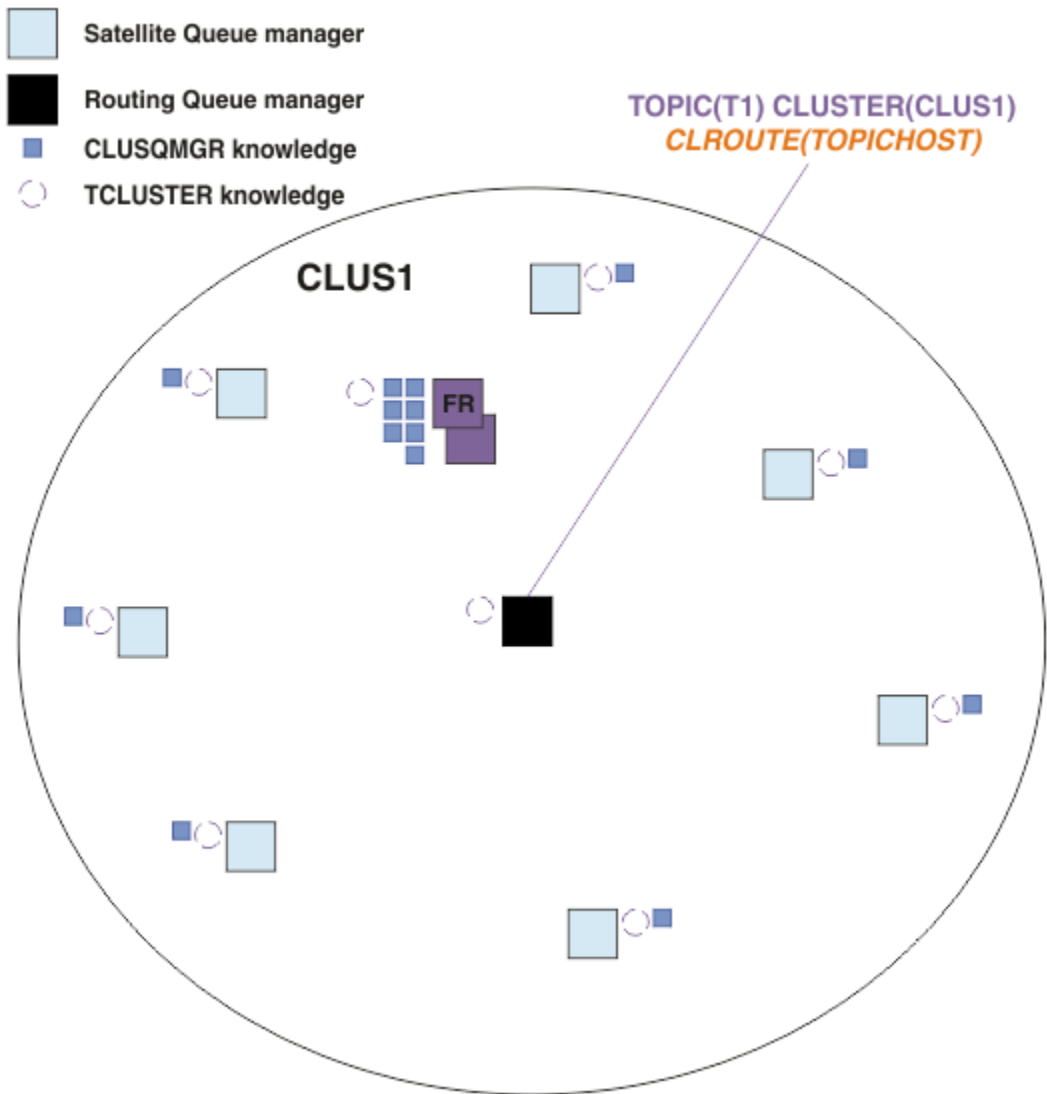


그림 21. 하나의 토픽 호스트에 정의된 하나의 토픽이 있는 토픽 호스트 라우트 발행/구독 클러스터

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.

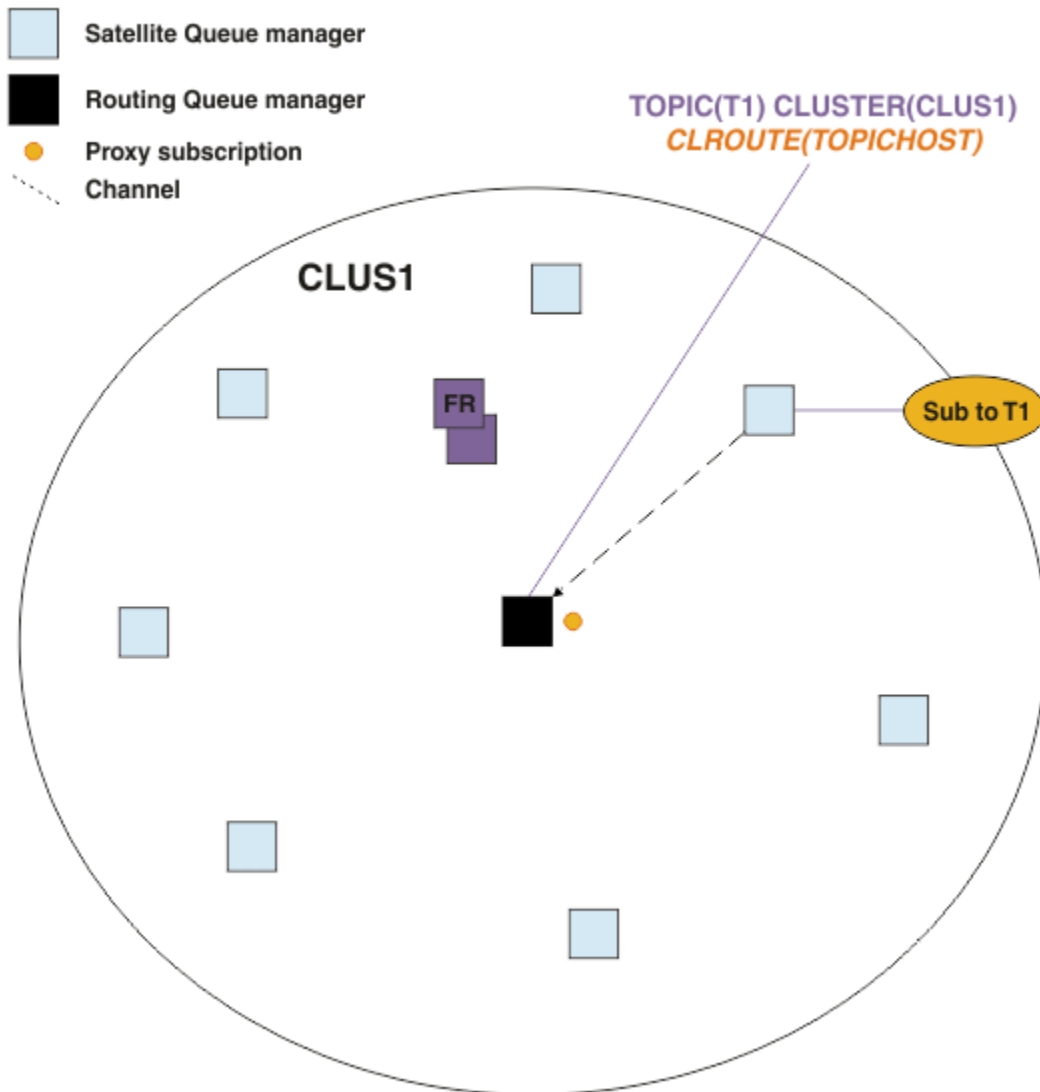


그림 22. 하나의 토픽 호스트에 정의된 하나의 토픽 및 하나의 구독자가 있는 토픽 호스트 라우트 발행/구독 클러스터

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 관리자(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.

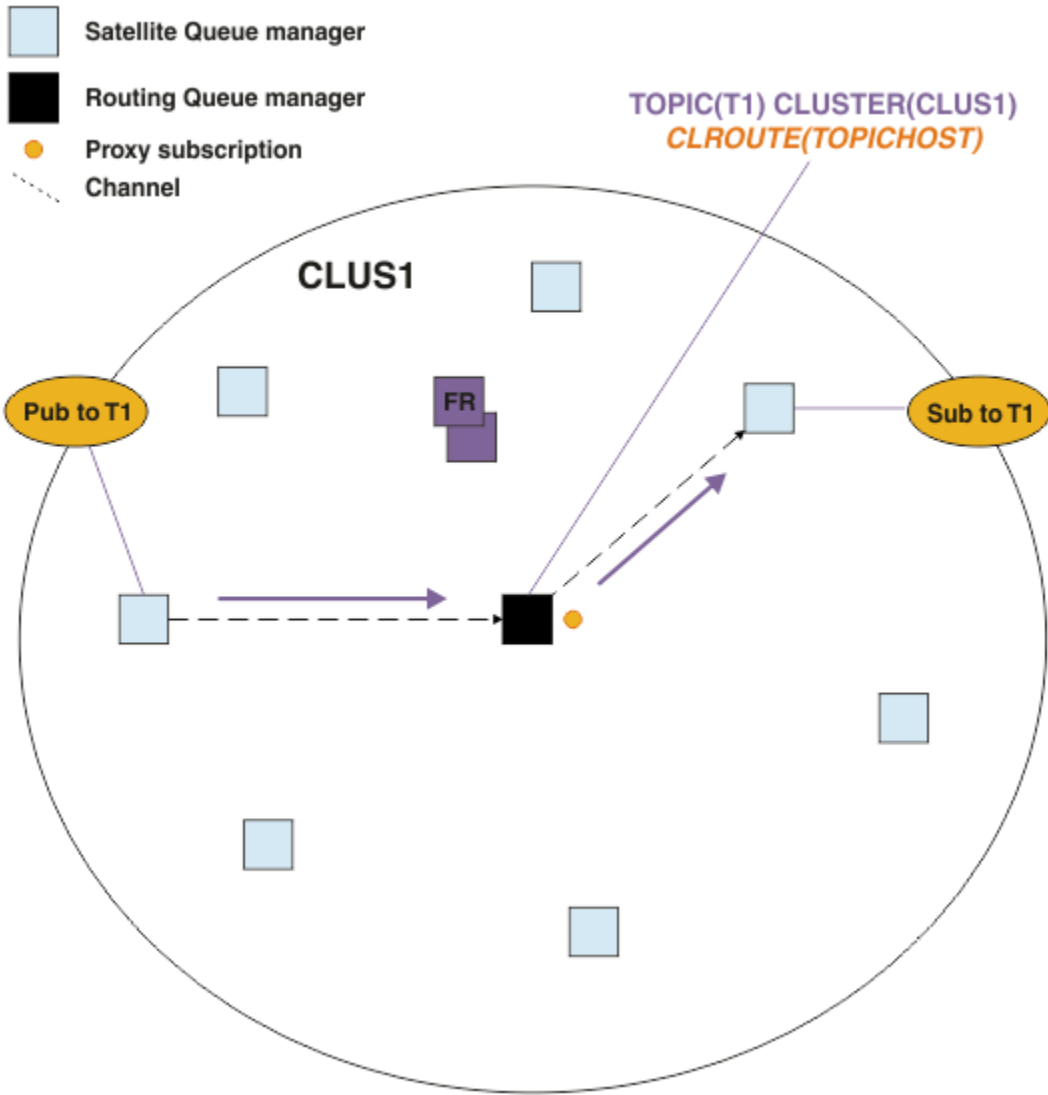


그림 23. 한 토픽, 한 구독자, 한 발행자가 있는 토픽 호스트 라우트 발행/구독 클러스터

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 호스팅할 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.

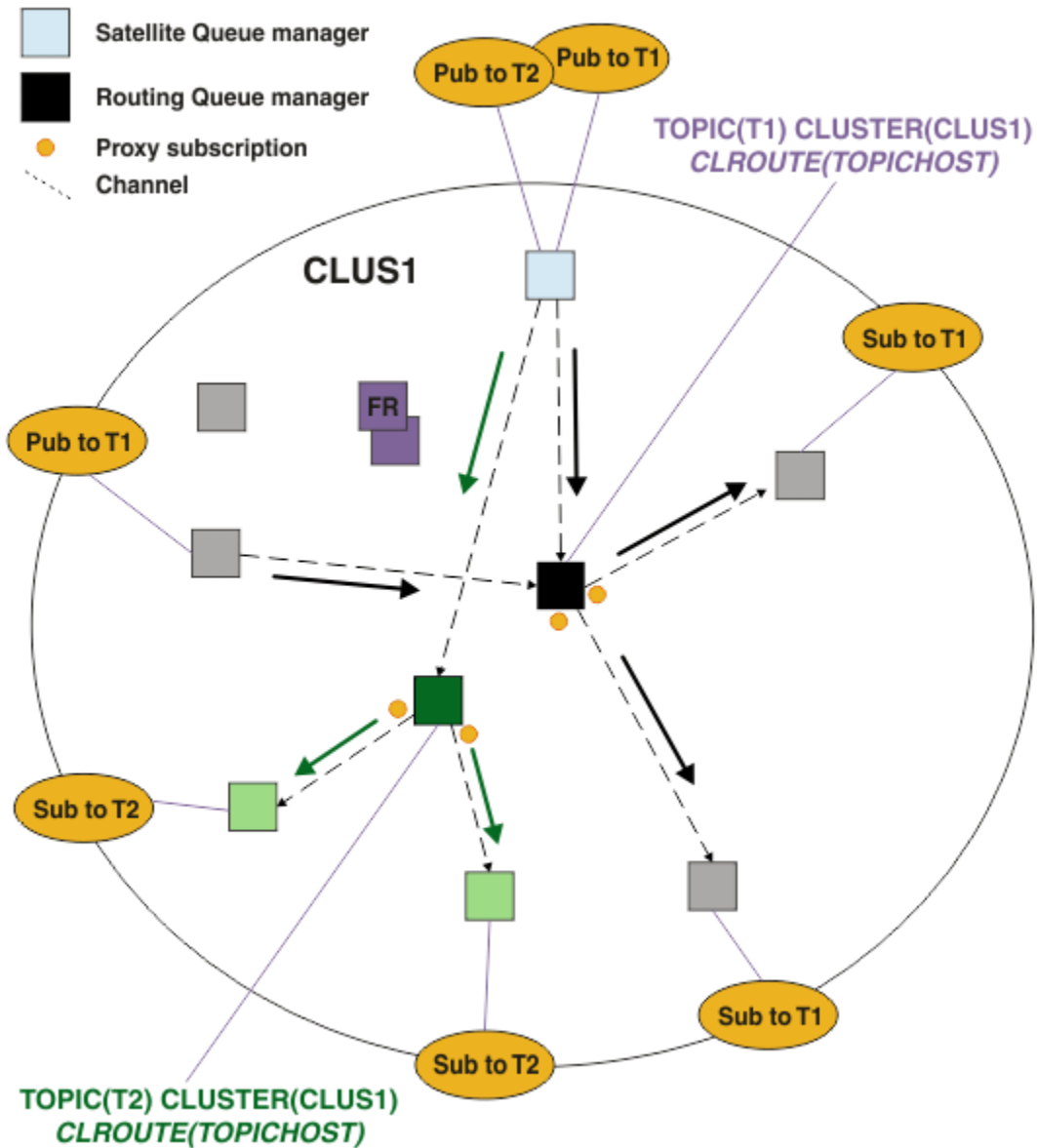


그림 24. 하나의 토픽 호스트에 정의된 두 개의 토픽이 있는 토픽 호스트 라우트 발행/구독 클러스터

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 구독](#)을 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 필요하다면 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.

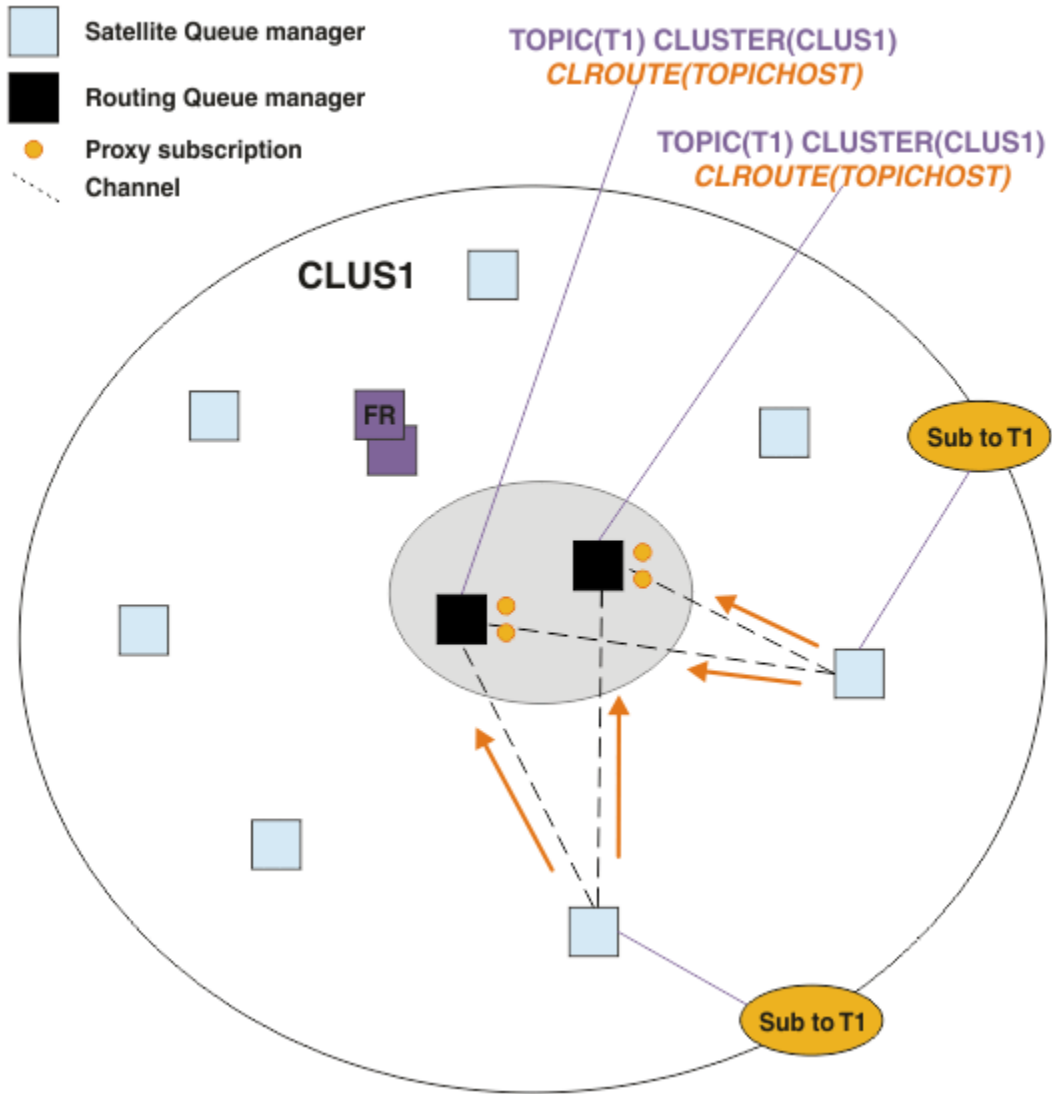


그림 25. 다중 토픽 호스트 발행/구독 클러스터에 프록시 구독 작성

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.

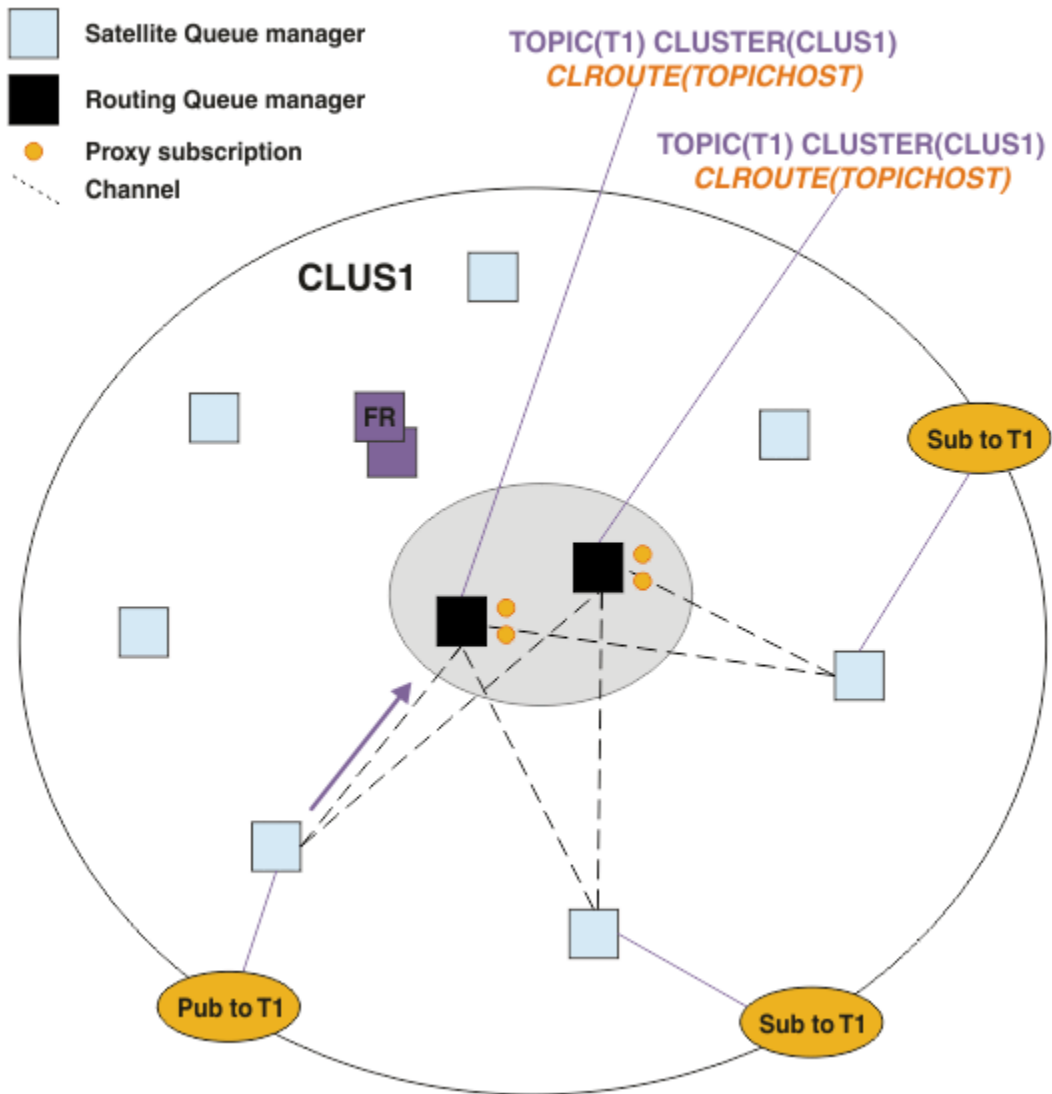


그림 26. 다중 토픽 호스트 발행/구독 클러스터의 발행물 수신

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

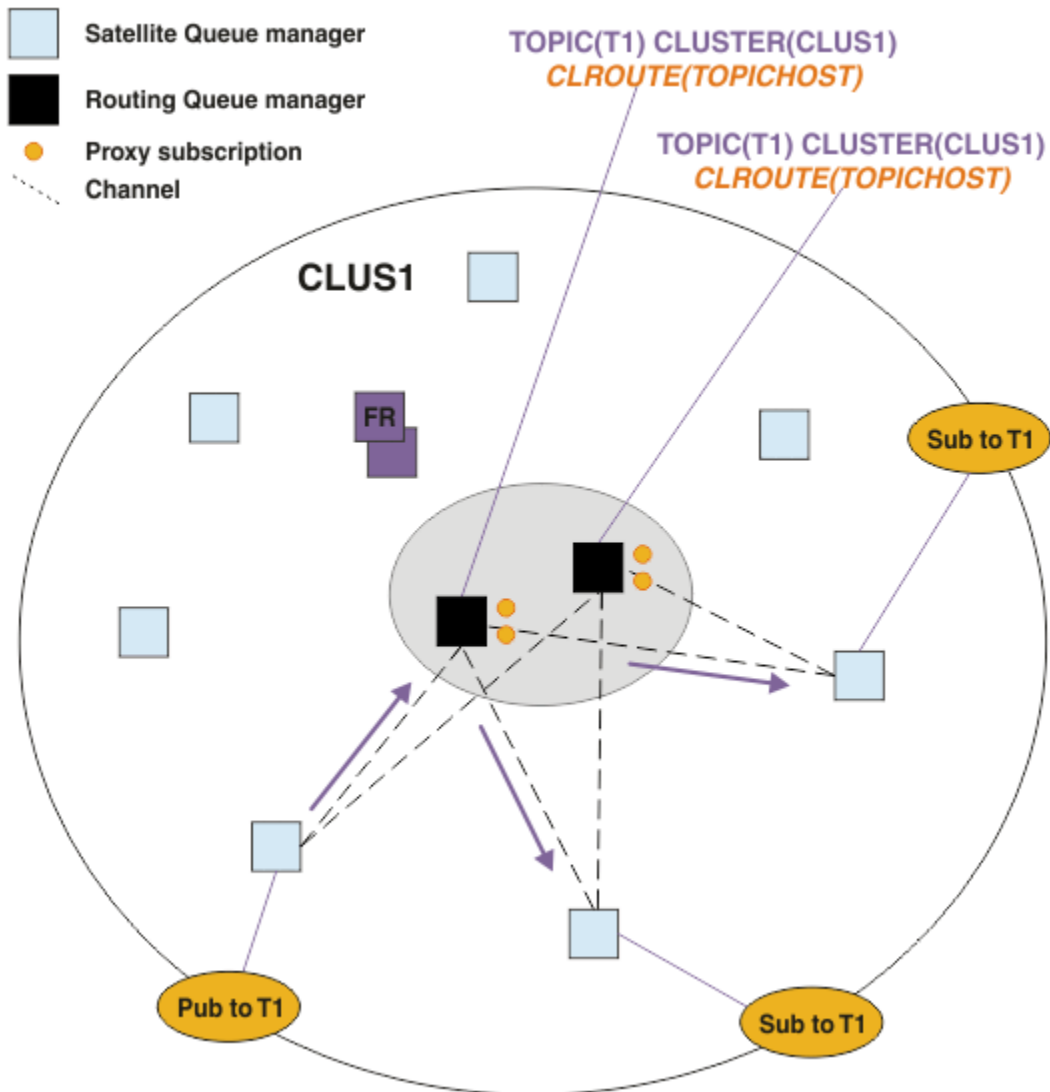


그림 27. 다중 토픽 호스트 발행/구독 클러스터의 구독자로 발행물 라우팅

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 토픽로지입니다.

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 접근 방식을 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 69 페이지의 『발행/구독 클러스터의 직접 라우팅』 및 94 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 67 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 64 페이지의 『분산 발행/구독 네트워크 계획』과 함께 다음 절에서 논의됩니다.

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

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

발행/구독 활동에 관련된 클러스터 큐 관리자의 수를 제한하는 이유

클러스터에서 발행/구독 메시지를 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 호스팅하는 큐 관리자에 집중하기 때문에 보통, 대형 클러스터에서는 토픽 호스트 라우트 모델이 더 잘 수행합니다. 이에 대한 예외는 토픽 정의를 호스팅하는 큐 관리자에서 구독 정보의 수동 새로 고침이 호출되어 토픽 호스트 큐 관리자가 클러스터의 모든 큐 관리자에 연결하는 경우입니다. [프록시 구독 재동기화](#)를 참조하십시오.

클러스터를 크기나 현재 로드로 인해 발행/구독에 사용할 수 없다고 설정하는 경우 우수 사례는 이 클러스터가 기지 않게 발행/구독 클러스터가 되지 않도록 하는 것입니다. **PSCLUS** 큐 관리자 특성을 사용하여 클러스터의 큐 관리자에 클러스터된 토픽을 추가하지 못하게 하십시오. 91 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 관리자(부터)의 채널.

AllSubs

구독이 작성된 모든 큐 관리자(부터)의 채널.

AllPubs

발행 애플리케이션이 연결된 모든 큐 관리자(부터)의 채널.

AllHosts

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

없음

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

참고:

1. 프록시 구독의 큐 관리자 새로 고치기가 이 큐 관리자로부터 이루어지는 경우 클러스터의 다른 모든 큐 관리자(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 관리자(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 관리자에 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 라우팅](#)을 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 구독](#)을 참조하십시오.

클러스터의 한 큐 관리자와 또 다른 큐 관리자 간 메시지 발행 비율이 높은 경우 두 지점 사이의 클러스터 채널 인 프라가 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 여딘가에 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 성능](#)을 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. [73 페이지의 『발행/구독 클러스터의 토픽 호스트 라우팅』](#)의 내용을 참조하십시오.

클러스터된 토픽과 로컬 토픽 오브젝트가 있으면 로컬 토픽이 우선합니다. [88 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 토픽로지의 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)**를 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 클러스터](#)를 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 메시지 정도 일시정지해야 합니다.

PROXYSUB

이 매개변수는 프록시 구독이 이루어지는 시기를 제어합니다.

- **FIRSTUSE**는 기본값으로, 분산 발행/구독 토픽로지에서 큐 관리자의 로컬 구독에 대한 응답으로 프록시 구독을 보내고 더 이상 필요하지 않을 때 이를 취소합니다. **FIRSTUSE**의 기본값에서 이 속성을 변경하려는 이유에 대한 자세한 내용은 [개별 프록시 구독 전달 및 모든 위치에서 발행](#)을 참조하십시오.
- 모든 위치에서 발행을 사용하려면 상위 레벨 토픽 오브젝트에 대한 **PROXYSUB** 매개변수를 **FORCE**로 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 과도한 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 성능](#)을 참조하십시오.

PUBSCOPE 및 SUBSCOPE

이 매개변수는 이 큐 관리자가 토픽로지(발행/구독 클러스터 또는 계층)의 큐 관리자에 발행물을 전파하거나 로컬 큐 관리자만으로 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 올바르게 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. [87 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 관리자에 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 이동](#)을 참조하십시오.

클러스터 토픽 정의를 변경해야 하는 경우에는 정의된 동일한 큐 관리자에 주의하여 수정하십시오. 다른 큐 관리자에 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 메시지가 토픽이 정의된 큐 관리자를 통해 라우팅됩니다. 따라서 클러스터에 있는 토픽 정의의 위치와 수가 중요합니다([73 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 가용성](#)을 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 추가](#)를 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(ASAPARENT)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 **DISABLED**로 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 메시지가 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 주어서 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 **DISABLED**로 설정하여 클러스터된 발행/구독 기능을 완전히 사용 불가능하게 할 수 있습니다.

69 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 67 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 **DISABLED**로 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 올바르게 구성된 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 **DISABLED**로 설정해야 합니다. 하지만 큐 관리자가 언제든지 클러스터를 조인하고 떠날 수 있는 경우처럼 실제로 이 상태를 확립하고 유지하기는 어려울 수 있습니다. 적어도 큐 관리자의 모든 전체 저장소에서 **PSCLUS**가 **DISABLED**로 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 **ENABLED**로 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 **ENABLED**에서 **DISABLED**로 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 **DISABLED**로 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

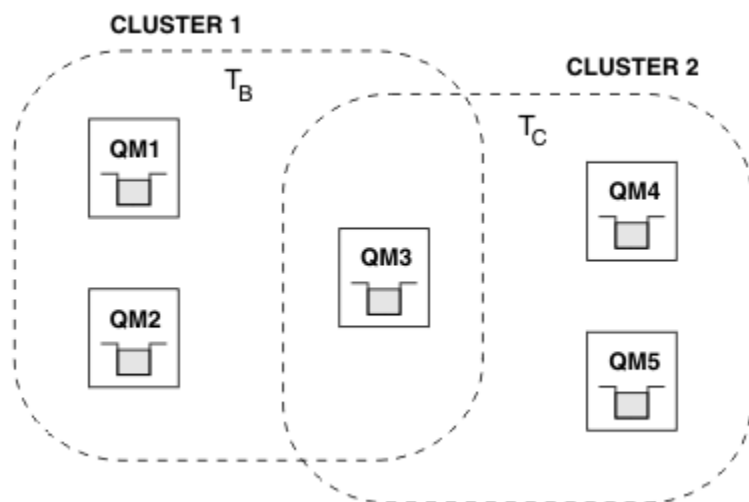


그림 28. 중첩 클러스터: 각각이 서로 다른 토픽을 구독하는 두 개의 클러스터

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_B 만 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/Football임) 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 결합](#)을 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. 67 페이지의 『발행/구독 클러스터 디자인』의 내용을 참조하십시오.

발행/구독 클러스터에서 보유한 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 와일드카드 처리됩니다. [와일드카드 및 클러스터 토픽](#)을 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 프로시 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 프로시 구독을 일관되게 보내지 않습니다.
- 이 기간 중 삭제된 이 큐 관리자의 클러스터 토픽에 대한 구독이 클러스터의 기타 멤버에서 프로시 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 프로시 구독 재동기화에 설명된 대로 프로시 구독을 재동기화하십시오. 모든 프로시 구독이 제대로 재동기화되면 발행/구독 애플리케이션을 다시 시작하십시오.

REFRESH CLUSTER 명령 완료에 시간이 오래 걸리는 경우 `SYSTEM.CLUSTER.COMMAND.QUEUE`의 `CURDEPTH`를 보고 상황을 모니터링하십시오.

관련 개념

62 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 안 됩니다. 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

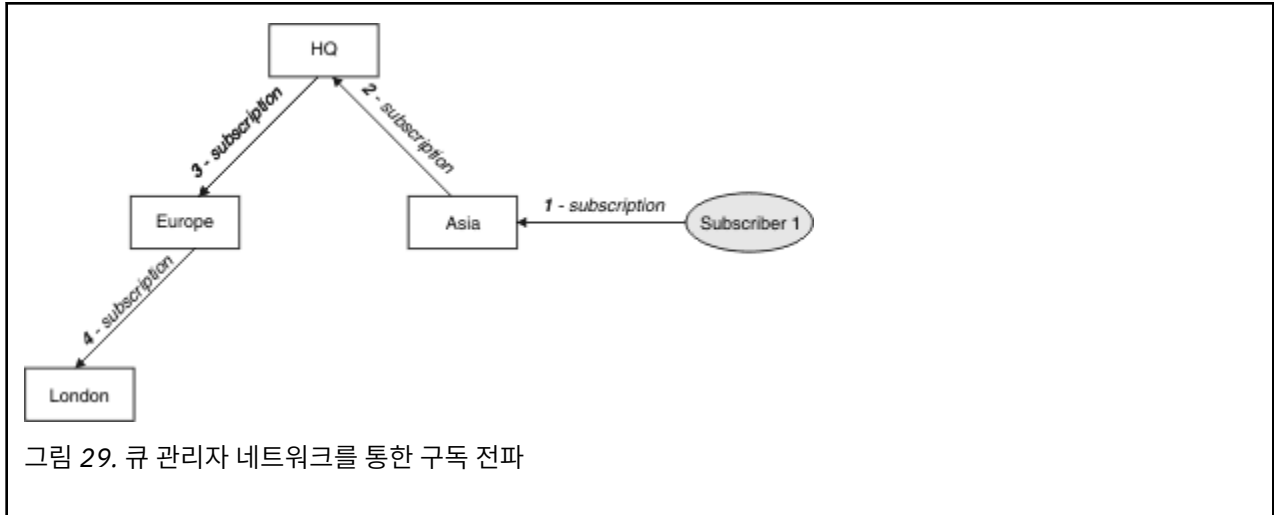
발행/구독 계층의 라우팅

분산 큐 관리자 토폴로지가 발행/구독 계층이고 큐 관리자에서 구독이 이루어지는 경우 프로시 구독은 기본적으로 계층의 모든 큐 관리자에 작성됩니다. 큐 관리자에 수신된 발행물은 계층을 통해 일치하는 구독을 호스트하는 각 큐 관리자에 라우팅됩니다.

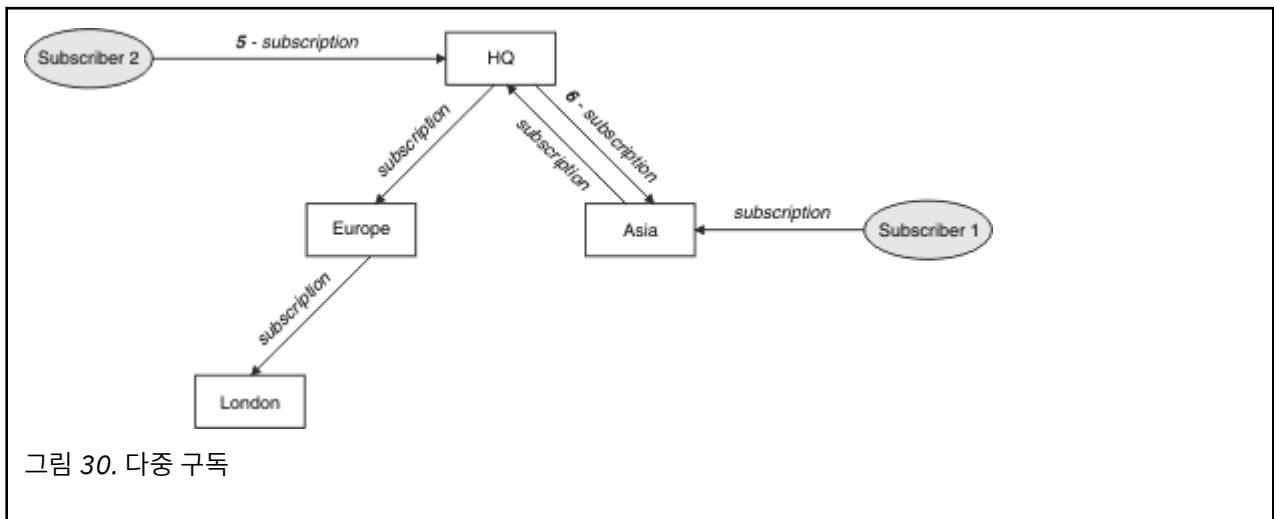
발행/구독 계층의 큐 관리자와 클러스터 간에 메시지가 라우팅되는 방식에 대한 소개는 분산 발행/구독 네트워크를 참조하십시오.

분산 발행/구독 계층에서 토픽에 대한 구독이 이루어지는 경우 큐 관리자는 연결된 큐 관리자에 구독이 전파되는 프로세스를 관리합니다. 프로시 구독은 네트워크의 모든 큐 관리자에 플로우됩니다. 프로시 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

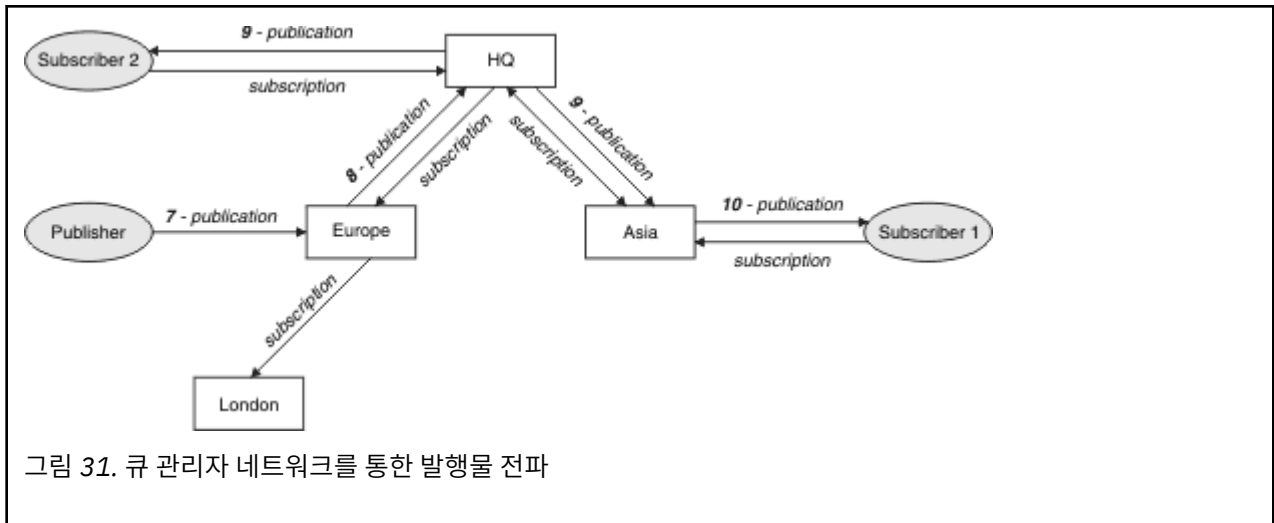
특정 토픽에 대한 구독을 등록합니다. *Asia* 큐 관리자의 이 구독에 대한 프록시 구독은 네트워크의 다른 모든 큐 관리자(2,3,4)에게 전달됩니다.



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 95 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(95 페이지의 그림 29의 3단계 참조).



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 95 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.



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 69 페이지의 『발행/구독 클러스터의 직접 라우팅』 및 73 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 정의](#)를 참조하십시오.

발행/구독 시스템 큐의 속성은 97 페이지의 표 7에 표시되어 있습니다.

표 7. 발행/구독 시스템 큐의 속성	
속성	기본값
DEFPSIST	예
DEFSOPT	SHARED
MAXMSGL	<p>Multi 멀티플랫폼의 경우: ALTER QMGR 명령의 MAXMSGL 매개변수 값</p> <p>z/OS z/OS의 경우: 4194304 (즉, 4MB)</p>
MAXDEPTH	999999999
SHARE	해당사항 없음
<p>z/OS</p> <p>z/OS</p> STGCLASS	이 속성은 z/OS 플랫폼에서만 사용됩니다.

참고: 애플리케이션이 넣은 메시지를 포함하는 유일한 큐는 SYSTEM.INTER.QMGR.PUBS입니다. **MAXDEPTH**는(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 MQ의 시스템 요구사항](#)의 내용을 참조하십시오.

IBM MQ는 파일 시스템에 큐 관리자 데이터를 저장합니다. 다음 링크를 사용하여 IBM MQ에 사용할 디렉토리 구조의 계획 및 구성을 알아보십시오.

- [102 페이지의 『멀티플랫폼에서 파일 시스템 지원 계획』](#)
- [103 페이지의 『멀티플랫폼에서 공유 파일 시스템에 대한 요구사항』](#)
- [112 페이지의 『멀티플랫폼에서 IBM MQ 파일 공유』](#)
- [Linux](#) [AIX](#) [114 페이지의 『AIX and Linux 시스템의 디렉토리 구조』](#)
- [Windows](#) [122 페이지의 『Windows 시스템의 디렉토리 구조』](#)
- [IBM i](#) [125 페이지의 『IBM i의 디렉토리 구조』](#)

AIX and Linux에서의 시스템 자원, 공유 메모리, 프로세스 우선순위에 대한 정보를 보려면 다음 링크를 사용하십시오.

- [Linux](#) [AIX](#) [129 페이지의 『IBM MQ 및 UNIX System V IPC 자원』](#)
- [AIX](#) [129 페이지의 『AIX의 공유 메모리』](#)
- [Linux](#) [AIX](#) [129 페이지의 『IBM MQ 및 UNIX 프로세스 우선순위』](#)

로그 파일에 대한 정보를 보려면 다음 링크를 사용하십시오.

- [128 페이지의 『멀티플랫폼에서 순환 또는 선형 로깅 선택』](#)
- [로그의 크기 계산](#)

관련 개념

[130 페이지의 『z/OS에서 IBM MQ 환경 계획』](#)

IBM MQ 환경을 계획할 때에는 데이터 세트, 페이지 세트, Db2, 커플링 기능, 로깅 및 백업 기능 필요에 대한 자원 요구사항을 고려해야 합니다. 이 주제를 사용하여 IBM MQ가 실행하는 환경을 계획하십시오.

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** 99 페이지의 『Long Term Support의 디스크 공간 요구사항』
- ▶ **CD** 100 페이지의 『Continuous Delivery의 디스크 공간 요구사항』

Long Term Support의 디스크 공간 요구사항

LTS ▶ **V 9.3.0**

표 8. IBM MQ 멀티플랫폼용 디스크 공간 요구사항 Long Term Support

플랫폼	클라이언트 설치 99 페이지의 『1』	서버 설치 99 페이지의 『2』	전체 설치 100 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(64 비트 설치) 100 페이지의 『4』	295MB	415MB	2310MB

참고:

- 클라이언트 설치에 다음 컴포넌트를 포함합니다.
 - 런타임
 - 클라이언트
- 서버 설치에 다음 컴포넌트를 포함합니다.
 - 런타임
 - 서버

3. 전체 설치에 사용 가능한 모든 컴포넌트를 포함합니다.
4. **Windows** 여기에 나열된 모든 컴포넌트가 Windows 시스템에 설치 가능한 기능이 아닙니다. 일부 기능은 때로 기타 기능에 포함됩니다. Windows 시스템의 IBM 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

V 9.3.0 **CD**

표 9. IBM MQ 멀티플랫폼용 디스크 공간 요구사항 Continuous Delivery			
플랫폼/CD 릴리스	클라이언트 설치 102 페이지의 『1』	서버 설치 102 페이지의 『2』	전체 설치 102 페이지의 『3』
AIX AIX			
V 9.3.0 IBM MQ 9.3.0	330MB	375MB	1760MB
V 9.3.1 IBM MQ 9.3.1	340MB	375MB	1815MB
V 9.3.2 IBM MQ 9.3.2	355MB	390MB	1440 MB
V 9.3.3 IBM MQ 9.3.3	355MB	390MB	1440 MB
V 9.3.4 IBM MQ 9.3.4	355MB	390MB	1440 MB
V 9.3.5 IBM MQ 9.3.5	355MB	390MB	1440 MB
Linux Linux for x86-64(64비트)			
V 9.3.0 IBM MQ 9.3.0	265MB	295MB	2010MB
V 9.3.1 IBM MQ 9.3.1	275MB	295MB	2010MB
V 9.3.2 IBM MQ 9.3.2	280MB	295MB	1195MB
V 9.3.3 IBM MQ 9.3.3	280MB	295MB	1195MB

표 9. IBM MQ 멀티플랫폼용 디스크 공간 요구사항 Continuous Delivery (계속)			
플랫폼/CD 릴리스	클라이언트 설치 102 페이지의 『1』	서버 설치 102 페이지의 『2』	전체 설치 102 페이지의 『3』
> V 9.3.4 IBM MQ 9.3.4	280MB	295MB	1195MB
> V 9.3.5 IBM MQ 9.3.5	280MB	295MB	1195MB
Linux Linux on Power Systems - Little Endian			
> V 9.3.0 IBM MQ 9.3.0	170MB	190MB	1350MB
> V 9.3.1 IBM MQ 9.3.1	170MB	195MB	1400MB
> V 9.3.2 IBM MQ 9.3.2	170MB	195MB	1075MB
> V 9.3.3 IBM MQ 9.3.3	170MB	195MB	1075MB
> V 9.3.4 IBM MQ 9.3.4	170MB	195MB	1075MB
> V 9.3.5 IBM MQ 9.3.5	170MB	195MB	1075MB
Linux IBM Z의 경우 Linux			
> V 9.3.0 IBM MQ 9.3.0	255MB	290MB	1435MB
> V 9.3.1 IBM MQ 9.3.1	255MB	290MB	1485MB
> V 9.3.2 IBM MQ 9.3.2	260MB	290MB	1160MB
> V 9.3.3 IBM MQ 9.3.3	260MB	290MB	1160MB
> V 9.3.4 IBM MQ 9.3.4	260MB	290MB	1160MB
> V 9.3.5 IBM MQ 9.3.5	260MB	290MB	1160MB
Windows Windows(64비트 설치) 102 페이지의 『4』			
> V 9.3.0 IBM MQ 9.3.0	290MB	410MB	2095MB
> V 9.3.1 IBM MQ 9.3.1	295MB	415MB	2310MB

표 9. IBM MQ 멀티플랫폼용 디스크 공간 요구사항 Continuous Delivery (계속)			
플랫폼/CD 릴리스	클라이언트 설치 102 페이지의 『1』	서버 설치 102 페이지의 『2』	전체 설치 102 페이지의 『3』
V 9.3.2 IBM MQ 9.3.2	300MB	415MB	1785MB
V 9.3.3 IBM MQ 9.3.3	300MB	415MB	1785MB
V 9.3.4 IBM MQ 9.3.4	300MB	415MB	1785MB
V 9.3.5 IBM MQ 9.3.5	300MB	415MB	1785MB

참고:

- 클라이언트 설치하는 다음 컴포넌트를 포함합니다.
 - 런타임
 - 클라이언트
- 서버 설치하는 다음 컴포넌트를 포함합니다.
 - 런타임
 - 서버
- 전체 설치하는 사용 가능한 모든 컴포넌트를 포함합니다.
- Windows** 여기에 나열된 모든 컴포넌트가 Windows 시스템에 설치 가능한 기능이 아닙니다. 일부 기능은 때로 기타 기능에 포함됩니다. Windows 시스템의 IBM 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 **amqmfscck** 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 **amqmfscck** 명령이 필요하지 않습니다.

관련 태스크

104 페이지의 『멀티플랫폼에서 공유 파일 시스템 작동 확인』

amqmfscck를 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 커밋되는 데이터와 동기화시킬 수 있어야 합니다. 트랜잭션 시스템에서는 다른 처리를 계속하기 전에 일부 쓰기가 안전하게 커밋되었는지 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 잠금 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 Windows 이외의 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.

니다. 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 MQ에 **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

amqmfscck를 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 **amqmfscck**를 실행하려면 IBM MQ 관리자여야 합니다.

이 태스크 정보

103 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 명령문](#)은 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 네트워크 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 [amqmfscck](#)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 [amqmfscck](#)와 동시에 실행하려면 110 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 `qmdata`를 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:

```
amqmfscck /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:

```
amqmfscck -c /shared/qmdata
```

IBM MQ 서버 2:

```
amqmfscck -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:

```
amqmfscck -w /shared/qmdata
```

IBM MQ 서버 2:

```
amqmfscck -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 시뮬레이션하기 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:

```
amqmfscck -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 씁니다.

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
amqmfscck -a /shared/qmdata	

표 10. 두 개의 서버에서 동시에 데이터 무결성 검사기 실행 (계속)	
IBM MQ 서버 1	IBM MQ 서버 2
<pre>Please start this program on a second machine with the same parameters. File lock acquired. Start a second copy of this program with the same parameters on another server. Writing data into test file. To increase the effectiveness of the test, interrupt the writing by ending the process, temporarily breaking the network connection to the networked storage, rebooting the server or turning off the power.</pre>	<pre>amqmfscck -a /shared/qmdata Waiting for lock... Waiting for lock... Waiting for lock... Waiting for lock... Waiting for lock... Waiting for lock...</pre>
<pre>Turn the power off here.</pre>	
	<pre>File lock acquired. Reading test file Checking the integrity of the data read. Appending data into the test file after data already found. The test file is full of data. It is ready to be inspected for data integrity.</pre>

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 **amqmfscck**의 **-x** 옵션을 사용하여 테스트 파일을 삭제하십시오. 더 큰 테스트 파일로 처음부터 테스트를 시도하십시오.

c) 테스트 파일에서 데이터의 무결성을 확인하십시오.

IBM MQ 서버 2:

```
amqmfscck -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:

```
amqmfscck -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

```
> amqmfscck /shared/qmdata
The tests on the directory completed successfully.
```

한 서버의 기본 파일 잠금 테스트 실패

```
> amqmfscck /shared/qmdata
AMQ6245: Error Calling 'write()[2]' on file '/shared/qmdata/amqmfscck.lck' error '2'.
```

두 개의 서버에서 잠금 테스트 성공

표 11. 두 개의 서버에서 잠금 성공	
IBM MQ 서버 1	IBM MQ 서버 2
<pre>> amqmfscck -w /shared/qmdata Please start this program on a second machine with the same parameters. Lock acquired. Press Return or terminate the program to release the lock.</pre>	
	<pre>> amqmfscck -w /shared/qmdata Waiting for lock...</pre>
<pre>[Return pressed] Lock released.</pre>	
	<pre>Lock acquired. The tests on the directory completed successfully</pre>

세 가지 예의 두 번째 세트는 상세 모드를 사용하여 동일한 명령을 표시합니다.

한 서버의 기본 파일 잠금 테스트 성공

```
> amqmfscck -v /shared/qmdata
System call: stat("/shared/qmdata")'
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fchmod(fd, 0666)
System call: fstat(fd)
System call:fcntl(fd, F_SETLK, F_WRLCK)
System call: write(fd)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd1 = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd1, F_SETLK, F_RDLCK)
System call: fd2 = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd2, F_SETLK, F_RDLCK)
System call: close(fd2)
System call: write(fd1)
System call: close(fd1)
The tests on the directory completed successfully.
```

한 서버의 기본 파일 잠금 테스트 실패

```
> amqmfscck -v /shared/qmdata
System call: stat("/shared/qmdata")
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fchmod(fd, 0666)
System call: fstat(fd)
System call:fcntl(fd, F_SETLK, F_WRLCK)
System call: write(fd)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd, F_SETLK, F_RDLCK)
System call: fdSameFile = open("/shared/qmdata/amqmfscck.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/amqmfscck.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
<pre>> amqmfscck -wv /shared/qmdata Calling 'stat("/shared/qmdata")' Calling 'fd = open("/shared/qmdata/ amqmfscck.lkw", O_EXCL O_CREAT O_RDWR, 0666)' Calling 'fchmod(fd, 0666)' Calling 'fstat(fd)' Please start this program on a second machine with the same parameters. Calling 'fcntl(fd, F_SETLK, F_WRLCK)' Lock acquired. Press Return or terminate the program to release the lock.</pre>	
	<pre>> amqmfscck -wv /shared/qmdata Calling 'stat("/shared/qmdata")' Calling 'fd = open("/shared/qmdata/ amqmfscck.lkw", O_EXCL O_CREAT O_RDWR,0666)' Calling 'fd = open("/shared/qmdata/amqmfscck.lkw, O_RDWR, 0666)' Calling 'fcntl(fd, F_SETLK, F_WRLCK) 'Waiting for lock...</pre>

표 12. 두 개의 서버에서 잠금 성공 - 상세 모드 (계속)	
IBM MQ 서버 1	IBM MQ 서버 2
[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 *amqmfscck*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.

104 페이지의 『멀티플랫폼에서 공유 파일 시스템 작동 확인』에 있는 단계 105 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 *amqmfscck*와 병렬로 실행하여 실패 중에 큐 관리자가 메시지 무결성을 유지보수하는지 확인하십시오.

프로시저

- 프로시저의 105 페이지의 『1』 단계에서 작성한 파일 시스템을 사용하여 다른 서버 QM1에 다중 인스턴스 큐 관리자를 작성하십시오.
다중 인스턴스 큐 관리자 작성을 참조하십시오.
- 큐 관리자의 가용성을 높게 하여 두 서버 모두에서 큐 관리자를 시작하십시오.

서버 1에서:

```
strmqm -x QM1
```

서버 2에서:

```
strmqm -x QM1
```

- amqsfhac*를 실행할 클라이언트 연결을 설정하십시오.
 - IBM MQ 설치 확인의 프로시저 또는 다시 연결 가능한 클라이언트 샘플의 예제 스크립트를 사용하여 클라이언트 연결을 설정하십시오.
 - 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 `server2`는 두 개 서버의 호스트 이름이고 `2345`는 채널 리스너가 대기 중인 포트입니다. 일반적으로 기본값은 `1414`입니다. `1414`를 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.

[104 페이지의 『멀티플랫폼에서 공유 파일 시스템 작동 확인』](#)의 [106 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

[111 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
```

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 커밋되었는지 여부에 따라 결과가 다릅니다.

관련 참조

[amqmfscck\(파일 시스템 검사\)](#)

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. 112 페이지의 그림 32에 표시된 각 qmgrs 및 log 디렉토리에 포함된 자체 큐 관리자 디렉토리 트리에 대한 독점 액세스를 요구합니다.

112 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.

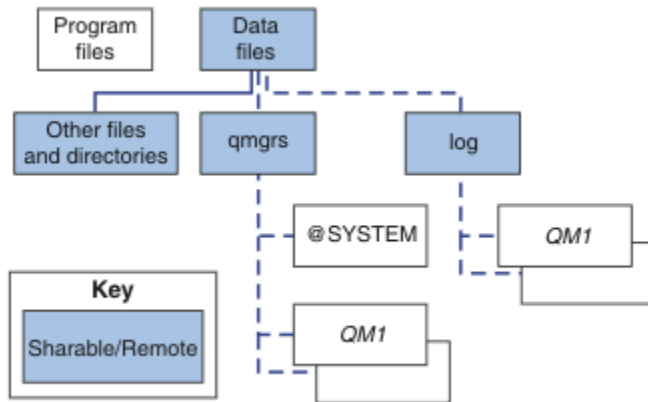


그림 32. IBM MQ 디렉토리 구조의 전체 보기

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/mqm에 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 스탠자에 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 [112 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 `mqm.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 [112 페이지의 그림 32](#)의 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.ini`에 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([113 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. [113 페이지의 그림 34](#)의 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 MQ는 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 가져옵니다. 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 114 페이지의 그림 35에 표시된 디렉토리 구조를 작성하십시오. 여기서 R는 제품의 릴리스입니다. 이는 IBM MQ 시스템에서 작성된 큐 관리자의 일반적인 디렉토리 구조입니다. 일부 디렉토리, 파일, .ini 속성 설정은 명확한 표시를 위해 생략되며 다른 큐 관리자 이름은 맵글링을 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.

114 페이지의 그림 35에서는 IBM MQ v7.R on AIX 여기서 R는 제품의 릴리스입니다. 대체 다중 인스턴스 구성의 예는 118 페이지의 『AIX and Linux 시스템의 예 디렉토리 구성』의 내용을 참조하십시오.

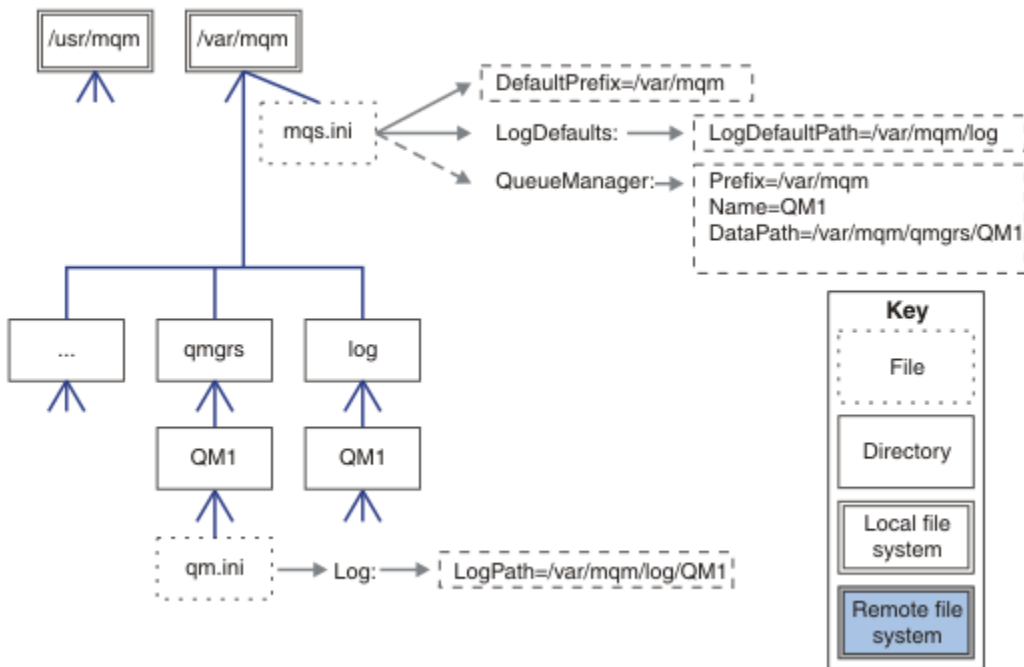


그림 35. AIX and Linux 시스템의 기본 IBM MQ 디렉토리 구조 예제

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/mqm을 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/qmgrs에 작성되고 로그 파일 디렉토리는 LogDefaultPath/log에 작성됩니다. LogDefaultPath 및 DefaultPrefix는 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. Prefix가 DefaultPrefix보다 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

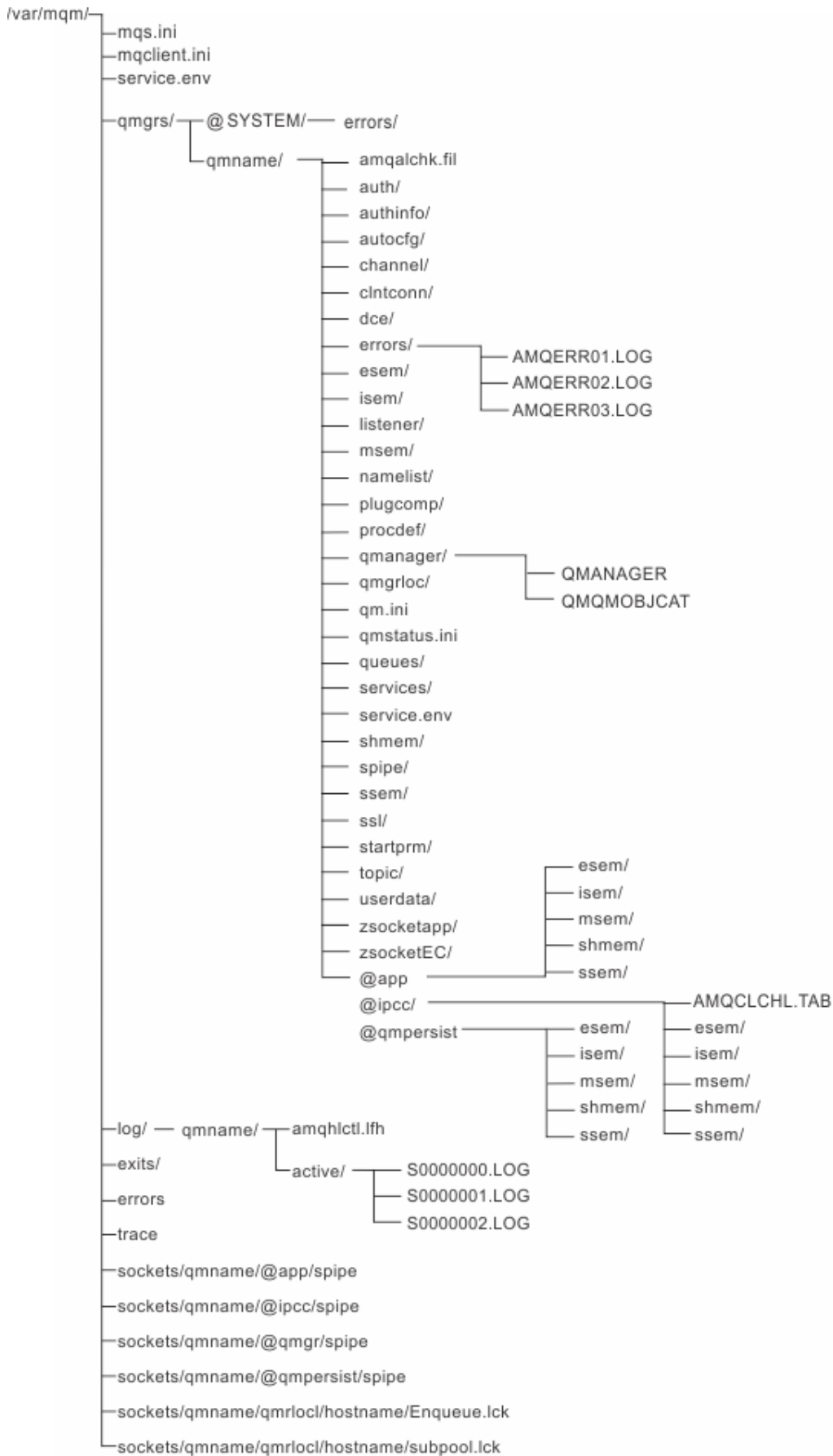
AIX and Linux 시스템의 디렉토리 콘텐츠

큐 관리자와 연관된 디렉토리의 콘텐츠

제품 파일 위치에 대한 자세한 정보는 [설치 위치 선택을 참조하십시오](#).

대체 디렉토리 구성에 대한 정보는 [102 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/

/var/mqm 디렉토리에는 개별 큐 관리자가 아닌 IBM MQ 설치에 전체적으로 적용되는 구성 파일 및 출력 디렉토리가 포함되어 있습니다.

표 13. AIX and Linux에서 /var/mqm 디렉토리의 문서화된 콘텐츠	
디렉토리 또는 파일 이름	콘텐츠
<u>mqs.ini</u>	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 셸에서 설정되고 내보내는지 확인하십시오.
<u>mqclient.ini</u>	IBM MQ MQI client 프로그램이 읽는 기본 클라이언트 구성 파일. MQCLNTCF 환경 변수를 사용하여 수정 가능한 파일 경로.
<u>service.env</u>	서비스 프로세스에 대한 시스템 범위 환경 변수를 포함합니다. 수정된 파일 경로.
<u>errors/</u>	시스템 범위 오류 로그 및 FFST 파일. 수정된 디렉토리 경로. FFST: IBM MQ for UNIX 및 Linux 시스템도 참조하십시오.
<u>sockets/</u>	시스템 용도로만 각 큐 관리자에 대한 정보를 포함합니다.
<u>trace/</u>	추적 파일. 수정된 디렉토리 경로.
<u>웹/</u>	mqweb 서버 디렉토리
<u>exits/</u>	사용자 채널 엑시트 프로그램을 포함한 기본 디렉토리. mqs.ini 파일의 ApiExit 스탠자에서 수정 가능한 위치.
<u>exits64/</u>	

/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 Linux에서 /var/mqm/qmgrs/qmname 디렉토리의 문서화된 콘텐츠	
디렉토리 또는 파일 이름	콘텐츠
<u>qm.ini</u>	큐 관리자가 시작될 때 읽는 큐 관리자 구성 파일.
<u>errors/</u>	큐 관리자 범위 오류 로그. qmname = @system 은 알 수 없거나 사용 불가능한 큐 관리자에 대한 채널 관련 메시지를 포함합니다.
<u>@ipcc/AMQCLCHL.TAB</u>	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 Linux에서 /var/mqm/qmgrs/qmname 디렉토리의 문서화된 콘텐츠 (계속)	
디렉토리 또는 파일 이름	콘텐츠
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/	
...	
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 Linux에서 /var/mqm/log/qmname 디렉토리의 문서화된 콘텐츠	
디렉토리 또는 파일 이름	콘텐츠
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 99 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/errors에 로컬로 저장하는 것이 더 좋습니다.

119 페이지의 그림 36은 대체 IBM MQ 디렉토리 구조가 파생된 템플릿입니다. 템플릿에서 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 생각하십시오. IBM MQ에서 경로에 큐 관리자 이름을 추가합니다.

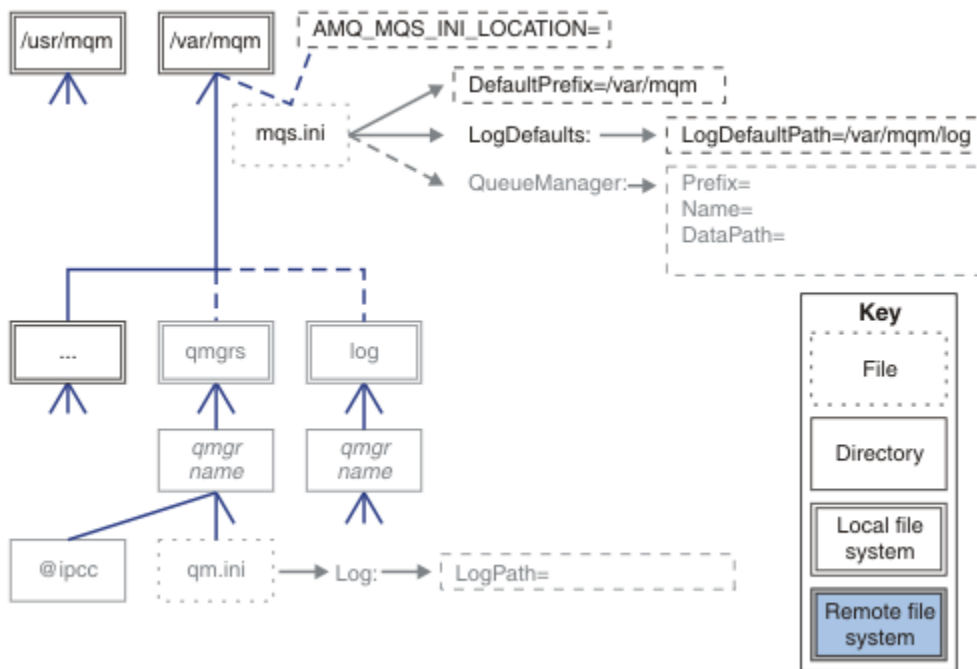


그림 36. 디렉토리 구조 패턴 템플릿

IBM MQ의 일반 디렉토리 구조

120 페이지의 그림 37는 **crtmqm QM1** 명령을 실행하여 IBM MQ에서 작성되는 기본 디렉토리 구조입니다.

mqs.ini 파일에는 DefaultPrefix의 값을 참조하여 작성된 QM1 큐 관리자에 대한 스탠자가 있습니다. qm.ini 파일의 Log 스탠자에는 mqs.ini의 LogDefaultPath를 참조하여 설정된 LogPath의 값이 있습니다.

선택적 **crtmqm** 매개변수를 사용하여 DataPath 및 LogPath의 기본값을 대체하십시오.

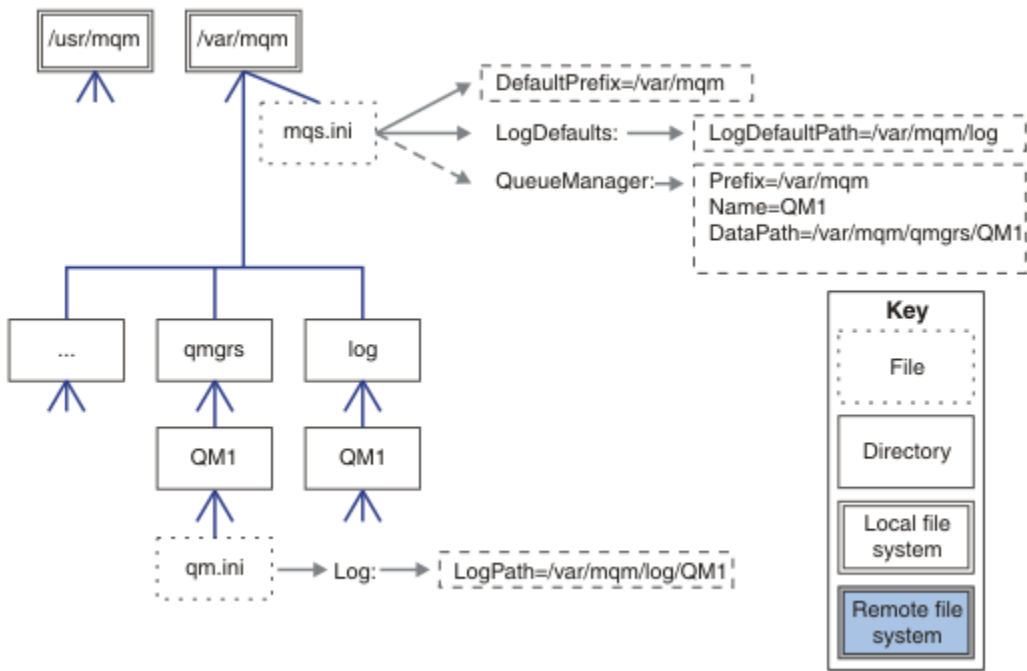


그림 37. AIX and Linux 시스템의 기본 IBM MQ 디렉토리 구조 예제

기본 qmgrs 및 log 디렉토리 공유

121 페이지의 『모두 공유』의 대안은 `qmgrs` 및 `log` 디렉토리를 별도로 공유하는 것입니다 (120 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.

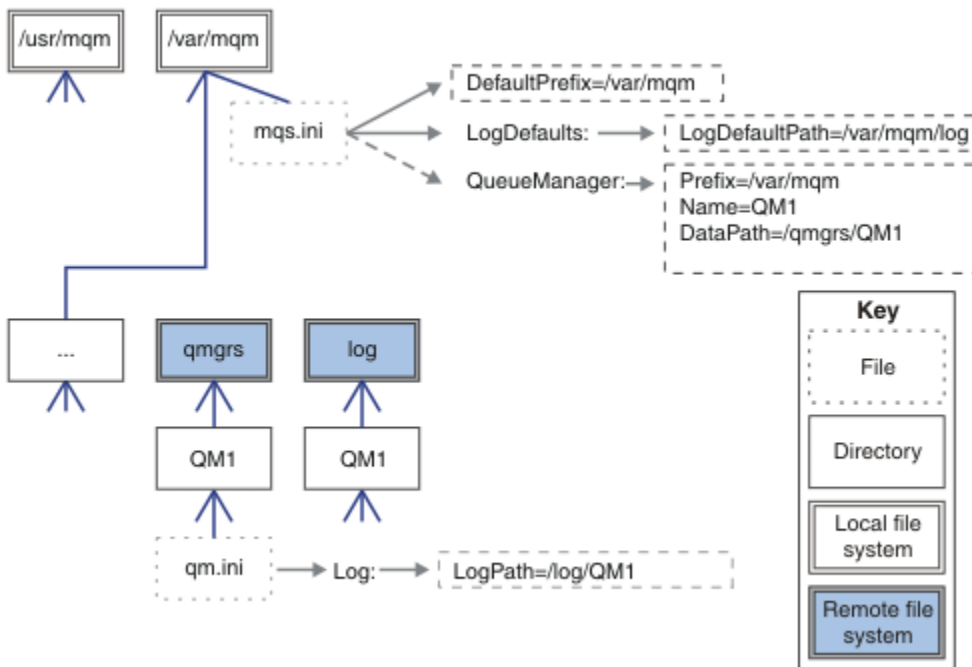


그림 38. qmgrs 및 log 디렉토리 공유

이름 지정된 qmgrs 및 log 디렉토리 공유

121 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 121 페이지의 그림 39의 설명과 정확히 같습니다.
2. 기본 경로를 변경하지 않은 채 그대로 두고 `crtmqm - ld /ha/log - md /ha/qmgrs QM1` 명령을 실행하십시오.

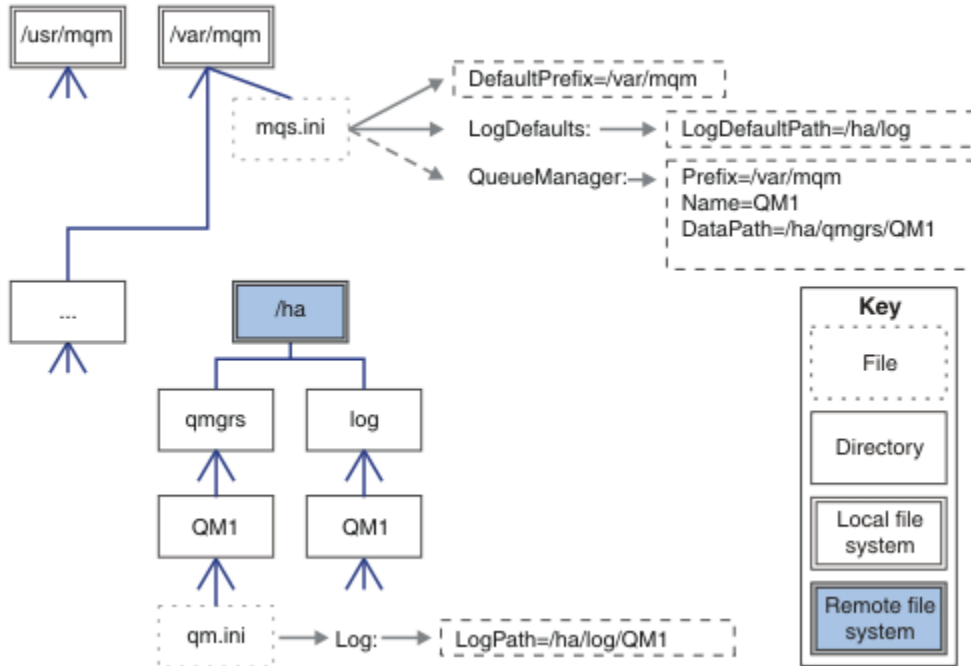


그림 39. 이름 지정된 qmgrs 및 log 디렉토리 공유

모두 공유

122 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 콘텐츠는 여러 서버의 큐 관리자 간에 공유됩니다.

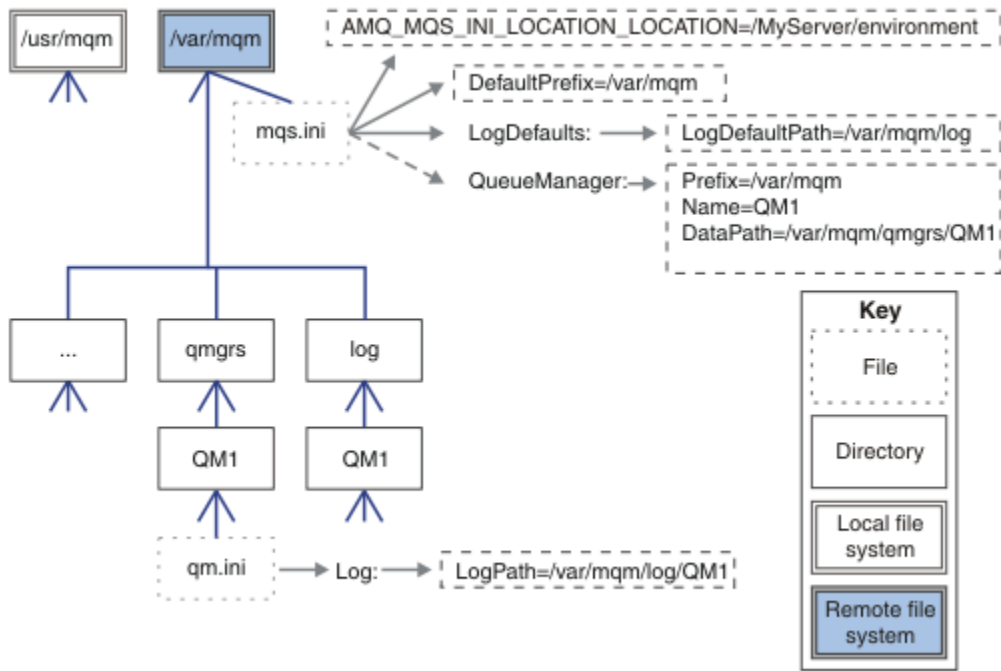


그림 40. 모두 공유

다중 인스턴스 큐 관리자에는 이를 사용할 수 없음을 유의하십시오. 다중 인스턴스 큐 관리자의 각 호스트는 세마 포어 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

C:\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 위치](#)를 참조하십시오.

사용 중인 설치 디렉토리 및 데이터 디렉토리를 알고 싶으면 `dspmqrver` 명령을 실행하십시오.

설치 디렉토리는 **InstPath** 필드에 나열되고 데이터 디렉토리는 **DataPath** 필드에 나열됩니다.

`dspmqrver` 명령을 실행하면 다음과 같은 정보가 표시됩니다.

```
>dspmqrver
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. `mq.s.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. `mq.s.ini` 파일에 저장된 큐 관리자 구성 정보는 `Name`, `Prefix`, `Directory`, `DataPath`로 감소합니다.

Windows 디렉토리 콘텐츠

IBM MQ 디렉토리의 위치 및 콘텐츠를 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
- 라이선스 파일

이 파일은 [123 페이지의 표 16](#)에 설명되어 있습니다.

2. 부분 큐 관리자에 고유하지 않은 잠재적으로 수정 가능한 파일 및 디렉토리. 이 파일과 디렉토리는 [124 페이지의 표 17](#)에 설명되어 있습니다.
3. 서버의 각 큐 관리자에 고유한 파일 및 디렉토리. 이 파일과 디렉토리는 [124 페이지의 표 18](#)에 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. <code>FilePath</code> 디렉토리에 있는 디렉토리 및 파일	
파일 경로	콘텐츠
<code>FilePath\bin</code>	명령 및 DLL
<code>FilePath\bin64</code>	명령 및 DLL(64비트)
<code>FilePath\conv</code>	데이터 변환 테이블
<code>FilePath\doc</code>	마법사 도움말 파일
<code>FilePath\MQExplorer</code>	탐색기 및 탐색기 도움말 Eclipse 플러그인
<code>FilePath\gskit8</code>	글로벌 보안 키
<code>FilePath\java</code>	JRE를 포함한, Java 자원
<code>FilePath\licenses</code>	라이선스 정보
<code>FilePath\Non_IBM_License</code>	라이선스 정보
<code>FilePath\properties</code>	내부적으로 사용됨
<code>FilePath\Tivoli</code>	
<code>FilePath\tools</code>	개발 자원 및 샘플
<code>FilePath\web</code>	편집 불가능한 파일에 대한 IBM MQ Console 및 REST API 설치 컴포넌트 파일 구조 에서 설명합니다.

표 16. <i>FilePath</i> 디렉토리에 있는 디렉토리 및 파일 (계속)	
파일 경로	컨텐츠
<i>FilePath</i> \Uninst	내부적으로 사용됨
<i>FilePath</i> \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. <i>DefaultPrefix</i> 디렉토리에 있는 디렉토리 및 파일	
파일 경로	컨텐츠
<i>DefaultPrefix</i> \config	내부적으로 사용됨
<i>DefaultPrefix</i> \conv	ccsid_part2.tbl 및 ccsid.tbl data 변환 제어 파일(데이터 변환에 설명됨)
<i>DefaultPrefix</i> \errors	큐 관리자에 해당되지 않은 오류 로그 AMQERR nn.LOG
<i>DefaultPrefix</i> \exits	채널 엑시트 프로그램
<i>DefaultPrefix</i> \exits64	채널 엑시트 프로그램(64비트)
<i>DefaultPrefix</i> \ipc	사용되지 않음
<i>DefaultPrefix</i> \qmgrs	124 페이지의 표 18에 설명됨
<i>DefaultPrefix</i> \trace	추적 파일
<i>DefaultPrefix</i> \web	사용자 편집 가능 파일에 대한 IBM MQ Console 및 REST API 설치 컴포넌트 파일 구조에서 설명합니다.
<i>DefaultPrefix</i> \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. <i>DataPath</i> 및 <i>Prefix</i> \qmgrs\QmgrName 디렉토리에 있는 디렉토리 및 파일	
파일 경로	컨텐츠
<i>DataPath</i> \@ipcc	클라이언트 연결 테이블, AMQCLCHL.TAB의 기본 위치
<i>DataPath</i> \authinfo	내부적으로 사용됨
<i>DataPath</i> \channel	
<i>DataPath</i> \clntconn	
<i>DataPath</i> \errors	오류 로그, AMQERR nn.LOG

표 18. <i>DataPath</i> 및 <i>Prefix\qmgrs\QmgrName</i> 디렉토리에 있는 디렉토리 및 파일 (계속)	
파일 경로	컨텐츠
<i>DataPath\listener</i>	내부적으로 사용됨
<i>DataPath\namelist</i>	
<i>DataPath\plugcomp</i>	
<i>DataPath\procdef</i>	
<i>DataPath\qmanager</i>	
<i>DataPath\queues</i>	
<i>DataPath\services</i>	
<i>DataPath\ssl</i>	
<i>DataPath\startprm</i>	
<i>DataPath\topic</i>	
<i>DataPath\active</i>	
<i>DataPath\active.dat</i>	
<i>DataPath\amqalchk.fil</i>	
<i>DataPath\master</i>	
<i>DataPath\master.dat</i>	
<i>DataPath\qm.ini</i>	큐 관리자 구성
<i>DataPath\qmstatus.ini</i>	큐 관리자 상태
<i>DataPath\userdata</i>	애플리케이션의 지속적 상태를 저장하는 데 사용할 수 있습니다.
<i>Prefix\qmgrs\QmgrName</i>	내부적으로 사용됨
<i>Prefix\qmgrs\@SYSTEM</i>	사용되지 않음
<i>Prefix\qmgrs\@SYSTEM\errors</i>	
<i>DataPath\autocfg</i>	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 i의 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

'-- qmqm

```
'-- doc
'-- inc
'-- lib
'-- samp
'-- licenses
'-- LicenseDoc
'-- 5724H72_V8R0M0
```

/QIBM/ProdData/mqm

이 디렉토리 아래의 서브디렉토리는 모든 제품 데이터(예: C++ 클래스, 추적 형식 파일, 라이선스 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

라이선스 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 라이선스 계약 파일을 저장합니다.

/QIBM/ProdData/LicenseDoc/5724H72_V8R0M0

라이선스 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 mq.s.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 xxxxx은(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 올바르게 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 관리](#)를 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 32비트 프로세스에서는 이 한계가 발생하지 않지만 메모리 요구사항이 높은 애플리케이션은 이유 코드 2102: MQRC_RESOURCE_PROBLEM으로 IBM MQ에 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 MQ에만 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 MQ에만 적용됩니다.

백그라운드에서 프로세스를 실행하는 경우 셸 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: **runmq1sr**)가 연관된 큐 관리자와 같은 *nice* 값을 가지는 것은 좋은 사례입니다. 셸이 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 -fl" 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 z/OS에서 IBM MQ 환경 계획

IBM MQ 환경을 계획할 때에는 데이터 세트, 페이지 세트, Db2, 커플링 기능, 로깅 및 백업 기능 필요에 대한 자원 요구사항을 고려해야 합니다. 이 주제를 사용하여 IBM MQ가 실행하는 환경을 계획하십시오.

IBM MQ 아키텍처를 계획하기 전에 기본 IBM MQ for z/OS 개념을 숙지하십시오([IBM MQ for z/OS 개념의 주제 참조](#)).

큐 관리자를 계획할 때 조직에서 다른 사용자와 함께 작업해야 합니다. 변경 제어 프로시저는 시간이 오래 걸릴 수 있기 때문에, 작업자들을 일찍 참여시키는 것이 좋습니다. 작업자들은 IBM MQ for z/OS를 구성하는 데 필요한 매개변수를 알려줄 수도 있습니다.

예를 들어 다음을 수행해야 할 수 있습니다.

- 스토리지 관리자는 큐 관리자 데이터 세트의 상위 레벨 규정자를 판별하고 큐 관리자 데이터 세트에 충분한 공간을 할당합니다.
- z/OS 시스템 프로그래머는 IBM MQ 서브시스템을 z/OS에 정의하고 APF는 IBM MQ for z/OS 라이브러리에 권한을 부여합니다.
- 네트워크 관리자는 IBM MQ for z/OS에 사용할 TCP/IP 스택 및 포트를 결정합니다.
- 보안 관리자는 큐 관리자 데이터 세트, IBM MQ for z/OS 자원에 대한 보안 프로파일 및 TLS 인증서에 대한 액세스를 설정합니다.
- Db2 관리자는 큐 공유 그룹을 구성할 때 Db2 테이블을 설정할 수 있습니다.

관련 개념

[IBM MQ 기술 개요](#)

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.

[z/OS 구성](#)

[IBM MQ for z/OS 관리](#)

z/OS 큐 관리자에 대한 계획

큐 관리자를 설정하는 경우 사용자의 계획이 큐 관리자의 확대를 허용해야 큐 관리자가 사용자 엔터프라이즈의 요구사항에 충족할 수 있습니다.

큐 관리자를 구성하는 가장 좋은 방법은 다음 단계와 같습니다.

1. 기본 큐 관리자 구성
2. 큐 관리자 간의 통신 및 원격 클라이언트 애플리케이션 통신을 수행하는 채널 시작기의 구성
3. 메시지를 암호화하고 보호하려면 [Advanced Message Security](#)를 구성하십시오.
4. IBM MQ를 통한 파일 전송을 사용하려면 [z/OS용 Managed File Transfer](#)를 구성하십시오.
5. 관리 또는 메시징 REST API나 IBM MQ Console을 사용하여 웹 브라우저에서 IBM MQ를 관리하려면 [mqweb](#) 서버를 구성하십시오.

일부 엔터프라이즈의 환경에는 수십만 개의 큐 관리자가 있습니다. 지금 및 5년 후 IBM MQ 네트워크를 고려해야 합니다.

z/OS에서 일부 큐 관리자는 초당 수천 개의 메시지를 처리하고 초당 100MB 넘게 로그합니다. 아주 높은 볼륨을 예상한다면 둘 이상의 큐 관리자를 고려해야 합니다.

z/OS에서, IBM MQ는 메시지가 커플링 기능에 저장되는 큐 공유 그룹(QSG)의 일부로 실행될 수 있으며 큐 공유 그룹의 큐 관리자가 메시지에 액세스할 수 있습니다. 큐 공유 그룹에서 실행하려면 필요한 큐 관리자의 수를 고려해야 합니다. 일반적으로 각 LPAR에 하나의 큐 관리자가 있습니다. 또한 하나의 큐 관리자가 정기적으로 CF 구조를 백업하도록 할 수 있습니다.

구성에 대한 일부 변경은 새 큐 정의와 같이 쉽게 수행할 수 있습니다. 로그 작성 및 페이지 크기 설정과 같은 일부 변경은 더 어려우며 큐 관리자나 큐 공유 그룹 이름 등의 일부 구성은 변경할 수 없습니다.

성능 및 성능 조정 정보는 [MP16 성능 SupportPac](#)에서 사용 가능합니다.

이름 지정 규칙

큐 관리자 데이터 세트에 대해 이름 지정 규칙이 있어야 합니다.

많은 엔터프라이즈에서 로드 라이브러리 이름 등에 릴리스 번호를 사용합니다. IBM MQ의 새 버전으로 마이그레이션할 때 CICS®, Batch 및 IMS JCL을 변경할 필요가 없도록 현재 사용 중인 버전 (예: MQM.V930.SCSQAUTH)을 가리키는 MQM.SCSQAUTH의 별명을 사용하는 것을 고려할 수 있습니다.

z/OS UNIX System Services의 기호 링크를 사용하여 현재 사용 중인 IBM MQ의 버전을 위한 설치 디렉토리를 참조할 수 있습니다.

큐 관리자에서 사용하는 데이터 세트(로그, 페이지 세트, JCL 라이브러리)에는 보안 프로파일 작성을 간소화하기 위한 이름 지정 규칙, 데이터 세트를 디스크에 저장할 위치를 제어하는 SMS 스토리지 클래스에 데이터 세트의 맵핑 그리고 데이터 세트의 속성이 필요합니다.

참고로 IBM MQ 버전을 페이지 세트나 로그 이름에 넣는 것은 좋은 생각이 아닙니다. 언젠가 새 버전으로 마이그레이션할 수 있고 그러면 데이터 세트가 "잘못된" 이름을 가지게 됩니다.

애플리케이션

비즈니스 애플리케이션을 이해하고 IBM MQ를 구성하는 최적의 방법을 알아야 합니다. 예를 들어 애플리케이션에 복구 및 반복 기능을 제공하는 논리가 있는 경우 비지속 메시지로 충분할 수 있습니다. IBM MQ가 복구를 처리하기를 원하면 지속 메시지를 사용하고 동기점에 메시지를 넣고 가져와야 합니다.

여러 비즈니스 트랜잭션에서 큐를 분리해야 합니다. 한 비즈니스 애플리케이션의 큐가 가득 차면 이것이 다른 비즈니스 애플리케이션에 영향을 주지 않기를 원할 것입니다. 가능하면 다른 페이지 세트 및 버퍼 풀 또는 구조로 큐를 분리하십시오.

메시지 프로파일을 이해해야 합니다. 많은 애플리케이션에서 큐에 몇 개의 메시지만 보유합니다. 다른 애플리케이션은 하루 동안 큐를 모아놓고 야간에 처리할 수 있습니다. 일반적으로 몇 개의 메시지만 보유하는 큐는 문제가 있는 경우 많은 시간 동안 메시지를 보유해야 하며 메시지가 처리되지 않습니다. CF 구조와 페이지 세트를 크기 조정하여 예상되는 최대 용량을 수용해야 합니다.

사후 구성

큐 관리자(및 컴포넌트)를 구성한 후 다음을 계획해야 합니다.

- 페이지 세트 백업.
- 오브젝트 정의 백업.
- 모든 CF 구조 백업 자동화.
- 문제점이 감지될 때 조치를 취하고 IBM MQ 메시지 모니터링.
- IBM MQ 통계 데이터 수집
- 가상 스토리지와 같은 자원 사용량 및 시간당 기록된 데이터 양을 모니터링. 이를 통해 자원 사용량이 증가하고 있는지 및 새 큐 관리자 설정과 같이 조치를 취해야 하는지를 알 수 있습니다.

z/OS의 스토리지 및 성능 요구사항 계획

IBM MQ 시스템에 맞는 실제적이고 도달 가능한 스토리지 및 성능 목표를 설정해야 합니다. 이 주제를 사용하여 성능 및 스토리지에 영향을 미치는 요인을 이해하십시오.

이 주제는 IBM MQ for z/OS의 스토리지 및 성능 요구사항에 대한 정보를 포함합니다. 포함된 절은 다음과 같습니다.

- [IBM MQ에 대한 z/OS 성능 옵션](#)
- [z/OS 워크로드 관리 중요도 및 속도 목표 판별](#)
- [132 페이지의 『라이브러리 스토리지』](#)
- [132 페이지의 『시스템 LX 사용』](#)
- [133 페이지의 『스토리지 구성』](#)
- [138 페이지의 『디스크 스토리지』](#)

자세한 정보는 [138 페이지의 『스토리지 및 성능 요구사항에 대한 자세한 정보 참조 위치』](#)의 내용을 참조하십시오.

IBM MQ에 대한 z/OS 성능 옵션

워크로드 관리를 통해 성능 목표를 정의하고 각 목표에 비즈니스 중요도를 지정합니다. 사용자는 비즈니스 조건에서 유효한 목표를 정의하고 시스템은 해당 목표를 달성하기 위해 작업에 제공해야 하는 프로세스 및 스토리지와 같은 자원의 양을 결정합니다. 워크로드 관리는 제공된 목표에 기초하여 디스패치 우선순위를 제어합니다. 워크로드 관리는 지정된 목표를 달성하기 위해 필요한 대로 우선순위를 올리거나 내립니다. 따라서 시스템의 모든 부분에 대한 정확한 우선순위를 상세히 조정하지 않아도 되며, 대신 비즈니스 목표에 초점을 맞출 수 있습니다.

다음과 같은 세 가지 종류의 목표가 있습니다.

응답 시간

작업을 빠르게 처리하려는 정도

실행 속도

프로세서, 스토리지, I/O 액세스, 큐 지연의 지연 없이 준비가 되었을 때 작업을 실행해야 하는 속도

임의 지정

성능 목표가 없이 우선순위가 낮은 작업의 범주

응답 시간 목표는 일반 사용자 애플리케이션에 적절합니다. 예를 들어, CICS 사용자는 워크로드 목표를 응답 시간 목표로 설정할 수 있습니다. IBM MQ 주소 공간은 속도 목표가 더 적절합니다. 큐 관리자에서 수행한 소량의 작업이 이 속도 목표로 계산되지만 이 작업은 성능에 중대한 영향을 미칩니다. 큐 관리자가 수행한 대부분의 작업은 일반 사용자 애플리케이션의 성능 목표로 계산됩니다. 채널 시작기 주소 공간에서 수행한 대부분의 작업은 자체 속도 목표로 계산됩니다. 채널 시작기가 수행하는 IBM MQ 메시지의 수신 및 송신은 일반적으로 이 메시지를 사용하는 비즈니스 애플리케이션의 성능에 중요합니다.

z/OS 워크로드 관리 중요도 및 속도 목표 판별

자세한 정보는 [133 페이지의 『z/OS 워크로드 관리 중요도 판별』](#)의 내용을 참조하십시오.

라이브러리 스토리지

제품 라이브러리의 디스크 스토리지를 할당해야 합니다. 정확한 수치는 구성에 따라 다르며 대상 및 분배 라이브러리와 SMP/E 라이브러리를 모두 포함해야 합니다.

IBM MQ for z/OS에서 사용하는 대상 라이브러리는 PDSE 형식을 사용합니다. PDSE 대상 라이브러리가 SYSPLEX 외부에서 공유되지 않는지 확인하십시오. 필수 라이브러리 및 크기와 필수 형식에 대한 자세한 정보는 프로그램 디렉토리를 참조하십시오. 프로그램 디렉토리에 대한 다운로드 링크는 [IBM MQ for z/OS 프로그램 디렉토리 PDF 파일](#)을 참조하십시오.

시스템 LX 사용

정의된 각 IBM MQ 서브시스템은 IPL 시 시스템 연계 색인(LX)을 하나 예약하고 큐 관리자가 시작될 때 많은 비시스템 연계 색인을 예약합니다. 시스템 연계 색인은 큐 관리자가 중지된 후 재시작될 때 다시 사용됩니다. 마찬가지로 분산 큐잉은 하나의 비시스템 연계 색인을 예약합니다. 만일 z/OS 시스템에 부적합한 시스템 LX가 정의된 경우에는 예약된 시스템 LX를 고려해야 합니다.

필요한 경우 시스템 LXs의 수는 SYS1.PARMLIB 멤버 IEASYSxx에서 NSYSLX 매개변수를 설정하여 증가될 수 있습니다.

z/OS z/OS 워크로드 관리 중요도 판별

서비스 정의를 통한 워크로드 관리 및 목표 정의에 대한 전체 정보는 를 참조하십시오.z/OS 제품 문서.

이 주제에서는 시스템의 다른 중요한 작업과 관련하여 z/OS 워크로드 관리 중요도 및 속도 목표를 설정하는 방법을 제안합니다. 자세한 정보는 *z/OS MVS 계획: 워크로드 관리* 를 참조하십시오.

큐 관리자 주소 공간은 서브시스템 서비스를 제공하므로 높은 우선순위로 정의해야 합니다. 채널 시작기는 애플리케이션 주소 공간이지만 일반적으로 리포트 큐 관리자로 송신되는 메시지가 지연되지 않도록 우선순위가 높습니다. AMS(Advanced Message Security)도 서브시스템 서비스를 제공하며 높은 우선순위로 정의되어야 합니다.

다음 서비스 클래스를 사용하십시오.

기본 **SYSSTC** 서비스 클래스

- VTAM 및 TCP/IP 주소 공간
- IRLM 주소 공간(IRLMPROC)

참고: VTAM, TCP/IP, IRLM 주소 공간은 모든 DBMS 주소 공간, 첨부된 주소 공간, 종속 주소 공간보다 디스패치 우선순위가 높아야 합니다. 워크로드 관리가 VTAM, TCP/IP 또는 IRLM의 우선순위를 DBMS 주소 공간의 우선순위 이하로 줄이지 못하게 하십시오.

다음에 대해 정의한 이름의 서비스 클래스(예를 들어, **PRODREGN**)에 대한 높은 속도 목표 및 중요도 **1**:

- IBM MQ 큐 관리자, 채널 시작기 및 AMS 주소 공간
- Db2(Db2 설정 스토어드 프로시저 주소 공간을 제외한, 모든 주소 공간)
- CICS(모든 리전 유형)
- IMS(BMP를 제외한 모든 리전 유형)

높은 속도 목표는 이러한 모든 주소 공간에서 시동 및 재시작이 가능한 한 빨리 수행되도록 하려는 경우에 바람직합니다.

CICS 및 IMS 리전에 대한 속도 목표는 시동 및 재시작 중에만 중요합니다. 트랜잭션이 실행하기 시작하면 워크로드 관리는 CICS 또는 IMS 속도 목표를 무시하고 리전에서 실행 중인 트랜잭션의 응답 시간 목표를 기반으로 우선순위를 지정합니다. 이러한 트랜잭션 목표는 구현하는 비즈니스 애플리케이션의 상대적 우선순위를 반영해야 합니다. 일반적으로 이러한 목표의 중요도 값은 2입니다. IBM MQ를 사용하는 모든 배치 애플리케이션도 유사하게 구현하는 비즈니스 애플리케이션의 상대적 우선순위를 반영하는 속도 목표 및 중요도를 가져야 합니다. 대개 중요도 및 속도 목표는 PRODREGN의 목표보다 낮습니다.

z/OS 스토리지 구성

64비트주소 공간에는 2GB 주소를 표시하는 "막대" 라는 가상 라인이 있습니다. 막대는 "막대 아래"라고 하는 2GB 주소 아래의 스토리지를 "막대 위"라고 하는 2GB 주소 위의 스토리지로부터 분리합니다. 막대 아래의 스토리지는 31비트 주소 지정 기능을 사용하고 막대 위의 스토리지는 64비트 주소 지정 기능을 사용합니다.

JCL REGION 매개변수를 사용하여 31비트스토리지의 한계를 지정하고 MEMLIMIT 매개변수를 사용하여 64비트스토리지의 한계를 지정할 수 있습니다. 이렇게 지정된 값은 z/OS 엑시트로 대체할 수 있습니다.

제안된 스토리지 구성

다음 표에서는 큐 관리자, 채널 시작기 및 AMS 주소 공간에 대해 제안된 **REGION** 및 **MEMLIMIT** 값을 표시합니다. 이 제안사항은 시작점으로 사용되어야 하며 다음의 정보를 사용하여 조정되어야 합니다.

- 134 페이지의 『큐 관리자 스토리지 구성』
- 136 페이지의 『IBM MQ 9.3 의 채널 시작기 스토리지 구성』
- **V 9.3.1** 137 페이지의 『IBM MQ 9.3.1 의 채널 시작기 스토리지 구성』

표 19. REGION 및 MEMLIMIT에 대해 제안된 정의	
주소 공간	스토리지 구성
큐 관리자	REGION=0M, MEMLIMIT=3G
IBM MQ 9.3.0의 채널 시작기	REGION=0M, MEMLIMIT=256M
V9.3.1 IBM MQ 9.3.1의 채널 시작기	REGION=0M, MEMLIMIT=2G
AMS 주소 공간	REGION=0M

MEMLIMIT 및 REGION 크기 관리

기타 메커니즘 (예: SYS1.PARMLIB의 SMFPRMxx 멤버에 있는 **MEMLIMIT** 매개변수 또는 IEFUSI 엑시트)은 설치 시 z/OS 주소 공간에 대한 막대 위의 기본 가상 스토리지 양을 제공하기 위해 사용될 수 있습니다. 막대 위의 스토리지 제한에 대한 전체 세부사항은 [막대 위의 메모리 관리](#)를 참조하십시오.

z/OS 큐 관리자 스토리지 구성

큐 관리자 주소 공간은 IBM MQ 설치에서 64비트스토리지의 주요 사용자일 수 있습니다. 큐 관리자에 대한 각 연결에서는 다음 텍스트에 설명된 대로 공용 스토리지를 할당해야 합니다. 64비트스토리지 외에도 큐 관리자 JCL에서 REGION=0M을 지정하여 큐 관리자가 사용 가능한 모든 31비트스토리지를 사용하도록 허용해야 합니다.

공용 스토리지

각 IBM MQ for z/OS 서브시스템에는 다음과 같은 대략적인 스토리지 요구사항이 있습니다.

- CSA 4KB
- ECSA 800KB+ CSQ6SYSP 시스템 매개변수 매크로의 **TRACTBL** 매개변수에 지정된 추적 테이블의 크기입니다. 자세한 정보는 CSQ6SYSP 사용을 참조하십시오.

또한 큐 관리자에 대한 각 동시 논리 연결에는 약 5KB의 ECSA가 필요합니다. 태스크가 종료되면 기타 IBM MQ 태스크가 이 스토리지를 재사용할 수 있습니다.

IBM MQ는 큐 관리자가 종료될 때까지 스토리지를 해제하지 않으므로 최대 동시 연결 수에 5KB를 곱하여 필요한 최대 ECSA 양을 계산할 수 있습니다. 동시 논리 연결의 수는 다음 수의 합계입니다.

- IBM MQ에 연결되어 있지만 연결이 끊어지지 않은 Batch, TSO, z/OS UNIX System Services, IMS 및 Db2 SPAS (Stored Procedure Address Space) 리전의 태스크 (TCB)입니다.
- IBM MQ 요청을 발행했지만 종료되지 않은 CICS 트랜잭션
- 작성되었지만(바인딩 연결용) 영구 삭제되거나 가비지 콜렉션되지 않은 JMS 연결, 세션, TopicSession 또는 QueueSession.
- 활성 IBM MQ 채널

ACELIM 구성 매개변수를 사용하여 큐 관리자에 대한 논리 연결에서 사용되는 공통 스토리지에 대한 한계를 설정할 수 있습니다. **ACELIM** 제어는 주로 Db2 스토어드 프로시저로 인해 IBM MQ 큐에서 조작성이 발생하는 사이트와 관련이 있습니다.

스토어드 프로시저에서 구동되는 경우 각 IBM MQ 조작성으로 인해 큐 관리자에 대한 새 논리 연결이 발생할 수 있습니다. 예를 들어, 테이블 로드나 대용량 Db2 작업 단위로 인해 공용 스토리지가 과도하게 요구될 수 있습니다.

ACELIM는 시스템의 연결 수를 제한하여 공통 스토리지 사용을 제한하고 z/OS 시스템을 보호하기 위한 것입니다. 과도한 양의 ECSA 스토리지를 사용하는 것으로 식별된 큐 관리자에서만 **ACELIM**를 설정해야 합니다. 자세한 정보는 CSQ6SYSP 사용의 **ACELIM** 절을 참조하십시오.

ACELIM의 값을 설정하려면 먼저 **ACELIM** 값으로 제어되는 서브풀에 현재 있는 스토리지의 양을 판별하십시오. 이 정보는 통계 CLASS(3) 추적에 의해 생성되는 SMF 115 하위 유형 5 레코드에 있습니다.

IBM MQ SMF 데이터는 SupportPac MP1B를 사용하여 형식화될 수 있습니다. **ACELIM**에 의해 제어되는 서브 풀에서 사용 중인 바이트 수는 *ACE/PEB*라는 행의 STGPOOL DD에 표시됩니다.

SMF 115 통계 레코드에 대한 자세한 정보는 [IBM MQ for z/OS 성능 통계 해석](#)을 참조하십시오.

확장 및 워크로드 증가용 공간을 제공하기 위해 충분한 여백으로 정상 값을 늘리십시오. 새 값을 1024로 나누어 **ACELIM** 구성에서 사용할 최대 스토리지 크기 (KB)를 산출하십시오.

사설 스토리지

큐 관리자 주소 공간은 많은 내부 제어 블록에 대해 64비트스토리지를 사용합니다. 큐 관리자 JCL의 **MEMLIMIT** 매개변수는 사용 가능한 최대 64비트스토리지 양을 정의합니다. 3GB의 스토리지 (**MEMLIMIT=3G**)는 사용하여야 하는 최소값이지만 구성에 따라 상당히 더 필요할 수 있습니다.

잠재적인 문제점을 방지하려면 **MEMLIMIT=NOLIMIT** 대신 특정 **MEMLIMIT** 값을 지정해야 합니다. **NOLIMIT** 또는 매우 큰 값을 지정하면 사용 가능한 모든 z/OS 가상 스토리지를 모두 사용할 수 있으며 이로 인해 시스템에서 페이지징이 발생합니다. **MEMLIMIT**의 값을 늘릴 때 사용할 수 있는 스토리지의 양에 대한 시스템 전체 한계가 있는 경우 z/OS 시스템 프로그래머와 새 설정에 대해 논의해야 합니다.

MEMLIMIT에 큰 값이 있는 경우 덤프에서 더 많은 데이터가 캡처되므로 덤프 데이터 세트의 크기를 늘려야 할 수 있습니다.

사용 중인 31비트 및 64비트 개인용 스토리지의 양과 나머지 여유 공간을 표시하는 **CSQY220I** 메시지에서 주소 공간 스토리지 사용량을 모니터링할 수 있습니다.

버퍼 풀

버퍼 풀은 큐 관리자 주소 공간에 있는 개인용 스토리지의 중요한 사용자입니다. 각 버퍼 풀 크기는 큐 관리자 초기화 시간에서 결정되며, 스토리지는 해당 버퍼 풀을 사용 중인 페이지 세트가 연결된 경우 버퍼 풀에 대해 할당됩니다. **LOCATION (ABOVE|BELOW)** 매개변수는 버퍼가 할당되는 위치를 지정하는 데 사용됩니다. **ALTER BUFFPOOL** 명령을 사용하여 버퍼 풀 크기를 동적으로 변경할 수 있습니다.

MEMLIMIT의 값을 계산할 때 **LOCATION(ABOVE)**로 구성된 경우 버퍼 풀 크기를 고려하는 것이 중요합니다. 다음과 같이 계산을 수행해야 합니다.

MEMLIMIT의 값을 2GB + **LOCATION(ABOVE)**로 구성된 버퍼 풀의 크기 (가장 가까운 GB로 반올림됨)로 계산하십시오. **MEMLIMIT**를 최소 3GB로 설정하고 버퍼 풀의 크기를 늘려야 하는 경우 필요에 따라 이 값을 늘리십시오.

예를 들어, **LOCATION(ABOVE)**로 구성된 세 개의 버퍼 풀의 경우, 버퍼 풀 1에는 10,000개의 버퍼가 있고 버퍼 풀 2와 3에는 각각 50,000개의 버퍼가 있습니다. 막대 위의 메모리 사용량은 $110,000$ (총 버퍼 수) * $4096 = 45,560,000$ 바이트 = 430MB입니다.

LOCATION에 관계없이 모든 버퍼 풀은 제어 구조에 64비트스토리지를 사용합니다. 버퍼 풀 수와 해당 풀의 버퍼 수가 증가되기 때문에 이는 중요해질 수 있습니다. 각 버퍼에는 약 200바이트의 추가 64비트스토리지가 필요합니다. 필요한 이전 구성의 경우: $200 * 110,000 = 22,000,000$ 바이트 = 21MB.

따라서 이 시나리오에서는 3GB를 **MEMLIMIT**에 사용할 수 있으며, 이는 증가 범위를 허용합니다. $21\text{MB} + 430\text{MB} + 2\text{GB}$ 는 3GB로 반올림됩니다.

일부 구성의 경우 실제 스토리지에 의해 영구적으로 백업된 버퍼가 있는 버퍼 풀을 사용할 경우 큰 성능상의 이점이 있을 수 있습니다. 버퍼 풀의 **PAGECLAS** 속성에 **FIXED4KB** 값을 지정하여 이를 수행할 수 있습니다. 그렇지만 LPAR에 사용 가능한 충분한 실제 스토리지가 있는 경우에만 이를 수행해야 합니다. 그렇지 않으면 다른 주소 공간이 영향을 받을 수 있습니다. **PAGECLAS**에 **FIXED4KB** 값을 사용해야 하는 시기에 대한 정보는 IBM MQ 지원 팩 MP16: IBM MQ for z/OS -용량 계획 및 튜닝을 참조하십시오.

MVS™ 페이지징이 있는 버퍼 풀을 너무 크게 만들면 반대로 성능에 영향을 미칠 수 있습니다. 메시지를 페이지 세트와 이동하는 IBM MQ를 사용하여 페이지징하지 않는 더 적은 버퍼 풀 사용을 고려할 수 있습니다.

IBM MQ 9.3.1 부터 **RECOVER CFSTRUCT** 명령은 64비트스토리지를 더 많이 사용합니다. 많은 경우에 사용 가능한 예미 64비트스토리지가 있어야 하므로 명령을 사용할 때 **MEMLIMIT**의 값을 늘릴 필요가 없습니다. 그러나 수백만 개 이상의 메시지를 포함하는 대형 구조 백업이 있는 경우에는 **RECOVER CFSTRUCT** 명령을 처리할 수 있는 모든 큐 관리자의 **MEMLIMIT** 를 500MB만큼 늘려야 합니다.

예를 들어, **MEMLIMIT=3G** 가 이미 있는 경우 **MEMLIMIT** 매개변수가 소수점을 허용하지 않으므로 **MEMLIMIT=4G** 사용을 고려해야 합니다.

공유 메시지 데이터 세트 (SMDS) 버퍼 및 **MEMLIMIT**

공유 메시지 데이터 세트를 사용하여 메시징 워크로드를 실행할 때 **DSBUFS** 및 **DSBLOCK** 속성을 조정하여 달성할 수 있는 두 가지 레벨의 최적화가 있습니다.

SMDS 버퍼에서 사용하는 막대 위 큐 관리자 스토리지의 양은 **DSBUFS x DSBLOCK**입니다. 이는 기본적으로 큐 관리자의 각 **CFLEVEL (5)** 구조에 대해 **100 x 256KB (25MB)** 가 사용됨을 의미합니다.

이 값이 너무 높지는 않지만 엔터프라이즈 또는 엔터프라이즈에 많은 **CFSTRUCT**가 있는 경우 일부는 버퍼 풀에 대해 높은 값인 **MEMLIMIT**를 할당할 수 있으며 때로는 깊은 색인 큐가 있으므로 전체적으로 막대 위의 스토리지가 부족할 수 있습니다.

IBM MQ 9.3의 채널 시작기 스토리지 구성

채널 시작기는 일반적으로 큐 관리자보다 훨씬 적은 64비트스토리지를 사용합니다. 64비트스토리지 외에도 큐 관리자 **JCL**에서 **REGION=0M** 을 지정하여 채널 시작기가 사용 가능한 모든 31비트스토리지를 사용하도록 허용해야 합니다.

공용 스토리지

채널 시작기는 일반적으로 최대 160KB의 **ECSA** 사용을 필요로 합니다.

31비트개인용 스토리지

채널 시작기에 사용 가능한 31비트스토리지는 **CHINIT**가 가질 수 있는 동시 연결 수를 제한합니다.

모든 채널은 채널 시작기 주소 공간에서 확장된 개인용 리전의 대략 170KB를 사용합니다. 메시지 채널 (예: 송신자 또는 수신자 채널) 의 경우, 32KB 보다 큰 메시지가 전송되면 스토리지가 메시지 크기만큼 늘어납니다. 이 증가된 스토리지는 다음 경우에 비워집니다.

- 송신 또는 클라이언트 채널에는 10개의 연속 메시지에 대해 현재 버퍼 크기의 절반 미만이 필요합니다.
- 하트비트가 송신되거나 수신됩니다.

스토리지가 **Language Environment** 내에서 재사용을 위해 사용 가능하게 되지만, **z/OS** 가상 스토리지 관리자에 의해 사용 가능한 것으로 표시되지 않습니다. 이는 채널의 수에 대한 상위 제한이 확장된 개인용 리전 크기에서 개별 사용자 시스템의 제한사항 및 도착 패턴과 메시지 크기에 의존한다는 것을 의미합니다.

채널 수의 상한은 확장된 리전 크기가 1.6GB를 초과할 가능성이 없기 때문에 다수의 시스템에서 대략 9000개로 설정되기 쉽습니다. 32KB보다 큰 메시지 크기의 사용은 시스템의 최대 채널 수를 줄입니다. 예를 들어, 100MB 길이의 메시지가 전송되고 확장된 리전 크기 1.6GB가 가정되면 최대 채널 수는 15개입니다.

채널 시작기 추적이 데이터 공간에 작성됩니다. 데이터베이스 스토리지의 크기는 **TRAXTBL** 매개변수로 제어합니다. **ALTER QMGR**을 참조하십시오.

64비트개인용 스토리지

채널 시작기 **JCL**의 **MEMLIMIT** 매개변수는 사용 가능한 최대 64비트스토리지 양을 정의합니다. 스토리지의 256MB, **MEMLIMIT=256M**은 사용해야 하는 최소값입니다. 구성에 따라 상당히 더 많은 구성이 필요할 수 있습니다.

잠재적인 문제점을 방지하려면 **MEMLIMIT = NOLIMIT**가 아닌 **SENSIBLE MEMLIMIT**값을 지정해야 합니다. **NOLIMIT** 또는 매우 큰 값을 지정하면 사용 가능한 모든 **z/OS** 가상 스토리지를 모두 사용하여 시스템에서 페이지할 수 있습니다. **MEMLIMIT**값을 늘릴 때 사용할 수 있는 스토리지의 양에 대한 시스템 전체 한계가 있는 경우에 대비하여 **z/OS** 시스템 프로그래머와 새 설정을 논의해야 합니다.

MEMLIMIT에 대한 큰 값이 있는 경우 덤프에서 더 많은 데이터가 캡처되므로 덤프 데이터 세트의 크기를 늘려야 할 수 있습니다.

채널 시작기에는 64비트저장영역의 단일 사용자인 SMF가 있습니다.

SMF

사용 가능한 경우 SMF 클래스 4계정 또는 통계에 64비트스토리지가 필요합니다. 최소 256MB 스토리지가 필요합니다. 사용 가능한 스토리지가 충분하지 않으면 채널 시작기가 CSQX124E 메시지를 발행하고 클래스 4계정 및 통계를 사용할 수 없습니다.

IBM MQ 9.3.1의 채널 시작기 스토리지 구성

채널 시작기는 일반적으로 큐 관리자보다 훨씬 적은 64비트스토리지를 사용합니다. 그러나 IBM MQ 9.3.1부터 사용량이 증가했습니다. 64비트스토리지 외에도 큐 관리자 JCL에서 REGION=OM을 지정하여 채널 시작기가 사용할 수 있는 모든 31비트스토리지를 사용하도록 허용해야 합니다.

공용 스토리지

채널 시작기는 일반적으로 최대 160KB의 ECSA 사용을 필요로 합니다.

31비트개인용 스토리지

채널 시작기에 사용 가능한 31비트스토리지는 CHINIT가 가질 수 있는 동시 연결 수를 제한합니다.

모든 채널은 채널 시작기 주소 공간에서 확장된 개인용 리전의 대략 170KB를 사용합니다. 메시지 채널 (예: 송신자 또는 수신자 채널)의 경우, 32KB보다 큰 메시지가 전송되면 스토리지가 메시지 크기만큼 증가합니다. 이 증가된 스토리지는 다음 경우에 비워집니다.

- 송신 또는 클라이언트 채널에는 10개의 연속 메시지에 대해 현재 버퍼 크기의 절반 미만이 필요합니다.
- 하트비트가 송신되거나 수신됩니다.

스토리지가 Language Environment내에서 재사용을 위해 사용 가능하게 되지만, z/OS 가상 스토리지 관리자에 의해 사용 가능한 것으로 표시되지 않습니다. 이는 채널의 수에 대한 상위 제한이 확장된 개인용 리전 크기에서 개별 사용자 시스템의 제한사항 및 도착 패턴과 메시지 크기에 의존한다는 것을 의미합니다.

채널 수의 상한은 확장된 리전 크기가 1.6GB를 초과할 가능성이 없기 때문에 다수의 시스템에서 대략 9000개로 설정되기 쉽습니다.

채널 시작기 추적이 데이터 공간에 작성됩니다. 데이터베이스 스토리지의 크기는 **TRAXTBL** 매개변수로 제어합니다. ALTER QMGR를 참조하십시오.

64비트개인용 스토리지

채널 시작기 JCL의 MEMLIMIT 매개변수는 사용 가능한 최대 64비트스토리지 양을 정의합니다. 2GB의 스토리지 (MEMLIMIT=2 GB)가 사용해야 하는 최소값입니다. 구성에 따라 상당히 더 많은 구성이 필요할 수 있습니다.

잠재적인 문제점을 방지하려면 MEMLIMIT = NOLIMIT가 아닌 SENSIBLE MEMLIMIT값을 지정해야 합니다. NOLIMIT 또는 매우 큰 값을 지정하면 사용 가능한 모든 z/OS 가상 스토리지를 모두 사용하여 시스템에서 페이징할 수 있습니다. MEMLIMIT값을 늘릴 때 사용할 수 있는 스토리지의 양에 대한 시스템 전체 한계가 있는 경우에 대비하여 z/OS 시스템 프로그래머와 새 설정을 논의해야 합니다.

MEMLIMIT에 대한 큰 값이 있는 경우 덤프에서 더 많은 데이터가 캡처되므로 덤프 데이터 세트의 크기를 늘려야 할 수 있습니다.

채널 시작기에는 두 명의 64비트저장영역 사용자 (SMF 및 서버 연결 채널)가 있습니다.

SMF

사용 가능한 경우 SMF 클래스 4계정 또는 통계에 64비트스토리지가 필요합니다. 최소 256MB 스토리지가 필요합니다. 사용 가능한 스토리지가 충분하지 않으면 채널 시작기가 CSQX124E 메시지를 발행하고 클래스 4계정 및 통계를 사용할 수 없습니다.

서버 연결 채널

IBM MQ 9.3.1 서버 연결 채널에서는 크기가 32KB보다 큰 메시지를 전송하는 경우 64비트스토리지에 메시지 버퍼를 할당합니다.

이러한 버퍼는 채널이 10개의 연속 메시지에 대해 현재 버퍼 크기의 절반 미만을 필요로 하거나 하트비트가 송신 또는 수신되는 경우에 해제됩니다.

MEMLIMIT값은 실행할 수 있는 동시 서버 연결 채널 수에 대한 상한을 설정합니다. MEMLIMIT=2G의 최소값을 사용하여 동일한 수의 채널이 IBM MQ 9.3.1의 이전 버전에서와 같이 실행될 수 있도록 하고 확장을 위한 일부 용량을 제공해야 합니다.

동시에 활성 상태인 서버 연결 채널의 최대 최대 수를 계산하여 MEMLIMIT에 대한 대략적인 값을 계산할 수 있으며, 해당 채널의 경우 전송할 것으로 예상되는 최대 메시지 크기를 계산할 수 있습니다. MEMLIMIT=2GB를 시작점으로 사용하고 반올림해야 합니다.

예를 들어, 최대 동시 서버 연결 채널 수를 2,000으로 설정하고 각 채널의 최대 메시지 크기를 1MB로 설정하면 서버 연결 채널은 64비트스토리지의 최대 2GB 미만을 사용합니다. 이는 2GB에 매우 근접하므로 MEMLIMIT=3G로 반올림해야 합니다.

디스크 스토리지

로그 데이터 세트, Db2 스토리지, 커플링 기능 스토리지, 페이지 데이터 세트의 디스크 스토리지 요구사항을 계획할 때 이 주제를 사용하십시오.

큐 관리자 데이터 세트를 둘 곳을 판별하려면 스토리지 관리자와 함께 작업하십시오. 예를 들어, 스토리지 관리자는 여러 다른 데이터 세트 유형에 대한 특정 DASD 볼륨 또는 SMS 스토리지 클래스, 데이터 클래스, 관리 클래스를 제공할 수 있습니다.

- 로그 데이터 세트는 DASD에 있어야 합니다. 이 로그는 작은 응답 시간에 I/O 활동이 많으며 백업할 필요가 없습니다.
- 아카이브 로그는 DASD 또는 테이프에 있을 수 있습니다. 이 로그는 작성된 후 백업에서 페이지 세트를 복구할 때와 같은 비정상 상황이 아니면 다시 읽지 않을 수 있습니다. 보유 날짜가 길어야 합니다.
- 페이지 세트는 낮거나 중간 정도의 활동이 있을 수 있으므로 정기적으로 백업해야 합니다. 사용량이 많은 시스템에서는 하루에 두 번 백업해야 합니다.
- BSDS 데이터 세트는 I/O 활동이 높지 않으므로 매일 백업해야 합니다.

모든 데이터 세트는 Db2에 사용되는 데이터 세트와 유사하므로 IBM MQ에 유사한 유지보수 프로시저를 사용할 수 있습니다.

데이터 스토리지를 계획하는 방법에 대한 자세한 내용은 다음 절을 참조하십시오.

• 로그 및 아카이브 스토리지

[153 페이지의 『아카이브 로그의 보관 기간』](#)에는 IBM MQ 시스템이 핸들링하는 메시지의 볼륨 및 활성 데이터 세트로 활성 로그가 오프로드되는 빈도에 따라, 활성 로그 및 아카이브 데이터 세트에 필요한 스토리지 양을 판별하는 방법이 설명되어 있습니다.

• Db2 스토리지

[169 페이지의 『Db2 스토리지』](#)에는 Db2가 IBM MQ 데이터에 대해 요구하는 스토리지 양을 판별하는 방법이 설명되어 있습니다.

• 커플링 기능 스토리지

[161 페이지의 『커플링 기능 자원 정의』](#)에는 커플링 기능 구조의 크기를 판별하는 방법이 설명되어 있습니다.

• 페이지 세트 및 메시지 스토리지

[139 페이지의 『페이지 세트 및 버퍼 풀 계획』](#)에는 애플리케이션이 교환하는 메시지의 크기, 이 메시지의 수, 메시지가 작성 또는 교환되는 비율에 따라 페이지 데이터 세트에 필요한 스토리지 양을 판별하는 방법이 설명되어 있습니다.

스토리지 및 성능 요구사항에 대한 자세한 정보 참조 위치

스토리지 및 성능 요구사항에 대한 자세한 정보를 찾기 위한 참조로 이 주제를 사용하십시오.

다음 소스에서 더 많은 정보를 찾을 수 있습니다.

표 20. 스토리지 요구사항에 대한 자세한 정보 참조 위치	
주제	찾을 위치
시스템 매개변수	CSQ6SYSP 사용 및 큐 관리자 사용자 정의
IBM MQ 설치에 필요한 스토리지	프로그램 디렉토리. 프로그램 디렉토리에 대한 다운로드 링크는 IBM MQ for z/OS 프로그램 디렉토리 PDF 파일 을 참조하십시오.
IEALIMIT 및 IEFUSI 엑시트	z/OS:MVS 설치 엑시트 문서에서 IEALIMIT 및 IEFUSI 를 참조하십시오.
최신 정보	IBM MQ SupportPac 웹 사이트 IBM MQ 및 기타 프로젝트 영역의 경우 SupportPacs .
워크로드 관리 및 서비스 정의를 통한 목표 정의	z/OS MVS 계획: 워크로드 관리

▶ z/OS 페이지 세트 및 버퍼 풀 계획

페이지 데이터 세트 및 버퍼 풀의 초기 수와 크기를 계획할 때 유용한 정보입니다.

이 주제에는 다음 절이 포함되어 있습니다.

- [139 페이지의 『페이지 세트 계획』](#)
 - [페이지 세트 사용](#)
 - [페이지 세트 수](#)
 - [페이지 세트 크기](#)
 - [z/OS 데이터 세트 암호화 계획](#)
- [140 페이지의 『페이지 세트의 크기 계산』](#)
 - [페이지 세트 0](#)
 - [페이지 세트 01 - 99](#)
 - [메시지의 스토리지 요구사항 계산](#)
- [142 페이지의 『동적 페이지 세트 확장 사용』](#)
- [143 페이지의 『버퍼 풀 정의』](#)

페이지 세트 계획

페이지 세트 사용

단기 메시지의 경우 페이지 세트에 사용되는 페이지가 일반적으로 거의 없고 시동 시, 체크포인트 중 또는 시스템 종료 시를 제외하면 데이터 세트에 대한 I/O가 아예 없거나 거의 없습니다.

장기 메시지의 경우에는 메시지를 포함한 페이지가 대개 디스크에 기록됩니다. 이 조작은 재시작 시간을 줄이기 위해 큐 관리자가 수행합니다.

단기 메시지를 다른 페이지 세트 및 다른 버퍼 풀에 배치하여 장기 메시지와 분리하십시오.

페이지 세트 수

여러 개의 큰 페이지 세트를 사용하면 필요한 페이지 세트가 거의 없어서 페이지 세트로 큐를 보다 간편하게 맵핑할 수 있다는 의미이므로 IBM MQ 관리자 역할이 더 수월할 수 있습니다.

여러 개의 더 작은 페이지 세트를 사용할 경우 많은 장점이 있습니다. 예를 들어, 백업 시간이 줄어들고 백업 및 재시작 중에 병렬로 I/O를 수행할 수 있습니다. 하지만 많은 수의 페이지 세트 중 하나에 각 큐를 맵핑해야 하는 IBM MQ 관리자의 역할에 상당한 성능 비용이 추가됨을 고려하십시오.

다음과 같이 최소 5개의 페이지 세트를 정의하십시오.

- 오브젝트 정의에 예약된 페이지 세트(페이지 세트 0)
- 시스템 관련 메시지에 대한 페이지 세트
- 성능 중심의 장기 메시지에 대한 페이지 세트
- 성능 중심의 단기 메시지에 대한 페이지 세트
- 다른 모든 메시지에 대한 페이지 세트

143 페이지의 『버퍼 풀 정의』에는 이러한 방식으로 페이지 세트에 메시지를 분산시키는 경우의 성능 이점이 설명됩니다.

페이지 세트의 크기

예상하는 최대 메시지 용량에 충분한 공간을 페이지 세트에 정의하십시오. 큐 서버 프로그램이 실행 중이 아니어서 메시지 빌드를 개발할 때와 같은 예상치 못한 최대 용량을 고려하십시오. 2차 범위로 페이지 세트를 할당하거나 동적 페이지 세트 확장을 사용으로 설정하여 이를 수행할 수 있습니다. 자세한 정보는 142 페이지의 『동적 페이지 세트 확장 사용』의 내용을 참조하십시오. 페이지 세트를 작게 만드는 것이 어려우므로 더 작은 페이지 세트를 할당하고 필요할 때마다 확장할 수 있게 하는 것이 더 낫습니다.

페이지 세트 크기를 계획할 때에는 비애플리케이션 메시지 데이터를 포함하여 생성될 수 있는 모든 메시지를 고려하십시오. 예를 들어, 애플리케이션이 요청한 트리거 메시지, 이벤트 메시지, 보고 메시지가 있습니다.

페이지 세트의 크기는 큰 페이지 세트가 복원 시간이 더 오래 걸리므로 백업에서 복원할 때 페이지 세트를 복구하는 데 걸리는 시간을 판별합니다.

참고: 페이지 세트의 복구도 백업을 생성한 이후에 기록된 로그 레코드를 큐 관리자가 처리하는 데 걸리는 시간에 따라 다릅니다. 이 기간은 백업 빈도에 따라 결정됩니다. 자세한 정보는 171 페이지의 『백업 및 복구 계획』의 내용을 참조하십시오.

참고: 4GB를 초과하는 페이지 세트의 경우 SMS 확장 주소 지정 기능을 사용해야 합니다.

z/OS 데이터 세트 암호화 계획

IBM MQ for z/OS 9.1.4 이상에서 실행 중인 큐 관리자의 페이지 세트에 z/OS 데이터 세트 암호화 기능을 적용할 수 있습니다.

이러한 페이지 세트는 EXTENDED 속성, 그리고 데이터가 AES 암호화되도록 보장하는 데이터 세트 키 레이블을 사용하여 할당해야 합니다.

데이터 세트 암호화를 사용하는 IBM MQ for z/OS의 저장 데이터에 대한 기밀성 섹션을 참조하십시오. 참조하십시오.

페이지 세트의 크기 계산

큐 관리자 오브젝트 정의(예: 큐 및 프로세스)의 경우 오브젝트가 고정 크기이고 영구적이므로 스토리지 요구사항 계산이 단순합니다. 하지만 메시지는 다음과 같은 이유로 계산이 더욱 복잡합니다.

- 메시지 크기가 다양합니다.
- 메시지가 일시적입니다.
- 검색된 메시지가 차지한 공간이 비동기 프로세스에 주기적으로 재이용됩니다.

필요에 따라 네트워크가 중지된 경우 메시지에 대한 추가 용량을 제공하는 4GB를 초과하는 큰 페이지 세트가 작성될 수 있습니다. 기존 페이지 세트를 수정할 수는 없습니다. 대신에 확장 주소 지정 기능 및 확장 형식 속성이 있는 새 페이지 세트를 작성해야 합니다. 새 페이지 세트는 이전 페이지 세트와 물리적 크기가 동일하고 이전 페이지 세트가 새 페이지 세트로 복사되어야 합니다. 역방향 마이그레이션이 필요한 경우 페이지 세트 0을 변경해서는 안 됩니다. 4GB 미만의 페이지 세트가 적절한 경우에는 조치가 필요하지 않습니다.

페이지 세트 0

페이지 세트 0은 오브젝트 정의에 대해 예약됩니다.

페이지 세트 0의 경우 필요한 스토리지는 다음과 같습니다.

```

(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)

```

이 값을 4096으로 나눠서 페이지 세트 데이터 세트에 대해 클러스터에 지정할 레코드 수를 판별하십시오.

공유 저장소에 저장된 오브젝트에 대해서는 허용할 필요가 없지만 페이지 세트 0에 저장 또는 복사된 오브젝트(GROUP 또는 QMGR 속성 지정의 오브젝트)에 대해 허용해야 합니다.

작성할 수 있는 총 오브젝트 수는 페이지 세트 0의 용량으로 제한됩니다. 정의할 수 있는 로컬 큐의 수는 524 287로 제한됩니다.

페이지 세트 01 - 99

페이지 세트 01 - 99의 경우 각 페이지 세트에 필요한 스토리지가 페이지 세트에 저장된 메시지의 수와 크기로 판별됩니다. (공유 큐의 메시지는 페이지 세트에 저장되지 않습니다.)

이 값을 4096으로 나눠서 페이지 세트 데이터 세트에 대해 클러스터에 지정할 레코드 수를 판별하십시오.

메시지의 스토리지 요구사항 계산

이 절은 페이지에 메시지가 저장되는 방식을 설명합니다. 이 방식을 이해하면 메시지에 대해 정의해야 하는 페이지 세트 스토리지 크기를 계산하는 데 도움이 될 수 있습니다. 페이지 세트의 모든 메시지에 필요한 대략적 공간을 계산하려면 해당 큐의 평균 메시지 크기 및 페이지 세트로 맵핑하는 모든 큐의 최대 큐 용량을 고려해야 합니다.

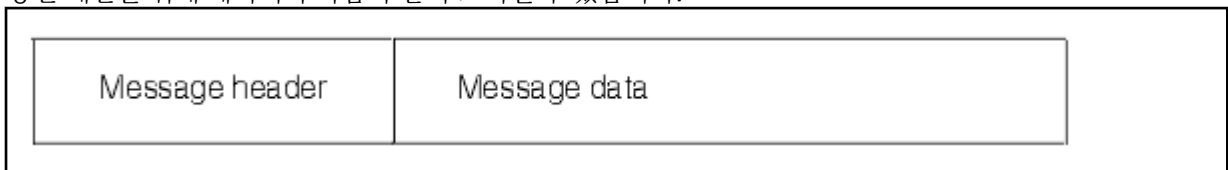
참고: 이 절에 제공된 제어 정보와 구조의 크기는 주요 릴리스에서 변경될 수 있습니다. 사용자 IBM MQ 릴리스 고유의 자세한 내용은 [SupportPac MP16 - z/OS 용 IBM MQ 용량 계획 및 튜닝 및 IBM MQ 제품군-성능 보고서](#)의 내용을 참조하십시오.

IBM MQ의 제어 이외의 이유(예: 통신 프로토콜의 문제점)로 메시지 "가져오기"가 지연될 수 있는 가능성을 허용해야 합니다. 이 경우 메시지의 "넣기" 속도가 "가져오기" 속도를 크게 초과할 수 있습니다. 이로 인해 페이지 세트에 저장되는 메시지 수가 크게 늘어나서 결과적으로 요구되는 스토리지 크기가 증가할 수 있습니다.

페이지 세트의 각 페이지 길이는 4096바이트입니다. 고정 헤더 정보를 허용할 경우 각 페이지의 메시지를 저장할 수 있는 공간은 4057바이트입니다.

각 메시지에 필요한 공간을 계산할 때 먼저 고려해야 할 사항은 메시지가 한 페이지에 맞는지(간단한 메시지) 또는 둘 이상의 페이지로 분할해야 하는지(긴 메시지) 여부입니다. 메시지를 이러한 방식으로 분할할 경우 공간 계산에 추가 제어 정보를 허용해야 합니다.

공간 계산을 위해 메시지가 다음과 같이 표시될 수 있습니다.



메시지 헤더 섹션은 메시지 디스크립터 및 기타 제어 정보를 포함하며 메시지 크기에 따라 크기가 다릅니다. 메시지 데이터 섹션은 모든 실제 메시지 데이터와 기타 헤더(예를 들어, 전송 큐 또는 IMS 브릿지 헤더)를 포함합니다.

페이지 세트 제어 정보에는 최소 두 페이지가 필요하며 이는 일반적으로 메시지에 필요한 총 공간의 1% 미만입니다.

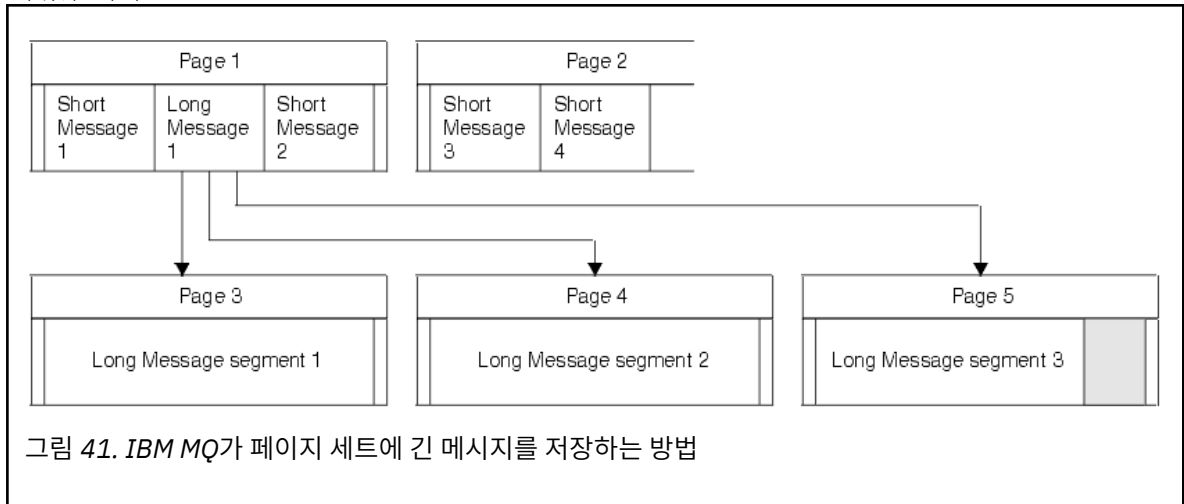
간단한 메시지

간단한 메시지는 한 페이지에 맞는 메시지로 정의됩니다.

작은 메시지는 각 페이지에 하나씩 저장됩니다.

긴 메시지

메시지 데이터 크기가 3596바이트보다 크고 4MB 이하이면 이 메시지는 긴 메시지로 분류됩니다. 긴 메시지로 표시될 경우 IBM MQ는 일련의 페이지에 메시지를 저장하고 간단한 메시지를 저장하는 것과 같은 방식으로 이러한 페이지를 가리키는 제어 정보를 저장합니다. 이는 [142 페이지의 그림 41](#)에 표시되어 있습니다.



매우 긴 메시지

매우 긴 메시지는 크기가 4MB를 초과하는 메시지입니다. 각 4MB에 1037개 페이지가 사용되도록 저장됩니다. 위에서 설명한 것처럼 나머지는 긴 메시지와 같은 방식으로 저장됩니다.

z/OS 동적 페이지 세트 확장 사용

큐 관리자가 실행되는 동안 페이지 세트를 동적으로 확장할 수 있습니다. 페이지 세트에는 123개의 확장이 있을 수 있으며 여러 디스크 볼륨에 분산될 수 있습니다.

페이지 세트가 확장될 때마다 새 데이터 세트 범위가 사용됩니다. 최대 범위 수에 도달할 때까지 또는 적합한 볼륨에 할당 가능한 스토리지가 더 이상 없을 때까지 큐 관리자는 필요에 따라 계속해서 페이지 세트를 확장합니다.

위의 이유 중 하나로 페이지 세트 확장이 실패하면 큐 관리자는 추가 확장 시도가 없음에 대해 페이지 세트에 표시합니다. 페이지 세트를 EXPAND(SYSTEM)으로 변경하여 이 표시를 재설정할 수 있습니다.

페이지 세트 확장은 페이지 세트에서 기존 공간의 90%가 할당될 때 다른 모든 페이지 세트 활동에 비동기식으로 발생합니다.

페이지 세트 확장 프로세스는 새로 할당된 범위를 형식화하여 큐 관리자에 사용 가능하게 합니다. 하지만 전체 범위가 형식화될 때까지는 사용 가능한 공간이 없습니다. 이는 큰 범위의 확장은 시간이 다소 걸릴 수 있으며 확장이 완료되기 전에 넣는 애플리케이션이 페이지 세트의 나머지 10%를 채우는 경우 '차단'할 수 있음을 의미합니다.

샘플 `thlqual.SCSQPROC(CSQ4PAGE)`는 2차 범위를 정의하는 방법을 보여줍니다.

새 범위의 크기를 제어하려면 DEFINE PSID 및 ALTER PSID 명령의 EXPAND 키워드의 다음 옵션 중 하나를 사용합니다.

- 사용자
- SYSTEM
- NONE

사용자

페이지 세트가 할당될 때 지정된 2차 범위 크기를 사용합니다. 값이 지정되지 않은 경우나 0 값이 지정된 경우 동적 페이지 세트 확장이 발생할 수 없습니다.

페이지 세트 확장은 페이지의 공간이 90% 사용될 때 발생하며 기타 페이지 세트 활동과 비동기식으로 수행됩니다.

이를 통해 한 번에 둘 이상의 단일 범위로 확장할 수 있습니다.

다음 예를 고려하십시오. 기본 범위가 100,000 페이지이고 보조 범위가 5000 페이지인 페이지 세트를 할당합니다. 9999개 페이지를 필요로 하는 메시지를 넣습니다. 페이지 세트에 이미 85,000 페이지가 사용 중인 경우 메시지를 쓰면 90% 가득 참 경계(90,000 페이지)가 초과됩니다. 이 시점에서, 기본 범위 100,000개 페이지에 추가 2차 범위가 할당되어 페이지 세트 크기는 105,000개 페이지가 됩니다. 메시지의 나머지 4999개 페이지는 계속해서 기록됩니다. 사용한 페이지 공간이 105,000개 페이지의 업데이트된 페이지 세트 크기의 90%인 94,500개 페이지에 도달하면 또 다른 5000개 페이지 범위가 할당되어 페이지 세트 크기는 110,000개 페이지가 됩니다. MQPUT의 끝에, 페이지 세트가 두 번 확장되고 94,500 페이지가 사용됩니다. 두 번째 페이지 세트 확장의 페이지는 할당되었지만 사용되지 않았습니다.

재시작 시 이전에 사용된 페이지 세트 보다 작은 데이터 세트로 대체된 경우 이전에 사용된 데이터 세트 크기에 도달할 때까지 확장됩니다. 오직 한 범위만 이 크기에 도달하면 됩니다.

SYSTEM

페이지 세트가 정의될 때 지정된 2차 범위 크기를 무시합니다. 대신에 큐 관리자가 현재 페이지 세트 크기의 약 10%인 값을 설정합니다. 값은 DASD의 가장 가까운 실린더로 반올림됩니다.

값이 지정되지 않은 경우나 0 값이 지정된 경우에도 동적 페이지 세트 확장이 여전히 발생할 수 있습니다. 큐 관리자가 현재 페이지 세트 크기의 약 10%인 값을 설정합니다. 새 값은 DASD의 특성에 따라 반올림됩니다.

페이지 세트 확장은 페이지 세트의 공간이 대략 90% 사용될 때 발생하며 기타 페이지 세트 활동과 비동기식으로 수행됩니다.

재시작 시 이전에 사용된 페이지 세트 보다 작은 데이터 세트로 대체된 경우 이전에 사용된 데이터 세트 크기에 도달할 때까지 확장됩니다.

NONE

추가적인 페이지 세트 확장이 수행되지 않습니다.

관련 참조

[ALTER PSID](#)

[DEFINE PSID](#)

[표시 사용법](#)

버퍼 풀 정의

이 주제를 사용하여 정의해야 하는 버퍼 풀의 수와 설정을 계획하십시오.

이 주제는 다음의 절로 구성되어 있습니다.

1. [143 페이지의 『정의할 버퍼 풀의 수 결정』](#)
2. [144 페이지의 『각 버퍼 풀에 대한 설정 결정』](#)
3. [145 페이지의 『예상한 로드 하의 버퍼 풀 성능 모니터』](#)
4. [145 페이지의 『버퍼 풀 특성 조정』](#)

정의할 버퍼 풀의 수 결정

처음에 4개의 버퍼 풀을 정의해야 합니다.

버퍼 풀 0

오브젝트 정의(페이지 세트 0의) 및 성능이 중요한 시스템 관련 메시지 큐(예: SYSTEM.CHANNEL.SYNCQ 큐와 SYSTEM.CLUSTER.COMMAND.QUEUE 및 SYSTEM.CLUSTER.REPOSITORY.QUEUE 큐)에 사용하십시오.

그러나 다수의 채널 또는 클러스터링이 사용되는 경우 버퍼 풀 특성 조정의 [145 페이지](#)의 『7』 번 항목을 고려하는 것이 중요합니다.

나머지 3개의 버퍼 풀은 사용자 메시지에 사용하십시오.

버퍼 풀 1

중요한 장기 메시지에 사용하십시오.

장기 메시지는 두 체크포인트보다 오래 시스템에 남아있는 메시지로, 체크포인트 시에 페이지 세트에 기록됩니다. 장기 메시지가 많은 경우 페이지 세트 I/O가 균등하게 분포될 수 있도록(버퍼 풀이 85% 차게 될 때마다 더 오래된 메시지가 DASD로 기록됨) 이 버퍼 풀이 상대적으로 작아야 합니다.

버퍼 풀이 너무 커서 버퍼 풀이 85%까지 차지 않으면 체크포인트 처리 시까지 페이지 세트 I/O가 지연됩니다. 이는 시스템 전반의 응답 시간에 영향을 미칠 수 있습니다.

몇 개의 장기 메시지만 예상할 경우 모든 메시지를 보유할 만큼 충분한 크기가 되도록 이 버퍼 풀을 정의하십시오.

버퍼 풀 2

성능이 중요한 단기 메시지에 사용하십시오.

일반적으로 몇 개의 버퍼를 사용한 버퍼 재사용 수준이 높습니다. 하지만 서버 애플리케이션이 실패할 때처럼 예상치 못한 메시지 누적을 허용할 정도의 버퍼 풀 크기여야 합니다.

버퍼 풀 3

다른 모든(일반적으로 성능이 중요한 문제가 아닌) 메시지에 사용하십시오.

데드-레터 큐, SYSTEM.COMMAND.* 큐 및 SYSTEM.ADMIN.* 큐와 같은 큐도 버퍼 풀 3에 맵핑될 수 있습니다.

가상 스토리지 제한조건이 있으며 더 작은 버퍼 풀이 필요한 경우 크기 축소를 위한 첫 번째 후보는 버퍼 풀 3입니다.

다음의 경우 추가 버퍼 풀을 정의해야 할 수 있습니다.

- 다양한 시점에 여러 다른 작동을 표시하기 때문에 특정 큐를 분리시켜야 할 경우.
 - 이러한 큐는 다양한 환경에서 가능한 최상의 성능을 필요로 하거나 버퍼 풀의 다른 큐에 부정적 영향을 미치지 않도록 분리시켜야 할 수 있습니다.
 - 각 큐는 자체 버퍼 풀과 페이지 세트로 분리시킬 수 있습니다.
- 서비스 클래스로 인해 여러 다른 큐 세트를 서로 분리시키려는 경우.
 - 버퍼 풀 설정에 제안된 정의에 설명된 대로, 각 큐 세트마다 두 가지 유형의 버퍼 풀 1 또는 2 중 하나나 둘 모두가 필요할 수 있어서, 특정 유형의 여러 버퍼 풀을 작성해야 합니다.

각 버퍼 풀에 대한 설정 결정

[143 페이지](#)의 『정의할 버퍼 풀의 수 결정』에 설명된 네 개의 버퍼 풀을 사용 중이면 버퍼 풀 설정을 위한 제안된 정의에서 버퍼 풀 크기에 대한 두 개의 값 세트를 제공합니다.

첫 번째 세트는 테스트 시스템에 적합하고 다른 세트는 프로덕션 시스템이나 결국은 프로덕션 시스템이 될 시스템에 적합합니다. 모든 경우에 **LOCATION(ABOVE)** 속성이 있는 버퍼 풀을 정의하십시오.

표 21. 버퍼 풀 설정을 위한 제안된 정의		
정의 설정	테스트 시스템	프로덕션 시스템
BUFFPOOL 0	1 050 버퍼	50 000 버퍼
BUFFPOOL 1	1 050 버퍼	20 000 버퍼
BUFFPOOL 2	1 050 버퍼	50 000 버퍼
BUFFPOOL 3	1 050 버퍼	20 000 버퍼

제안된 네 개의 버퍼 풀보다 더 많이 필요하면 버퍼 풀의 예상된 큐 작동을 가장 정확하게 설명하는 버퍼 풀(1 또는 2)을 선택하고 버퍼 풀 설정에 제안된 정의의 정보를 사용하여 크기를 조절하십시오.

모든 버퍼 풀을 막대 위에서 찾을 수 있도록 MEMLIMIT가 충분히 높게 설정되어 있는지 확인하십시오.

예상한 로드 하의 버퍼 풀 성능 모니터

버퍼 풀 성능 통계를 분석하여 버퍼 풀의 사용을 모니터할 수 있습니다. 특히, QPSTSOS, QPSTSTLA, QPSTDMC의 값이 0으로 유지되도록 버퍼 풀이 충분히 크지 확인해야 합니다.

추가 정보는 [버퍼 관리자 데이터 레코드를 참조하십시오.](#)

버퍼 풀 특성 조정

필요에 따라 다음을 사용하여 144 페이지의 『각 버퍼 풀에 대한 설정 결정』의 버퍼 풀 설정을 조정하십시오.

145 페이지의 『예상한 로드 하의 버퍼 풀 성능 모니터』의 성능 통계를 지침으로 사용하십시오.

1. IBM MQ의 이전 버전에서 마이그레이션하는 경우 사용 가능한 실제 스토리지가 더 있는 경우에만 기존 설정을 변경하십시오.
2. 일반적으로 버퍼 풀이 더 클수록 성능이 더 우수하며 버퍼 풀이 막대 위에 있으면 훨씬 더 클 수 있습니다.
하지만 버퍼 풀이 실제 스토리지에 상주하도록 항상 사용 가능한 실제 스토리지가 충분히 있어야 합니다. 페이지를 유발하는 큰 버퍼 풀보다는 페이지를 유발하지 않는 더 작은 버퍼 풀이 더 좋습니다.
또한 발생할 가능성이 있는 경우 페이지 세트 확장을 고려해야 하더라도 사용하는 페이지 세트의 총 크기보다 큰 버퍼 풀은 의미가 없습니다.
3. 더 나은 애플리케이션 격리를 제공하므로 버퍼 풀당 하나의 페이지 세트를 목표로 합니다.
4. 실제 스토리지가 충분해서 운영 체제를 통해 버퍼 풀이 페이지되지 않는 경우 버퍼 풀에 페이지 고정 버퍼를 사용할 것을 고려하십시오.

이는 특히 버퍼 풀의 I/O가 많을 경우 I/O 이전에 버퍼를 페이지 고정하고 그 후에 고정 해제하는 것과 연관된 CPU 비용이 절감되므로 중요합니다.

5. 버퍼 풀이 막대 아래에 위치할 정도로 충분히 작지만 막대 위에 두면 여러 혜택이 있습니다. 이러한 항목은 다음과 같습니다.
 - 31비트 가상 스토리지 제한조건 완화 - 예를 들어, 공용 스토리지의 공간이 더 많아집니다.
 - 버퍼 풀이 과도하게 사용될 때 예상치 못하게 버퍼 풀의 크기를 늘려야 하는 경우 큐 관리자에 미치는 영향과 위험이 덜하며 이미 막대 위에 있는 버퍼 풀에 버퍼를 더 추가함으로써 버퍼 풀을 막대 위로 이동시킨 후 버퍼를 더 추가하는 것보다 워크로드가 덜합니다.
6. 15% 여유 임계값이 절대 초과되지 않도록(즉, QPSTCBSL 나누기 QPSTNBUF가 항상 15%를 넘도록) 버퍼 풀 0 및 단기 메시지용 버퍼 풀(버퍼 풀 2)을 성능 조정하십시오. 버퍼가 15%를 초과하여 여유 상태이면 두 체 크포인트 이전의 메시지가 페이지 세트에 기록되어도 이 버퍼 풀을 사용하는 페이지 세트의 I/O를 정상 조작 중에 대부분 피할 수 있습니다.



주의: 이 매개변수의 최적 값은 개별 시스템의 특성에 따라 다릅니다. 제공된 값은 단지 가이드라인으로서만 제공되며 사용자 시스템에는 맞지 않을 수 있습니다.

7. 시스템* 매우 깊은 큐(예: SYSTEM.CHANNEL.SYNCQ)는 충분한 스토리지를 사용할 수 있으면 고유 버퍼 풀에 배치하는 것이 유리할 수 있습니다.

IBM MQ SupportPac MP16 - z/OS 용 IBM MQ 용량 계획 및 튜닝에 버퍼 풀 성능 조정에 대한 추가 정보가 제공됩니다.

z/OS 로깅 환경 계획

이 주제를 사용하여 IBM MQ에 사용되는 로그 아카이브와 로그의 수, 크기, 배치를 계획하십시오.

로그는 다음 용도로 사용됩니다.

- 지속 메시지에 대한 복구 정보 쓰기
- 지속 메시지를 사용하여 작업 단위에 대한 정보 기록
- 큐 정의와 같은 오브젝트 변경에 대한 정보 기록

- CF 구조 백업

및 기타 내부 정보용.

IBM MQ 로깅 환경은 단일 또는 이중 활성 로그의 사용 여부, 아카이브 로그 볼륨에 사용할 매체, 보유할 로그 버퍼 수와 같은 옵션을 지정하는 시스템 매개변수 매크로를 사용하여 설정됩니다.

이러한 매크로는 부트스트랩 및 로그 데이터 세트 작성 및 시스템 매개변수 모듈 조정에 설명되어 있습니다.

참고: 큐 공유 그룹을 사용 중이면 반드시 SHAREOPTIONS(2 3)으로 부트스트랩 및 로그 데이터 세트를 정의하십시오.

이 절에는 다음 주제에 대한 정보가 포함되어 있습니다.

z/OS 로그 데이터 세트 정의

이 주제를 사용하여 로그 데이터 세트의 가장 적절한 구성을 결정하십시오.

이 주제는 다음 질문에 대답하는 데 도움이 되는 정보를 포함합니다.

- 설치에서 단일 또는 이중 로깅을 사용해야 하는 이유
- 필요한 활성 로그 데이터 세트 수
- 147 페이지의 『활성 로그는 얼마나 커야 하나?』
- 활성 로그 배치
- 148 페이지의 『z/OS 데이터 세트 암호화를 사용한 활성 로그 암호화』

설치에 단일 또는 이중 로깅을 사용해야 하나?


일반적으로 데이터 손실 위험을 최소화하기 위해 프로덕션에는 이중 로깅을 사용해야 합니다. 테스트 시스템이 프로덕션을 반영하게 하려면 두 시스템이 모두 이중 로깅을 사용해야 하며, 그렇지 않은 경우에는 테스트 시스템이 단일 로깅을 사용할 수 있습니다.

단일 로깅의 경우 한 세트의 로그 데이터 세트에 데이터가 기록됩니다. 이중 로깅의 경우에는 두 세트의 로그 데이터 세트에 데이터가 기록되므로, 데이터 세트가 우발적으로 삭제되는 것과 같은 문제가 한 로그 데이터 세트에 발생할 경우 다른 로그 세트의 해당 데이터 세트를 사용하여 데이터를 복구할 수 있습니다.

이중 로깅의 경우 단일 로깅보다 DASD가 두 배 더 필요합니다.

이중 로깅을 사용하면 데이터 복구의 적합한 프로비저닝을 위해 이중 BSDS 및 이중 아카이브도 사용하십시오.

이중 활성 로깅은 추가되는 성능 비용이 적습니다.

 **주의:** Metro Mirror와 같은 디스크 미러링 기술의 사용이 반드시 이중 로깅 및 이중 BSDS를 대체할 필요는 없습니다. 미러링된 데이터 세트가 우발적으로 삭제될 경우 두 사본이 모두 손실됩니다.

지속 메시지를 사용하는 경우 단일 로깅은 최대 용량을 10 - 30% 정도 늘리고 응답 시간도 개선할 수 있습니다.

동일한 수의 활성 로그를 제공하는 데 단일 로깅은 2 - 310개의 활성 로그 데이터 세트를 사용하는 반면 이중 로깅은 4 - 620개의 활성 로그 데이터 세트를 사용합니다. 따라서 단일 로깅은 로그되는 데이터의 양을 감소시키며 이는 설치가 I/O 제한 방식인 경우에 중요할 수 있습니다.

얼마나 많은 활성 로그 데이터 세트가 필요하나?

로그 수는 큐 관리자의 활동에 따라 다릅니다. 처리량이 낮은 테스트 시스템의 경우 세 개의 활성 로그 데이터 세트가 적합할 수 있습니다. 처리량이 높은 프로덕션 시스템의 경우에는 사용 가능한 최대 수의 로그를 원할 수 있으므로 로그 오프로딩에 대한 문제가 있는 경우 문제를 해결할 시간이 더 많습니다.

적어도 세 개의 활성 로그 데이터 세트가 있어야 하지만 더 많이 정의하는 것이 더 좋습니다. 예를 들어, 로그를 채우는 데 걸리는 시간이 최대 로드 중 로그를 아카이브하는 데 걸리는 시간에 근접하면 더 많은 로그를 정의하십시오.

참고: 페이지 세트 및 활성 로그 데이터 세트는 EAV (Extended Address Volume) 의 EAS (Extended Address Space) 파트에 상주할 수 있으며 아카이브 로그 데이터 세트도 EAS에 상주할 수 있습니다.

로그 아카이브의 가능한 지연을 오프셋하려는 경우에도 더 많은 로그를 정의해야 합니다. 테이프에 있는 아카이브 로그를 사용하는 경우 테이프 마운트에 필요한 시간을 허용하십시오.

DASD 부족이나 테이프 쓰기 불가능으로 인해 시스템이 아카이브할 수 없는 경우에 하루의 데이터를 보관할 만큼 충분한 활성 로그 공간을 보유할 것을 고려하십시오. 모든 활성 로그가 가득 차면 IBM MQ에서 지속 메시지 또는 트랜잭션을 처리할 수 없게 됩니다. 충분한 활성 로그 공간을 확보하는 것이 매우 중요합니다.

아카이브 지연이나 문제점으로 인한 영향을 최소화하기 위한 방법으로 새 활성 로그 데이터 세트를 동적으로 정의할 수 있습니다. **DEFINE LOG** 명령을 사용하여 활성 로그에 공간이 부족하여 큐 관리자 'stall' 을 방지함으로써 새 데이터 세트를 신속하게 온라인으로 설정할 수 있습니다.

31개를 초과한 활성 로그 데이터 세트를 정의하려면 버전 2 형식 BSDS를 사용하도록 로깅 환경을 구성해야 합니다. 버전 2 형식 BSDS가 사용되면 각 로그 복사 링마다 최대 310개의 활성 로그 데이터 세트를 정의할 수 있습니다. 버전 2 형식 BSDS로 변환하는 방법에 대한 정보는 154 페이지의 『주소 지정 가능 최대 로그 범위 증가 계획』의 내용을 참조하십시오.

로그 맵 인쇄 유틸리티(CSQJU004)를 실행하거나 큐 관리자 초기화 중 발행된 CSQJ034I 메시지에서 큐 관리자가 버전 2 이상 BSDS를 사용 중인지 확인할 수 있습니다. CSQJ034I 메시지에서 FFFFFFFFFFFFFFFFFF의 로그 RBA 범위 끝은 버전 2 이상 형식 BSDS가 사용 중임을 표시합니다. 0000FFFFFFFFFFFFFFFF의 로그 RBA 범위의 끝은 CSQJ034I 메시지에서 버전 1 포맷 BSDS가 사용 중임을 표시합니다.

큐 관리자가 버전 2이상의 형식 BSDS를 사용 중인 경우 **DEFINE LOG** 명령을 사용하여 31개이상의 활성 로그 데이터 세트를 로그 사본 링에 동적으로 추가할 수 있습니다.

활성 로그는 얼마나 커야 하나?

IBM MQ 8.0에서 디스크로 아카이브할 때 지원되는 최대 활성 로그 크기는 4GB입니다. 제품의 이전 릴리스에서는 디스크에 아카이브할 때 지원되는 최대 활성 로그 크기가 3GB입니다.

테이프에 아카이브할 때 최대 활성 로그 크기는 4GB입니다.

프로덕션 및 테스트 시스템에 크기가 최소 1GB인 활성 로그를 작성해야 합니다.

중요사항: IDCAMS가 할당하는 크기를 반올림하므로 데이터 세트를 할당할 때 주의해야 합니다.

3GB 로그를 할당하려면 다음 옵션 중 하나를 지정하십시오.

- Cylinders(4369)
- Megabytes(3071)
- TRACKS(65535)
- RECORD(786420)

이들 중 하나는 2.99995GB를 할당합니다.

4GB 로그를 할당하려면 다음 옵션 중 하나를 지정하십시오.

- Cylinders(5825)
- Megabytes(4095)
- TRACKS(87375)
- RECORD(1048500)

이들 중 하나는 3.9997GB를 할당합니다.

데이터 세트가 여러 볼륨으로 분산된 스트라이핑 데이터 세트를 사용할 때에는 스트라이핑에 사용된 각 DASD 볼륨마다 지정된 크기 값이 할당됩니다. 따라서 스트라이핑에 네 개의 볼륨 및 4GB 로그를 사용하려면 다음을 지정해야 합니다.

- CYLinders(1456)
- Megabytes(1023)

이 속성을 설정하면 $4 * 1456 = 5824$ 실린더 또는 $4 * 1023 = 4092$ 메가바이트가 할당됩니다.

참고: 스트라이핑은 확장 형식 데이터 세트를 사용할 때 지원됩니다. 일반적으로 스토리지 관리자가 이를 설정합니다.

프로시저 수행에 대한 정보는 [활성 로그 크기 늘리기](#)를 참조하십시오.

활성 로그 배치

큐 관리자를 위한 스토리지 풀을 설정하기 위해 스토리지 관리 팀과 작업하십시오. 다음을 고려해야 합니다.

- 큐 관리자가 올바른 SMS 정의를 사용하기 위한 이름 지정 규칙.
- 활성 로그 및 아카이브 로그에 필요한 공간. 스토리지 풀은 하루 전체 동안의 활성 로그에 충분한 공간이 있어야 합니다.
- 실패에 대한 성능 및 복원력.

성능상의 이유로 활성 로그 데이터 세트를 스트라이핑할 것을 고려해야 합니다. I/O가 다수의 볼륨에 분산되고 I/O 응답 시간이 감소하여 처리량이 더 높아집니다. 스트라이핑 사용 시의 활성 로그 크기 할당에 대한 정보는 이전 텍스트를 참조하십시오.

RMF 또는 유사한 제품의 보고서를 사용하여 I/O 통계를 검토해야 합니다. 데이터 세트의 위치로 인해 지연이 발생하지 않도록 IBM MQ 데이터 세트에 대해 매월(또는 더 자주) 이러한 통계 검토를 수행하십시오.

일부 경우 더 많은 IBM MQ 페이지 세트 I/O가 있을 수 있으며 동일한 DASD에 위치한 경우 이는 IBM MQ 로그 성능에 영향을 미칠 수 있습니다.

이중 로깅을 사용하는 경우 활성 및 아카이브 로그의 각 세트를 반드시 분리해서 보관하십시오. 예를 들어, 별도의 DASD 서브시스템이나 각기 다른 디바이스에 각 세트를 할당하십시오.

그러면 한 볼륨이 손상되거나 영구 삭제될 경우에 둘 모두가 손실되는 위험이 감소합니다. 로그의 두 사본이 모두 손실된 경우에는 데이터 손실 확률이 높습니다.

새 활성 로그 데이터 세트를 작성할 때 CSQJUFMT를 사용하여 데이터 세트를 미리 형식화해야 합니다. 로그가 사전 형식화되지 않은 경우 큐 관리자는 처음으로 로그를 사용할 때 로그를 형식화하며 이는 성능에 영향을 줍니다.

스피닝이 큰 디스크가 있는 이전 DASD의 경우 최상의 성능을 위해 어느 볼륨을 사용할지 주의해야 했습니다.

많은 PC 크기의 디스크로 데이터가 분산되는 최신의 DASD의 경우는 어느 볼륨을 사용할지 많이 걱정할 필요가 없습니다.

스토리지 관리자가 엔터프라이즈 DASD를 검사해서 성능 문제점을 검토하고 해결해야 합니다. 가용성을 위해 한 DASD 서브시스템에 하나의 로그 세트를 사용하고 다른 DASD 서브시스템에는 이중 로그를 사용할 수 있습니다.

z/OS 데이터 세트 암호화를 사용한 활성 로그 암호화

IBM MQ for z/OS 9.1.4 이상에서 실행 중인 큐 관리자의 활성 로그 데이터 세트에 z/OS 데이터 세트 암호화 기능을 적용할 수 있습니다.

데이터가 AES 암호화되도록 하는 데이터 세트 키 레이블 및 EXTENDED 속성을 사용하여 이러한 활성 로그 데이터 세트를 할당해야 합니다.

데이터 세트 암호화를 사용하는 IBM MQ for z/OS의 저장 데이터에 대한 기밀성 섹션을 참조하십시오. 참조하십시오.

IBM MQ의 MetroMirror 사용

이전에는 SPPRC(Synchronous Peer to Peer Remote Copy)라고 알려진, IBM Metro Mirror는 쓰기 작업이 완료된 것으로 간주되기 전에 기본 볼륨과 보조 볼륨 모두에서 쓰기 작업이 완료된 두 스토리지 서브시스템 간의 동기식 복제 솔루션입니다. Metro Mirror는 스토리지 서브시스템 장애 시 데이터 손실을 요구하지 않는 환경에서 사용할 수 있습니다.

지원되는 데이터 세트 유형

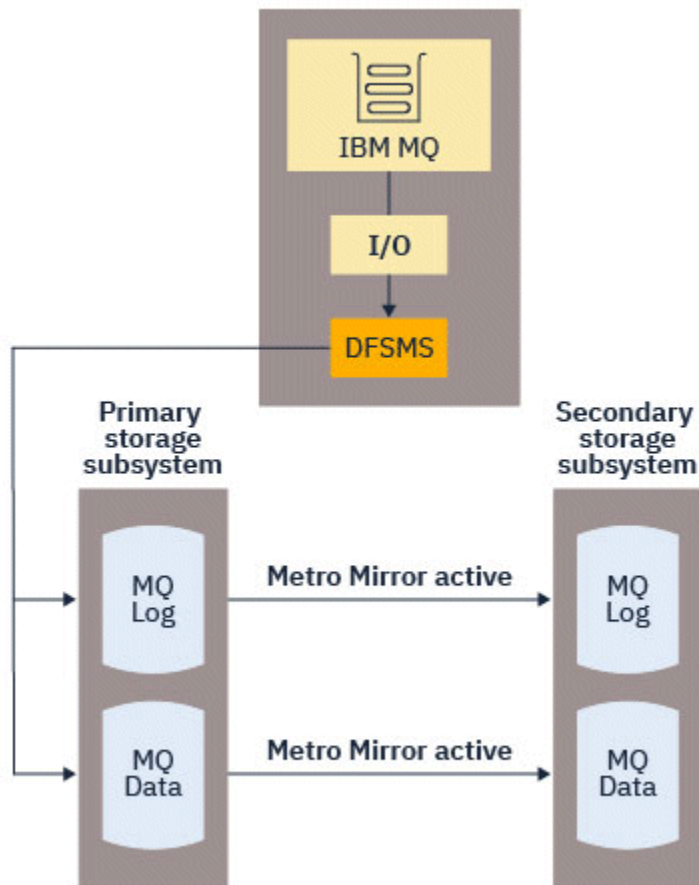
다음의 모든 IBM MQ 데이터 세트 유형은 Metro Mirror를 사용하여 복제할 수 있습니다. 그러나 정확하게 어떤 것을 복제할 것인지는 엔터프라이즈의 가용성 요구사항에 따라 다릅니다.

- 활성화 로그
- 아카이브 로그
- 부트스트랩 데이터 세트(BSDS)
- 페이지 세트
- 공유 메시지 데이터세트(SMDS)
- 예를 들어, MSTR JCL의 CSQINP* DD 카드에서 구성에 사용된 데이터 세트

IBM MQ 활성화 로그와 함께 zHyperWrite 사용

Metro Mirror를 사용하여 복제된 데이터 세트에 쓰기를 수행할 때, 쓰기는 먼저 기본 볼륨에 작성된 다음 보조 볼륨에 복제됩니다. 이 복제는 스토리지 서브시스템에 의해 수행되며, 쓰기를 실행한 애플리케이션, 예를 들면, IBM MQ에 대해 투명합니다.

이 프로세스는 다음 다이어그램에서 설명합니다.



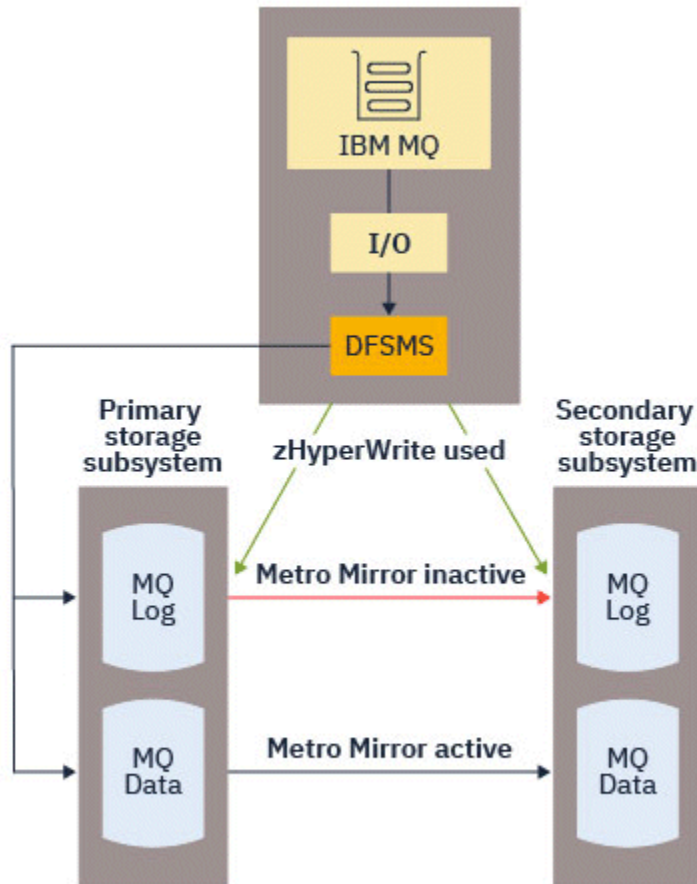
쓰기가 IBM MQ로 리턴되기 전에 기본 및 보조 스토리지 서브시스템에 대한 쓰기를 모두 완료해야 하기 때문에 Metro Mirror 사용은 성능에 영향을 미칠 수 있습니다. 이 성능 영향과 Metro Mirror 사용의 가용성 이점의 밸런스를 맞추어야 합니다.

IBM MQ 활성화 로그는 Metro Mirror의 성능 영향에 가장 민감합니다. IBM MQ에서는 이 성능 영향을 줄이는 데 도움이 되도록 활성화 로그와 함께 zHyperWrite를 사용할 수 있습니다.

zHyperWrite는 Metro Mirror를 사용하여 복제되는 데이터 세트에 대한 쓰기의 성능 영향을 줄이기 위해 z/OS와 함께 작동하는 스토리지 서브시스템 기술입니다. zHyperWrite를 사용하면 기본 및 보조 볼륨에 대한 쓰기가 스

토리지 서브시스템 레벨에서 순차적으로 실행되는 대신 DFSMS(Data Facility Storage Management Subsystem) 레벨에서 병렬로 실행되므로 성능에 미치는 영향이 감소합니다.

다음 다이어그램은 활성 로그에 사용되는 zHyperWrite 및 다른 IBM MQ 데이터 세트 유형에 사용되는 Metro Mirror를 설명합니다. zHyperWrite 쓰기가 실패하면 DFSMS가 Metro Mirror를 사용하여 쓰기를 투명하게 재발행한다는 점을 참고하십시오.



IBM MQ의 zHyperWrite는 활성 로그 데이터 세트에서만 지원됩니다.

활성 로그로 zHyperWrite를 사용하려면 다음을 수행해야 합니다.

- zHyperWrite를 사용하도록 IBM MQ를 구성하고
- 활성 로그는 zHyperWrite 가능한 볼륨에 있어야 합니다.

다음 방법 중 하나를 사용하여 zHyperWrite를 사용하기 위해 IBM MQ를 구성할 수 있습니다.

- 시스템 매개변수 모듈에서 ZHYWRITE(YES)를 지정하십시오.
- SET LOG ZHYWRITE(YES) 명령을 실행하십시오.

활성 로그 데이터 세트가 zHyperWrite 가능 볼륨에 있도록 다음 조건을 설정하십시오.

- Metro Mirror 볼륨 및 zHyperWrite 지원 볼륨을 사용으로 설정
- 볼륨이 HyperSwap을 사용하도록 설정되었는지 확인
- IECIOSxx 매개변수에서 HYPERWRITE=YES 지정

V 9.3.5 IBM MQ 9.3.5 이전에는 모든 이전 조건이 충족되면 zHyperWrite에 대해 활성 로그에 대한 쓰기가 사용으로 설정되었습니다. 이러한 조건 중 하나 또는 그 이상이 충족되지 않으면 IBM MQ는 정상적으로 활성 로그에 기록되고 Metro Mirror는 구성된 경우 쓰기를 복제합니다.

V 9.3.5 IBM MQ 9.3.5부터 ZHYWRITE (YES) 가 지정되면, IBM MQ 는 로그가 zHyper쓰기 가능 볼륨에 있는지 여부에 관계없이 활성 로그에 쓸 때 항상 zHyperWrite를 사용하려고 시도합니다. 로그가 zHyper쓰기 가

능 볼륨에 없는 경우 Metro Mirror 가 구성되어 있으면 쓰기를 복제합니다. 로그가 zHyper쓰기 가능 볼륨에 없는 경우 zHyperWrite를 사용하려고 시도해도 부정적인 영향을 미치지 않습니다.

참고:

- IBM MQ는 모든 활성 로그 데이터 세트가 zHyperWrite 가능 볼륨에 있을 것을 요구하지 않습니다. IBM MQ가 일부 활성 로그 데이터 세트가 zHyperWrite 가능 볼륨에 있음을 감지하고 다른 로그 데이터 세트가 그렇지 않으면 CSQJ166E 메시지를 발행하고 처리를 계속 수행합니다.

- IBM MQ는 데이터 세트를 처음 열 때 활성 로그 데이터 세트가 zHyperWrite 가능한지 여부를 확인합니다. 로그 데이터 세트는 큐 관리자가 시작할 때 또는 DEFINE LOG 명령을 사용하여 동적으로 추가할 때 열립니다. 큐 관리자가 로그 데이터 세트를 열고 있는 동안 로그 데이터 세트가 zHyperWrite 가능하도록 설정되는 경우 큐 관리자는 다시 시작될 때까지 이를 감지하지 못합니다.

DISPLAY LOG 명령의 출력을 사용하여 현재 활성 로그 데이터 세트가 zHyperWrite를 사용할 수 있는지 여부를 표시할 수 있습니다. 다음 예제는 두 개의 데이터 세트 모두 zHyperWrite 가능으로 설정되었음을 표시합니다. 큐 관리자가 ZHYWRITE(YES)로 구성된 경우, 이러한 로그에 대한 쓰기는 zHyperWrite에 대해 사용으로 설정됩니다.

```
Copy %Full zHyperWrite DSName
1 4 CAPABLE MQTST.SUBSYS.MQDL.LOGCOPY1.DS001
2 4 CAPABLE MQTST.SUBSYS.MQDL.LOGCOPY2.DS001
```

z/OS 로그 아카이브 스토리지 계획

이 주제를 사용하여 아카이브 로그 데이터 세트를 유지보수하기 위한 여러 다른 방법을 이해하십시오.

아카이브 로그 데이터 세트를 표준 레이블 테이프 또는 DASD에 두고 DFHSM(Data Facility Hierarchical Storage Manager)으로 이를 관리할 수 있습니다. 아카이브 로그 데이터 세트의 각 z/OS 논리 레코드는 활성 로그 데이터 세트로부터의 VSAM 제어 간격입니다. 블록 크기는 4KB의 배수입니다.

아카이브 로그 데이터 세트는 IBM MQ가 선택한 이름과 함께 동적으로 할당됩니다. 이러한 할당에 필요한 데이터 세트 이름 접두부, 블록 크기, 단위 이름, DASD 크기는 시스템 매개변수 모듈에 지정됩니다. 설치 시 IBM MQ가 아카이브 로그 데이터 세트 이름에 날짜 및 시간을 추가하도록 선택할 수도 있습니다.

IBM MQ의 경우 새 아카이브 로그에 대한 특정 볼륨을 지정할 수 없으며 이를 관리하려면 스토리지 관리 루틴을 사용할 수 있습니다. 할당 오류가 발생하면 다음 번 오프로딩이 트리거될 때까지 오프로딩이 연기됩니다.

설치 시 이중 아카이브 로그를 지정하면 활성 로그에서 검색된 각 로그 제어 간격이 두 개의 아카이브 로그 데이터 세트에 기록됩니다. 아카이브 로그 데이터 세트 쌍에 포함된 로그 레코드는 동일하지만 다중 볼륨 데이터 세트의 볼륨 끝 지점은 동기화되지 않습니다.

아카이브 로그는 테이프 또는 DASD에 상주해야 합니까?

아카이브 로그에 테이프 또는 DASD를 사용할지 결정할 때 고려해야 할 수많은 요인이 있습니다.

- 테이프 또는 디스크에 대해 결정하기 전에 작동 프로시저를 검토하십시오. 예를 들어, 테이프에 아카이브하도록 선택할 경우 요구 시 충분한 테이프 드라이브가 있어야 합니다. 재해 후 모든 서브시스템이 테이프 드라이브를 원할 때 예상하는 만큼의 여유 테이프 드라이브가 없을 수도 있습니다.
- 복구 중 테이프의 아카이브 로그는 테이프가 마운트되는 즉시 사용 가능합니다. DASD 아카이브가 사용되었고 데이터 세트가 HSM(Hierarchical Storage Manager)을 사용하여 테이프에 마이그레이션된 경우 HSM이 각 데이터 세트를 디스크로 다시 불러오는 동안 지연이 발생합니다. 아카이브 로그가 사용되기 전에 데이터 세트를 다시 불러올 수 있습니다. 하지만 필요한 올바른 순서를 예측하는 것이 항상 가능하지는 않습니다.
- DASD에서 아카이브 로그를 사용할 때 많은 로그가 필요하면(백업에서 복원한 후 페이지 세트를 복구할 때와 같은 경우일 수 있음) 모든 아카이브 로그를 보유하기 위해 상당한 양의 DASD를 필요로 할 수 있습니다.
- 사용량이 낮은 시스템이나 테스트 시스템에서는 테이프 마운트가 필요하지 않도록 DASD에 아카이브 로그를 두는 것이 보다 편리할 수 있습니다.

- 둘 모두 RECOVER CFSTRUCT 명령을 발행하고 지속 작업 단위를 백아웃하면 로그가 역방향으로 읽힙니다. 하드웨어 압축이 있는 테이프 드라이브는 역방향으로 읽는 조작에서 제대로 수행하지 않습니다. 테이프에서의 역방향 읽기를 피하려면 DASD에 충분한 로그 데이터를 계획하십시오.

DASD에 아카이브하면 보다 빠른 복구가 가능하지만 테이프에 아카이브할 때보다 비쌉니다. 이중 로깅을 사용하면 아카이브 로그의 1차 사본은 DASD에 두고 2차 사본은 테이프에 두도록 지정할 수 있습니다. 그러면 많은 DASD를 사용하지 않고도 복구 속도가 증가하며 테이프를 백업으로 사용할 수 있습니다.

테이프에서 DASD로 로그를 아카이브하는 방법과 역방향 프로세스를 수행하는 방법에 관한 자세한 정보는 [153 페이지의 『아카이브 로그의 스토리지 매체 변경』](#)의 내용을 참조하십시오.

테이프에 아카이브

테이프 디바이스에 아카이브하도록 선택하면 IBM MQ가 최대 20개 볼륨으로 확장할 수 있습니다.

세트가 한 테이프 볼륨에 맞도록 활성 로그 데이터 세트의 크기 변경을 고려하는 경우 BSDS의 사본은 활성 로그 데이터 세트의 사본과 동일한 테이프 볼륨에 위치함에 유의하십시오. 활성 로그 데이터 세트의 크기를 줄여서 테이프 볼륨의 BSDS에 필요한 공간을 오프셋하십시오.

테이프에 이중 아카이브 로그를 사용하는 경우에는 한 사본을 로컬에 보유하고 다른 사본은 재해 복구에 사용할 수 있게 오프사이트에 보유하는 것이 일반적입니다.

DASD 볼륨에 아카이브

IBM MQ는 비테이프 디바이스(DASD)에 할당된 모든 아카이브 로그 데이터 세트를 카탈로그화하도록 요구합니다. DASD에 아카이브하도록 선택하는 경우 CSQ6ARVP 매크로의 CATALOG 매개변수는 YES여야 합니다. 이 매개변수가 NO이고 아카이브 로그 데이터 세트를 DASD에 배치하기로 결정한 경우, IBM MQ가 여전히 데이터 세트를 카탈로그화하더라도 아카이브 로그 데이터 세트가 할당될 때마다 CSQJ072E 메시지를 수신합니다.

아카이브 로그 데이터 세트가 DASD에 보유된 경우에는 아카이브 로그 데이터 세트가 또 다른 볼륨으로 확장될 수 있습니다. 즉, 다중 볼륨이 지원됩니다.

DASD를 사용하도록 선택하면 1차 공간 할당(수량 및 블록 크기 모두)이 활성 로그 데이터 세트 또는 해당 BSDS의 데이터를 포함할 정도로 충분히 크지 확인하십시오(둘 중 어느 것이 더 크든 상관 없이).

이는 오프로드 프로세스 동안 원하지 않는 z/OS X' B37 ' 또는 X' E37 ' 이상종료 코드의 가능성을 최소화합니다. 1차 공간 할당은 CSQ6ARVP 매크로의 PRIQTY(1차 수량) 매개변수로 설정됩니다.

아카이브 로그 데이터 세트는 대형 또는 확장 형식 순차 데이터 세트에 존재할 수 있습니다. SMS ACS 루틴은 이제 DSNTYPE (LARGE) 또는 DSNTYPE (EXT) 을 사용합니다.

IBM MQ에서는 아카이브 로그를 확장 형식 데이터 세트로 할당할 수 있습니다. 확장 형식이 사용되는 경우 최대 아카이브 로그 크기는 65535개 트랙에서 최대 활성 로그 크기인 4GB까지 증가합니다. 아카이브 로그는 확장 주소 볼륨(EVA)의 확장 주소 지정 공간(EAS)에서 할당할 수 있습니다.

필수 하드웨어 및 소프트웨어 레벨을 사용할 수 있는 경우 zEDC를 사용하여 COMPACTION으로 정의된 데이터 클래스에 아카이브 로그를 할당하면 아카이브 로그를 보유하기 위해 필요한 디스크 스토리지가 감소합니다. 자세한 정보는 IBM MQ for z/OS: 자세한 정보는 IBM zEnterprise Data Compression (zEDC) 및 zEnterprise Data Compression (zEDC) 을 사용하여 스토리지 점유 축소를 참조하십시오.

z/OS 데이터 세트 암호화 기능은 IBM MQ에서 실행 중인 큐 관리자의 아카이브 로그에 적용할 수 있습니다. 이러한 아카이브 로그는 ACS(Automatic Class Selection) 루틴을 통해 TEXTENDED 속성으로 정의된 데이터 클래스 및 데이터를 AES로 암호화해야 하는 데이터 세트 키 레이블에 할당해야 합니다.

아카이브 로그 데이터 세트와 SMS 사용

MVS/DFP 스토리지 관리 서브시스템(DFSMS)을 설치한 경우 아카이브 로그 데이터 세트에 대한 ACS(Automatic Class Selection) 사용자 엑시트 필터를 쓸 수 있으며 이는 SMS 환경에 맞게 데이터 세트를 변환시키는 데 유용합니다.

예를 들어, 이러한 필터는 DFSMS가 관리할 수 있는 DASD 데이터 세트로 출력을 라우팅할 수 있습니다. 이러한 방식으로 ACS 필터를 사용하는 경우 주의해야 합니다. SMS는 DASD 데이터 세트의 카탈로그화를 요구하

므로 CSQ6ARVP 매크로의 CATALOG DATA 필드에 YES가 포함되었는지 확인해야 합니다. 그렇지 않으면 CSQJ072E 메시지가 리턴됩니다. 하지만 IBM MQ는 여전히 데이터 세트를 카탈로그화합니다.

ACS 필터에 대한 자세한 정보는 [집계 백업 처리 중 DFSMSHsm에서 동적으로 할당하는 데이터 세트를 참조 하십시오.](#)

z/OS 아카이브 로그의 스토리지 매체 변경
아카이브 로그에서 사용하는 스토리지 매체를 변경하기 위한 프로시저입니다.

이 태스크 정보

이 태스크에서는 아카이브 로그에 사용되는 스토리지 매체를 변경하는 방법을 설명합니다(예: 테이프에 아카이브에서 DASD에 아카이브로 이동).

다음 중에서 변경 방법을 선택할 수 있습니다.

1. 다음 번에 큐 관리자가 다시 시작할 때부터 적용되도록 CSQ6ARVP 매크로만 사용해서 변경합니다.
2. CSQ6ARVP 매크로를 사용하여 변경하고 SET ARCHIVE 명령을 사용하여 동적으로 변경합니다. 즉, 변경사항이 큐 관리자가 다음 번에 로그 파일을 아카이브하는 시점부터 적용되고 큐 관리자가 다시 시작된 후에도 지속됩니다.

프로시저

1. 아카이브 로그가 테이프가 아니라 DASD에 저장되도록 변경하는 방법:
 - a) 152 페이지의 『DASD 볼륨에 아카이브』 섹션을 읽고 CSQ6ARVP 매개변수를 검토하십시오.
 - b) CSQ6ARVP에서 다음 매개변수를 변경하십시오.
 - UNIT 및 필요한 경우 UNIT2 매개변수를 업데이트합니다.
 - DASD에 최적인 설정이 테이프와 다르므로 BLKSIZE 매개변수를 업데이트합니다.
 - PRIQTY와 SECQTY 매개변수를 가장 큰 활성 로그 또는 BSDS를 보유할 만큼 충분히 크게 설정합니다.
 - CATALOG 매개변수를 YES로 설정합니다.
 - ALCUNIT 설정이 원하는 것인지 확인합니다. BLK는 디바이스 유형과 무관하므로 이를 사용해야 합니다.
 - ARCWTOR 매개변수를 NO로 설정합니다(아직 설정되지 않은 경우).
2. 아카이브 로그가 DASD가 아니라 테이프에 저장되도록 변경하는 방법:
 - a) 152 페이지의 『테이프에 아카이브』 섹션을 읽고 CSQ6ARVP 매개변수를 검토하십시오.
 - b) CSQ6ARVP에서 다음 매개변수를 변경하십시오.
 - UNIT 및 필요한 경우 UNIT2 매개변수를 업데이트합니다.
 - 테이프에 최적인 설정이 DASD와 다르므로 BLKSIZE 매개변수를 업데이트합니다.
 - ALCUNIT 설정이 원하는 것인지 확인합니다. BLK는 디바이스 유형과 무관하므로 이를 사용해야 합니다.
 - ARCWTOR 매개변수의 설정을 검토합니다.

z/OS 아카이브 로그의 보관 기간

이 절의 정보를 사용하여 백업 전략을 계획하십시오.

USING CSQ6ARVP 또는 SET SYSTEM 명령에서 ARCETN 매개변수를 사용하여 아카이브 로그가 보관되는 기간(일)을 지정합니다. 이 기간이 지나면 z/OS에서 데이터 세트가 삭제될 수 있습니다.

아카이브 로그 데이터 세트가 더 이상 필요하지 않으면 수동으로 이를 삭제할 수 있습니다.

- 큐 관리자는 복구를 위해 아카이브 로그를 필요로 할 수 있습니다.

큐 관리자는 최신 1000개의 아카이브만을 BSDS에 보관할 수 있습니다. 아카이브 로그가 BSDS에 없으면 복구에 사용 불가능하며 감사, 분석 또는 재실행 유형 용도로만 사용됩니다.

- 로그에서 정보를 추출할 수 있도록 아카이브 로그를 보관할 수 있습니다. 예를 들어, 로그에서 메시지를 추출하고 메시지를 넣거나 가져오는 사용자 ID를 검토합니다.

BSDS는 로그에 대한 정보와 기타 복구 정보를 포함합니다. 이 데이터 세트는 고정 크기입니다. 아카이브 로그의 수가 CSQ6LOGP의 MAXARCH 값에 도달하거나 BSDS가 가득 차면 가장 오래된 아카이브 로그 정보를 덮어씁니다.

BSDS에서 아카이브 로그 항목을 제거하는 유틸리티가 있지만 일반적으로 BSDS는 가장 오래된 아카이브 로그 레코드를 래핑하고 오버레이합니다.

아카이브 로그가 필요한 시기

페이지 세트는 정기적으로 백업해야 합니다. 백업 빈도는 페이지 세트가 손실된 경우에 필요한 아카이브 로그를 판별합니다.

CF 구조는 정기적으로 백업해야 합니다. 백업 빈도는 CF 구조의 데이터가 손실된 경우에 필요한 아카이브 로그를 판별합니다.

아카이브 로그는 복구에 필요할 수 있습니다. 다음 정보는 여러 다른 IBM MQ 자원에 대한 문제가 있는 경우에 아카이브 로그가 필요할 수 있는 시기를 설명합니다.

페이지 세트의 손실

백업에서 시스템을 복구하고 큐 관리자를 다시 시작해야 합니다.

백업을 수행하기 전에 백업이 수행된 시점의 로그와 최대 3개의 로그 데이터 세트가 필요합니다.

모든 LPAR가 CF 구조에 대한 연결이 손실되었거나 구조를 사용할 수 없음

구조를 복구하려면 RECOVER CFSTRUCT 명령을 사용하십시오.

구조 복구는 백업을 수행한 큐 관리자의 로그에서 구조 백업 자체 외에, 마지막 백업 이후에(백업이 수행된 시점으로 돌아가서) 구조에 액세스한 모든 큐 관리자의 로그가 필요합니다.

CF 구조의 백업을 자주 수행한 경우 데이터는 활성 로그에 있어야 하며 아카이브 로그는 필요하지 않습니다.

CF 구조의 최근 백업이 없는 경우 아카이브 로그가 필요할 수 있습니다.

참고: 모든 비지속 메시지는 손실되고 모든 지속 메시지는 다음 태스크를 수행하여 재작성됩니다.

1. 로그에서 마지막 CF 구조 백업 읽기
2. 구조를 사용한 모든 큐 관리자에서 로그 읽기
3. 백업 이후의 업데이트를 병합하여

관리 구조 재빌드

관리 구조를 다시 빌드해야 하는 경우 정보는 QSG의 각 큐 관리자를 위한 로그의 마지막 체크포인트에서 읽습니다.

큐 관리자가 활성 상태가 아니면 QSG의 다른 큐 관리자가 로그를 읽습니다.

아카이브 로그는 필요하지 않습니다.

SMDS 데이터 세트 손실

SMDS 데이터 세트가 손실되거나 데이터 세트가 손상되면 데이터 세트를 사용할 수 없게 되고 FAILED 상태로 설정됩니다. CF 구조는 변경되지 않습니다.

SMDS 데이터 세트를 복원하려면 다음을 수행해야 합니다.

1. SMDS 데이터 세트를 재정의하십시오.
2. RECOVER CFSTRUCT 명령을 실행하여 CF 구조를 복구하십시오.

참고: CF 구조의 모든 비지속 메시지는 유실되고 모든 지속 메시지는 복원됩니다.

큐 관리자 로그에 대한 요구사항은 사용 불가능한 구조에서 복구하는 경우와 동일합니다.

주소 지정 가능 최대 로그 범위 증가 계획

대형 로그 상대 바이트 주소(RBA)를 사용하도록 큐 관리자를 구성하여 주소 지정 가능한 최대 로그 범위를 늘릴 수 있습니다.

IBM MQ for z/OS 8.0부터 로그 RBA 크기가 증가되었습니다. 이 변경에 대한 개요는 [대형 로그 상대 바이트 주소](#)를 참조하십시오.

V 9.3.0 IBM MQ 9.3.0 이상에서 작성된 큐 관리자는 기본적으로 8바이트로그 RBA를 사용하므로 변환이 필요하지 않습니다.

언제든지 8바이트로그 RBA값을 사용하도록 큐 관리자를 변환할 수 있습니다. 큐 공유 그룹에는 8바이트로그 RBA가 사용 가능한 일부 큐 관리자와 6바이트로그 RBA가 사용 가능한 일부 큐 관리자가 포함될 수 있습니다.

변경 실행 취소

변경은 백아웃할 수 없습니다.

변경 소요 시간

변경을 위해서는 큐 관리자를 재시작해야 합니다. 큐 관리자를 중지하고 부트스트랩 데이터 세트(BSDS) 또는 데이터 세트에 대해 CSQJUCNV 유틸리티를 실행하여 새 데이터 세트를 작성하고 해당 부트스트랩 데이터 세트 이름을 변경한 후에 큐 관리자를 재시작하십시오. CSQJUCNV 유틸리티는 일반적으로 실행하는데 수초가 걸립니다.

변경의 영향

- 8바이트 로그 RBA를 사용하면 로그 데이터 세트에 데이터를 쓸 때마다 바이트가 추가됩니다. 따라서 지속 메시지로 구성되는 워크로드의 경우 로그에 작성되는 데이터의 크기가 약간 증가됩니다.
- 페이지 세트 또는 커플링 기능(CF) 구조에 쓴 데이터는 영향을 받지 않습니다.

관련 태스크

[대형 로그 상대 바이트 주소 구현](#)

z/OS 채널 시작기 계획

채널 시작기는 큐 관리자 사이에 통신을 제공하고 자체 주소 공간에서 실행됩니다.

두 가지 유형의 연결이 있습니다.

1. 네트워크를 통해 큐 관리자에 애플리케이션을 연결. 이는 클라이언트 채널이라고도 합니다.
2. 큐 관리자 간의 연결. 이는 MCA 채널이라고도 합니다.

리스너

채널 리스너 프로그램은 수신 네트워크 요청을 대기하고 필요할 때 적절한 채널을 시작합니다. 인바운드 연결을 처리하기 위해서는 채널 시작기에 적어도 하나의 IBM MQ 리스너 태스크가 구성되어 있어야 합니다. 리스너는 TCP 리스너 또는 LU 6.2 리스너가 될 수 있습니다.

각 리스너에는 TCP 포트 또는 LU 이름이 필요합니다.

각 채널 시작기에 대해 둘 이상의 리스너를 가질 수 있음을 참고하십시오.

TCP/IP

채널 시작기는 동일한 z/OS 이미지에서 둘 이상의 TCP 스택으로 작동할 수 있습니다. 예를 들어, 내부 연결을 위해 하나의 TCP 스택을, 외부 연결을 위해 다른 TCP 스택을 이용할 수 있습니다.

출력 채널을 정의하는 경우

1. 연결의 목적지 호스트와 포트를 설정합니다. 이는 다음과 같습니다.
 - IP 주소(예: 10.20.4.6)
 - 호스트 이름(예: mvs-prod.myorg.com)

호스트 이름을 사용하여 목적지를 지정하는 경우 IBM MQ는 DNS(Domain Name System)를 사용하여 목적지의 IP 주소를 해석합니다.

2. 다중 TCP 스택을 사용하는 경우 사용할 IP 스택 주소를 지정하는 **LOCLADDR** 매개변수를 채널 정의에 지정할 수 있습니다.

고가용성 DNS 서버를 한 대 또는 여러 대 보유하도록 계획해야 합니다. DNS를 이용할 수 없으면 아웃바운드 채널을 시작할 수 없고 호스트 이름을 사용하여 수신 연결에 맵핑하는 채널 인증 규칙을 처리할 수 없습니다.

APPC 및 LU 6.2

APPC를 사용하는 경우 채널 시작기에는 APPC의 구성 및 LU 이름이 필요합니다.

큐 공유 그룹

단일 시스템 이미지를 제공하고 수신 IBM MQ 연결 요청이 큐 공유 그룹의 어떤 큐 관리자로도 이동할 수 있게 하려면 일부 구성을 수행해야 합니다. 예를 들면, 다음과 같습니다.

1. 하드웨어 네트워크 라우터. 이 라우터는 엔터프라이즈에서 볼 수 있는 하나의 IP 주소를 가지며 이 하드웨어에 연결된 임의의 큐 관리자에게 초기 요청을 라우트할 수 있습니다.
2. 가상 IP 주소(VIPA). 엔터프라이즈 전체 IP 주소가 지정되고 해당 주소를 SYSPLEX의 TCP 스택 중 하나에 라우트할 수 있습니다. TCP 스택은 그런 다음 SYSPLEX의 대기 큐 관리자로 이를 라우트할 수 있습니다.

IBM MQ 트래픽 보호

TLS 연결을 사용하여 연결의 데이터를 보호하도록 IBM MQ 를 구성할 수 있습니다. TLS를 사용하기 위해서는 디지털 인증서와 키 링이 필요합니다.

또한 호환 가능한 IBM MQ 정의 및 호환 가능 인증서를 보유하는지 확인하기 위해 채널의 원격 끝 지점에서 직원과 작업해야 합니다.

IBM MQ에 연결할 수 있는 연결 및 사용자 ID는 다음을 기반으로 제어할 수 있습니다.

- IP 주소
- 클라이언트 사용자 ID
- 리모트 큐 관리자 또는
- 디지털 인증서(채널 인증 레코드 참조)

유효한 사용자 ID와 비밀번호를 제공하는지 확인하여 클라이언트 애플리케이션을 제한하는 것도 가능합니다(연결 인증 참조).

채널 시작기를 작동시킨 후 한 번에 하나씩 TLS를 사용하도록 각 채널을 구성할 수 있습니다.

채널 시작기 모니터링

채널 시작기 및 채널에 대한 정보를 제공하는 MQSC 명령이 있습니다.

- **DISPLAY CHINIT** 명령은 채널 시작기 및 활성 리스너에 대한 정보를 제공합니다.
- **DISPLAY CHSTATUS** 명령은 채널의 활동 및 상태를 보여줍니다.

채널 시작기는 채널 시작기 태스크 및 채널 활동에 대한 정보를 가진 SMF 레코드를 생성할 수도 있습니다. 자세한 정보는 157 페이지의 『SMF 데이터에 대한 채널 시작기 계획』의 내용을 참조하십시오.

채널 시작기는 채널이 시작 및 중지될 때 작업 로그에 메시지를 생성합니다. 엔터프라이즈의 자동화 기능에서 이 메시지를 사용하여 상태를 캡처할 수 있습니다. 일부 채널은 몇 초 동안에만 활성화되므로 많은 메시지가 생성될 수 있습니다. z/OS 메시지 처리 기능을 사용하거나 **SET SYSTEM** 명령으로 **EXCLMSG** 를 설정하여 이러한 메시지를 억제할 수 있습니다.

IBM MQ 채널 정의 구성

많은 큐 관리자가 함께 연결된 경우 모든 오브젝트 정의를 관리하기가 어려울 수 있습니다. IBM MQ 클러스터링을 사용하면 관리가 간단해집니다.

전체 저장소로 두 개의 큐 관리자를 지정합니다. 다른 큐 관리자들은 저장소 중 하나에 대해 들어오는 연결 한 개와 나가는 연결 한 개가 필요합니다. 다른 큐 관리자로의 연결이 필요하다면 큐 관리자가 채널을 자동으로 작성하고 시작합니다.

클러스터에 많은 수의 큐 관리자를 보유할 계획인 경우 전용 저장소로 작동하고 애플리케이션 트래픽이 없는 큐 관리자를 보유하도록 계획해야 합니다.

자세한 정보는 [18 페이지의 『분산 큐와 클러스터 계획』](#)의 내용을 참조하십시오.

채널 시작기를 구성하기 전에 수행할 조치

1. TCP/IP를 사용할지 아니면 APPC를 사용할지 결정하십시오.
2. TCP를 사용하는 경우 IBM MQ에 하나 이상의 포트를 할당하십시오.
3. DNS 서버가 필요한 경우 필요에 따라 서버를 고가용성으로 구성하십시오.
4. APPC를 사용하는 경우 LU 이름을 할당하고 APPC를 구성하십시오.

채널 시작기를 구성한 후 프로덕션 환경으로 들어가기 전에 수행할 조치

1. 보유할 연결 계획:
 - a. 원격 애플리케이션에서 클라이언트 연결
 - b. 기타 큐 관리자로 들어오고 나가는 MCA 채널. 일반적으로 각 리모트 큐 관리자로 들어가고 나오는 채널을 보유합니다.
2. 클러스터링 설정 또는 기존 클러스터링 환경 결합.
3. 채널 시작기 앞에 가용성을 위해 다중 TCP 스택, VIPA 또는 외부 라우터를 사용할지 여부를 고려하십시오.
4. TLS 사용을 계획하는 경우:
 - a. 키 링 설정
 - b. 인증서 설정
5. 채널 인증을 계획하는 경우:
 - a. 인바운드 세션을 MCA 사용자 ID에 맵핑하기 위한 기준을 결정
 - b. 큐 관리자 매개변수 **REVDNS**를 설정하여 역방향 DNS 검색을 사용
 - c. 보안 검토. 예를 들어, 기본 채널을 삭제하고 채널에 대한 **MCAUSER** 속성에서 필수 권한만 가진 사용자 ID를 지정하십시오.
6. 채널 시작기에서 생성한 회계 및 통계 SMF 레코드를 캡처하고 이를 후처리
7. 작업 로그 메시지 모니터링 자동화.
8. 필요하면 처리량을 늘리기 위해 네트워크 환경을 조정. TCP를 사용하면 대량의 전송 및 수신 버퍼가 처리량을 개선합니다. 다음 명령을 사용하여 MQ가 특정 TCP 버퍼 크기를 사용하도록 강제 실행할 수 있습니다.

```
RECOVER QMGR(TUNE CHINTCPBDYNSZ nnnnn)  
RECOVER QMGR(TUNE CHINTCPSBDYNSZ nnnnn)
```

이는 채널에 대한 SO_RCVBUF 및 SO_SNDBUF를 지정된 바이트 크기(nnnnn)로 설정합니다.

관련 개념

[130 페이지의 『큐 관리자에 대한 계획』](#)

큐 관리자를 설정하는 경우 사용자의 계획이 큐 관리자의 확대를 허용해야 큐 관리자가 사용자 엔터프라이즈의 요구사항에 충족할 수 있습니다.

SMF 데이터에 대한 채널 시작기 계획

채널 시작기에 대한 SMF 데이터 수집 구현을 계획해야 합니다.

채널 시작기는 두 가지 유형의 레코드를 생성합니다.

- 채널 시작기 및 채널 시작기에 있는 태스크에 대한 정보가 포함된 통계 데이터.

- **DISPLAY CHSTATUS** 명령과 유사한 정보가 있는 채널 회계 데이터.

다음을 사용하여 통계 데이터 수집을 시작합니다.

```
START TRACE(STAT) CLASS(4)
```

다음 명령을 사용하여 중지하십시오.

```
STOP TRACE(STAT) CLASS(4)
```

다음을 사용하여 회계 데이터 수집을 시작합니다.

```
START TRACE(ACCTG) CLASS(4)
```

다음 명령을 사용하여 중지하십시오.

```
STOP TRACE(ACCTG) CLASS(4)
```

채널 정의 또는 큐 관리자의 **STATCHL** 속성을 사용하기 위해 수집된 회계 데이터가 있는 채널을 제어할 수 있습니다.

- 클라이언트 채널의 경우 큐 관리자 레벨에서 **STATCHL**을 설정해야 합니다.
- 자동으로 정의된 클러스터 송신자 채널의 경우, **STATACLS** 큐 관리자 속성이 있는 회계 데이터의 수집을 제어할 수 있습니다.

큐 관리자에 대한 **STATCHL**의 기본값은 OFF입니다. 채널 회계 데이터를 수집하려면 클래스 4의 계정 추적을 시작하는 것 외에도 큐 관리자 또는 채널 정의에서 기본값인 **STATCHL**의 값을 변경해야 합니다.

SMF 레코드는 다음과 같은 경우에 생성됩니다.

- **V9.3.0** IBM MQ for z/OS 9.3.0부터는 CSQ6SYSP **STATIME** 또는 **ACCTIME** 매개변수로 표시되는 시간 간격이 경과되었거나, SMF 데이터 수집 브로드캐스트에서 **STATIME** 또는 **ACCTIME**인 경우입니다. 채널 시작기 및 큐 관리자에 대한 SMF 데이터 수집 요청이 동기화됩니다.
- **STOP TRACE(ACCTG) CLASS(4)** 또는 **STOP TRACE(STAT) CLASS(4)** 명령이 실행되었습니다. 또는
- 채널 시작기가 시스템 종료된 경우. 이 시점에서 모든 SMF가 기록됩니다.

채널이 SMF 간격 중에 중지되면, 회계 데이터는 다음에 SMF 처리가 실행될 때 SMF에 기록됩니다. 클라이언트가 연결되어 일부 작업을 실행하고 연결이 끊어진 후에 다시 연결되고 또 다시 연결이 끊어지면 두 세트의 채널 회계 데이터가 생성됩니다.

통계 SMF 데이터는 일반적으로 하나의 SMF 레코드에 맞지만 많은 수의 태스크가 사용 중인 경우에는 여러 개의 SMF 레코드가 작성될 수 있습니다.

회계 데이터는 사용으로 설정된 각 채널에 대해 수집되며 일반적으로 하나의 SMF 레코드에 맞습니다. 그렇지만 많은 수의 태스크가 사용 중인 경우에는 여러 개의 SMF 레코드가 작성될 수 있습니다.

채널 시작기 SMF 데이터의 수집 비용은 적습니다. 일반적으로 CPU 사용량 증가는 적은 퍼센트이며 측정 오류 내이기도 합니다.

이 함수를 사용하기 전에 z/OS 시스템 프로그래머와 같이 작업하여 SMF에 추가 레코드에 대한 용량이 확보되고 새 SMF 데이터를 포함하기 위해 SMF 레코드를 추출하는 프로세스를 변경하도록 해야 합니다.

채널 시작기 통계 데이터의 경우, SMF 레코드 유형은 115 및 하위유형 231입니다.

채널 시작기 회계 데이터의 경우, SMF 레코드 유형은 116 및 하위유형 10입니다.

이 데이터를 처리하는 자체 프로그램을 작성하거나 데이터를 인쇄하고 스프레드시트로 가져오기에 적합한 CSV(쉼표로 구분되는 값)의 데이터를 작성하는 프로그램 MQSMI이 포함된 SupportPac **MP1B**를 사용할 수 있습니다.

채널 시작기 SMF 데이터를 캡처할 때 문제가 발생하면 자세한 정보는 [채널 시작기에 대해 SMF 데이터를 캡처할 때 발생하는 문제 처리\(CHINIT\)](#)를 참조하십시오.

관련 태스크

[IBM MQ 성능 통계 해석](#)

[채널 계정 데이터 문제점 해결](#)

z/OS z/OS TCP/IP 환경 계획

네트워크에서 최고의 처리량을 가져오려면 64KB 이상의 TCP/IP 송수신 버퍼를 사용해야 합니다. 이 크기로 시스템은 해당 버퍼 크기를 최적화합니다.

[대기 시간이 긴 네트워크에 대한 동적 적정 크기 조정이란 무엇입니까?](#) 참조 참조하십시오.

다음 Netstat 명령으로 시스템 버퍼 크기를 검사할 수 있습니다. 예:

```
TSO NETSTAT ALL (CLIENT csq1CHIN
```

결과는 다음 두 값을 포함하여 많은 정보를 표시합니다.

```
ReceiveBufferSize: 0000065536  
SendBufferSize: 0000065536
```

65536은 64KB입니다. 버퍼 크기가 65536미만인 경우 네트워크 팀과 같이 TCP/IP 프로시저의 PROFILE DDName에서 **TCPSENDBFRSIZE** 및 **TCPRCVBUFRSIZE** 값을 늘리기 위해 작업해야 합니다. 예를 들어, 다음 명령을 사용할 수 있습니다.

```
TCPCONFIG TCPSENDBFRSIZE 65536 TCPRCVBUFRSIZE 65536
```

시스템 전반의 **TCPSENDBFRSIZE** 또는 **TCPRCVBUFRSIZE** 설정을 변경할 수 없는 경우에는 IBM 소프트웨어 지원 센터에 문의하십시오.

z/OS 큐 공유 그룹(QSG) 계획

공유 큐잉 환경을 구현하는 가장 쉬운 방법은 큐 관리자를 구성하고 해당 큐 관리자를 QSG에 추가한 다음 다른 큐 관리자를 QSG에 추가하는 것입니다.

큐 공유 그룹은 Db2 테이블을 사용하여 구성 정보를 저장합니다. 동일한 Db2 데이터 공유 그룹을 공유하는 모든 QSG에서 사용하는 하나의 테이블 세트가 있습니다.

공유 큐 메시지는 커플링 기능(CF)의 구조에 저장됩니다. 각 QSG에는 CF 구조의 자체 세트가 있습니다. 사용자의 요구사항에 맞게 구조를 구성해야 합니다.

크기가 63KB 를 초과하는 메시지는 CF에 저장할 수 없습니다. 이러한 메시지에 대해 공유 메시지 데이터 세트(SMDS) 또는 Db2 를 사용해야 합니다.

메시지 프로파일 및 용량 계획

공유 큐 메시지의 메시지 프로파일을 이해해야 합니다. 다음은 고려해야 할 요소의 예입니다.

- 평균 및 최대 메시지 크기
- 일반 큐 용량 및 예외 큐 용량. 예를 들어 하루 전체 동안 메시지를 보유할 수 있는 충분한 용량이 필요하고 일반 큐 용량이 100개 메시지 미만입니다.

메시지 프로파일이 변경되면 구조의 크기를 늘리거나 나중에 SMDS를 구현할 수 있습니다.

메시지의 대형 최대 볼륨을 처리할 수 있게 하려면 구조의 사용량이 지정된 임계값에 도달할 때 메시지를 SMDS 로 오프로드하도록 IBM MQ를 구성할 수 있습니다.

CF 구조를 이중화하려는지를 결정하십시오. 이는 CFRM 정책의 CF 구조 정의에 의해 제어됩니다.

1. 이중화된 구조는 두 개의 커플링 기능을 사용합니다. 하나의 CF에 문제점이 있어도 서비스 인터럽트가 없으며 사용 가능한 세 번째 CF에서 구조를 다시 빌드할 수 있습니다. 중복된 구조는 공유 큐의 조작 성능에 상당한 영향을 줄 수 있습니다.
2. 구조가 이중화되지 않은 경우 CF에 문제가 있다는 것은 구조를 다른 CF에서 재빌드할 수 있을 때까지 해당 CF의 구조에 있는 공유 큐를 사용할 수 없음을 의미합니다.

이 경우 IBM MQ는 다른 CF에서 자동으로 구조를 재빌드하도록 구성될 수 있습니다. 지속 메시지가 큐 관리자의 로그에서 복구됩니다.

CF 정의를 변경하는 것은 용이함을 참고하십시오.

비지속 메시지만 포함하도록 또는 지속 및 비지속 메시지 둘 다를 포함하도록 구조를 정의할 수 있습니다.

지속 메시지를 보유하는 구조는 정기적으로 백업되어야 합니다. 장애 시 구조의 복구에 필요한 시간을 최소화하도록 최소한 메시지 CF 구조를 백업하십시오. 백업은 백업을 수행하는 큐 관리자의 로그 데이터 세트에 저장됩니다.

공유 큐에서 메시지의 높은 처리량을 기대하는 경우 CF 구조를 백업하기 위한 전용 큐 관리자를 보유하는 것이 가장 좋습니다. 이렇게 하면 큐 관리자 로그에서 읽어야 하는 데이터가 적어지므로 구조를 복구하는 데 필요한 시간이 줄어듭니다.

채널

IBM MQ QSG에 연결된 애플리케이션에 대해 단일 시스템 이미지를 제공하려면 공유 입력 채널을 정의할 수 있습니다. 이를 설정하면, 큐 공유 그룹 환경으로 들어오는 연결이 QSG의 어떤 큐 관리자라도 이동할 수 있습니다.

이러한 채널을 위해 네트워크 라우터 또는 가상 IP 주소(VIPA)를 설정해야 할 수 있습니다.

공유 출력 채널을 정의할 수 있습니다. 공유 출력 채널 인스턴스는 QSG의 모든 큐 관리자에서 시작될 수 있습니다.

자세한 정보는 [공유 채널](#)을 참조하십시오.

보안

외부 보안 관리자를 사용하여 IBM MQ 자원을 보호합니다. RACF®를 사용하는 경우 RACF 프로파일의 접두부는 큐 관리자 이름입니다. 예를 들어, APPLICATION.INPUT은 qmgrName.APPLICATION.INPUT. 라는 MQQUEUE 클래스의 프로파일을 사용하여 보호됩니다.

큐 공유 그룹을 사용할 때 큐 관리자 이름의 접두부가 있는 프로파일로 자원을 계속 보호하거나 큐 공유 그룹 이름을 사용하여 프로파일의 접두부로 지정할 수 있습니다. 예를 들어 qsgName.APPLICATION.INPUT.

큐 공유 그룹 이름의 접두부가 있는 프로파일을 사용해야 하는데, 이는 모든 큐 관리자에 대해 단일 정의가 있으므로 작업을 줄여주고 큐 관리자 간 정의의 불일치를 방지하기 때문입니다.

관련 개념

130 페이지의 『[큐 관리자에 대한 계획](#)』

큐 관리자를 설정하는 경우 사용자의 계획이 큐 관리자의 확대를 허용해야 큐 관리자가 사용자 엔터프라이즈의 요구사항에 충족할 수 있습니다.

커플링 기능 및 오프로드 스토리지 환경 계획

커플링 기능(CF) 구조, 공유 메시지 데이터 세트(SMDS) 환경 또는 Db2 환경의 형식과 초기 크기를 계획할 때 이 주제를 사용하십시오.

이 절에는 다음 주제에 대한 정보가 포함되어 있습니다.

- [161 페이지의 『커플링 기능 자원 정의』](#)
 - [오프로드 스토리지 메커니즘 결정](#)
 - [구조 계획](#)
 - [사용자 구조의 크기 계획](#)

- 공유 큐를 구조로 맵핑
- 166 페이지의 『공유 메시지 데이터 세트(SMDS) 환경 계획』
- 169 페이지의 『Db2 환경 계획』

커플링 기능 자원 정의

공유 큐를 사용하려면 IBM MQ가 CFRM 정책에 사용할 커플링 기능 구조를 정의해야 합니다. 이를 수행하려면 먼저 구조에 대한 정보로 CFRM 정책을 업데이트한 후 정책을 활성화해야 합니다.

설치에 사용 가능한 커플링 기능을 설명하는 기존 CFRM 정책이 있을 수 있습니다. 관리 데이터 유틸리티는 사용자가 제공한 텍스트 설명을 기반으로 정책의 콘텐츠를 수정하는 데 사용됩니다. 새 구조의 이름, 구조가 정의된 커플링 기능, 구조의 크기를 정의하는 정책에 명령문을 추가해야 합니다.

CFRM 정책은 IBM MQ 구조가 이중화되는지 여부와 실패 시나리오에 재할당되는 방식도 결정합니다. 공유 큐 복구에는 커플링 기능에 영향을 미치는 실패에 대한 복원성을 제공하도록 CFRM을 구성하기 위한 권장사항이 포함되어 있습니다.

오프로드 스토리지 환경 결정

공유 큐에 대한 메시지 데이터는 커플링 기능에서 오프로드되어 Db2 테이블 또는 공유 메시지 데이터 세트(SMDS) 라는 IBM MQ 관리 데이터 세트에 저장될 수 있습니다. 너무 커서(즉, 63KB 초과) 커플링 기능에 저장할 수 없는 메시지는 항상 오프로드해야 하고 더 작은 메시지는 선택적으로 오프로드하여 커플링 기능 공간 사용을 줄일 수 있습니다.

자세한 정보는 [공유 메시지에 대한 오프로드 옵션 지정](#)을 참조하십시오.

구조 계획

큐 공유 그룹(QSG)에는 최소 2개의 구조를 정의해야 합니다. 관리 구조로 알려진 첫 번째 구조는 큐 공유 그룹에서 IBM MQ 내부 활동을 조정하는 데 사용됩니다. 이 구조는 사용자 데이터를 보유하지 않습니다. 고정 이름 *qsg-name* CSQ_ADMIN(여기서, *qsg-name*은 큐 공유 그룹의 이름)이 있습니다. 후속 구조를 애플리케이션 구조라고 하며 IBM MQ 공유 큐에 메시지를 보유하는 데 사용됩니다. 각 구조에 최대 512개의 공유 큐를 보유할 수 있습니다.

*qsg-name*CSQSYSAPPL이라는 애플리케이션 구조는 시스템 큐에 사용됩니다. 이 구조를 정의하는 것은 선택사항이지만, 특정 기능을 사용하는 데 필요합니다. 기본적으로 SYSTEM.QSG.CHANNEL.SYNCQ 및 SYSTEM.QSG.UR.RESOLUTION.QUEUE 큐는 *qsg-name*CSQSYAPPL 구조에 정의됩니다.

다중 구조 사용

큐 공유 그룹은 최대 64개의 커플링 기능 구조에 연결할 수 있습니다. 이러한 구조 중 하나는 관리 구조여야 합니다. 관리 구조가 정의된 경우 이러한 구조 중 다른 구조는 *qsg-name*CSQSYSAPPL 구조일 수 있습니다. 메시지 데이터에 최대 63개(*qsg-name*CSQSYSAPPL이 정의된 경우 62개)를 사용할 수 있습니다. 다음과 같은 이유로 다수의 애플리케이션 구조를 사용하도록 선택할 수 있습니다.

- 많은 수의 메시지를 보유할 수 있는 몇 개의 큐가 있으므로 전체 커플링 기능의 모든 자원이 필요합니다.
- 많은 수의 공유 큐에 대한 요구사항이 있으며, 각 구조가 512개의 큐만 포함할 수 있으므로 다수의 구조로 큐를 분할해야 합니다.
- 구조의 사용 특성에 관한 RMF 보고서에는 구조에 포함된 큐를 많은 커플링 기능에 분산시키도록 제안되어 있습니다.
- 데이터 분리 이유로 다른 큐 데이터와 물리적으로 다른 커플링 기능에 일부 큐 데이터를 보유하려고 합니다.
- 지속 공유 메시지의 복구는 구조 레벨 속성 및 명령(예: BACKUP CFSTRUCT)을 사용하여 수행됩니다. 백업 및 복구를 단순화하기 위해 지속 메시지를 보유하는 구조와 다른 구조에 비지속 메시지를 보유하는 큐를 지정할 수 있습니다.

구조를 할당할 커플링 기능을 선택하는 경우 다음 사항을 고려하십시오.

- 데이터 분리 요구사항.
- 커플링 기능의 휘발성(즉, 정전 시 데이터를 보존하는 능력).
- 커플링 기능 간 또는 커플링 기능과 액세스하는 시스템 간의 실패 독립성.
- 커플링 기능에 설치된 커플링 기능 제어 코드(CFCC)의 레벨(IBM MQ에는 레벨 9 이상이 필요함).

구조의 크기 계획

관리 구조

관리 구조(*qsg-name* CSQ_ADMIN)는 큐 공유 그룹의 각 큐 관리자에 대한 1000개의 목록 항목을 포함할 정도로 커야 합니다. 큐 관리자가 시작되면 큐 공유 그룹에 현재 정의된 큐 관리자 수에 맞게 구조가 충분히 큰지 확인합니다. 큐 관리자는 CSQ5PQSG 유틸리티로 추가된 경우 큐 공유 그룹에 정의된 것으로 간주됩니다. `MQSC DISPLAY GROUP` 명령으로 그룹에 정의된 큐 관리자를 확인할 수 있습니다.

참고: 구조의 크기를 계산할 때 큐 공유 그룹의 큐 관리자 수 외에도 대형 작업 단위의 크기를 허용해야 합니다.

162 페이지의 표 22에는 큐 공유 그룹에 정의된 다양한 수의 큐 관리자에 대한 관리 구조의 최소 필수 크기가 표시되어 있습니다. 이 크기는 CFCC 레벨 14 커플링 기능 구조에 대해 설정되며 더 높은 레벨의 CFCC에서는 더 커야 할 수 있습니다.

큐 공유 그룹에 정의된 큐 관리자 수	필수 스토리지
1	6144KB
2	6912KB
3	7976KB
4	8704KB
5	9728KB
6	10496KB
7	11520KB
8	12288KB
9	13056KB
10	14080KB
11	14848KB
12	15616KB
13	16640KB
14	17408KB
15	18176KB
16	19200KB
17	19968KB
18	20736KB
19	21760KB
20	22528KB

표 22. 최소 관리 구조 크기 (계속)	
큐 공유 그룹에 정의된 큐 관리자 수	필수 스토리지
21	23296KB
22	24320KB
23	25088KB
24	25856KB
25GB	27136KB
26	27904KB
27	28672KB
20	29696KB
29	30464KB
30	31232KB
31	32256KB

기존 큐 공유 그룹에 큐 관리자를 추가할 때에는 스토리지 요구사항이 162 페이지의 표 22에 권장된 크기를 초과하여 증가할 수 있습니다. 그럴 경우 다음 프로시저를 수행하여 `qsg-nameCSQ_ADMIN` 구조에 필요한 스토리지를 추정하십시오.

1. 큐 공유 그룹의 기존 멤버에서 MQSC 명령 **DISPLAY CFSTATUS(CSQ_ADMIN)**를 실행하십시오.
2. CSQ_ADMIN 구조에 대한 ENTSMAX 정보를 추출하십시오.
3. 이 수치가 1000회 미만이면 큐 공유 그룹에서 정의하려는 큐 관리자의 총 수가 구조 크기를 늘립니다.

애플리케이션 구조

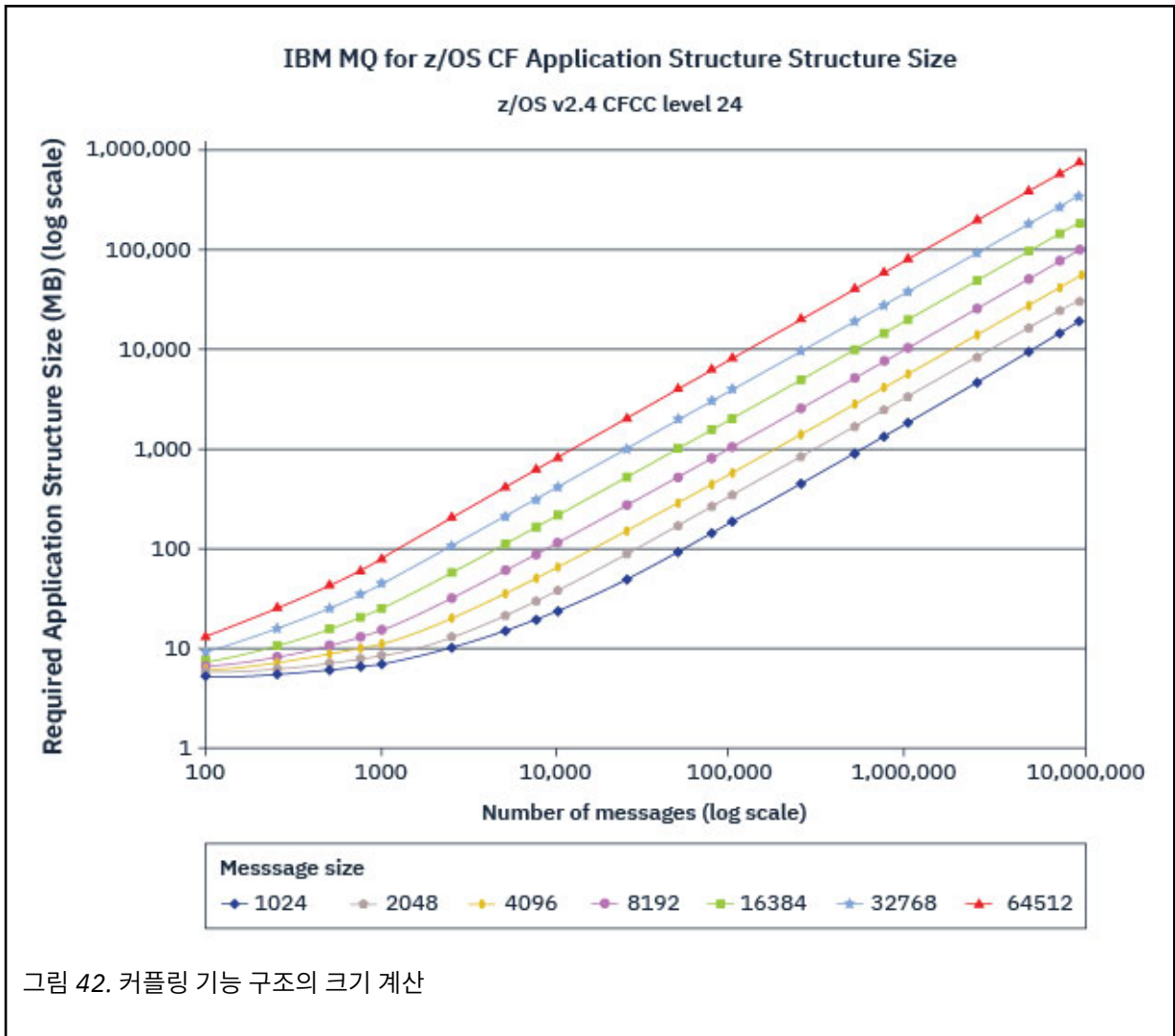
IBM MQ 메시지를 보유하는 데 필요한 구조의 크기는 애플리케이션 구조에 동시에 보유할 메시지의 가능한 수 및 크기에 따라 다릅니다.

164 페이지의 그림 42의 그래프는 공유 큐에 메시지를 보유하기 위해 작성해야 하는 CF 구조의 크기를 보여줍니다. 할당 크기를 계산하려면 다음 정보가 필요합니다.

- 큐의 평균 메시지 크기
- 구조에 저장할 총 메시지 수

가로축을 따라 메시지 수를 찾으십시오. 메시지 크기에 해당하는 곡선을 선택하고 세로축에서 필요한 값을 결정하십시오. 예를 들어, 길이가 1KB인 200 000개 메시지는 값이 256 - 512MB 범위입니다.

164 페이지의 표 23에는 동일한 정보가 표 형식으로 제공됩니다.



커플링 기능 구조의 크기를 계산하려면 다음 표를 사용하십시오.

표 23. 커플링 기능 구조의 크기 계산

메시지 수	1KB	2KB	4KB	8KB	16KB	32KB	63KB
100	6MB	6MB	7MB	7MB	8MB	10MB	14MB
1000	8MB	9MB	12MB	17MB	27MB	48MB	88MB
10000	25MB	38MB	64MB	115MB	218MB	423MB	821MB
100000	199MB	327MB	584MB	1097MB	2124MB	4177MB	8156MB

CFRM 정책은 다음 명령문을 포함해야 합니다.

- INITSIZE는 첫 번째 커넥터가 연결될 때 구조에 할당된 크기(KB)입니다.
- SIZE는 구조의 가능한 최대 크기입니다.
- FULLTHRESHOLD는 구조가 가득 찼음을 알리는 메시지 IXC585E를 z/OS가 발행하는 임계값의 백분율 값을 설정합니다.

우수 사례는 INITSIZE 및 SIZE가 인수 2 이내가 되도록 하는 것입니다. 예를 들어, 앞에서 판별된 수치를 사용하여 다음 명령문을 포함할 수 있습니다.

```
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)
```

구조 사용이 경고 메시지를 발행하는 임계값에 도달하면 개입이 필요합니다. IBM MQ를 사용하여 구조의 일부 큐에 대한 MQPUT 조작을 금지해서 애플리케이션이 더 많은 메시지를 쓰지 못하게 하거나, 추가 애플리케이션을 시작하여 큐에서 더 많은 메시지를 가져오거나, 큐에 메시지를 넣고 있는 일부 애플리케이션을 일시 정지할 수 있습니다.

또는 z/OS 기능을 사용하여 해당 위치의 구조 크기를 변경할 수 있습니다. 다음 z/OS 명령은

```
SETXCF START,ALTER,STRNAME=structure-name,SIZE=newsize
```

구조의 크기를 *newsize*로 변경합니다. *newsize*는 구조에 대한 CFRM 정책에 지정된 SIZE 값 미만이고 현재 커플링 기능 크기보다는 큼니다.

MQSC `DISPLAY CFSTATUS` 명령으로 커플링 기능 구조 사용을 모니터링할 수 있습니다.

조치가 수행되지 않고 큐 구조가 채워지면 MQRC_STORAGE_MEDIUM_FULL 리턴 코드가 애플리케이션에 리턴됩니다. 관리 구조가 가득 차면 정확한 증상은 오류가 발생한 프로세스에 따라 다르지만 다음 문제점을 포함할 수 있습니다.

- 명령에 대한 응답이 없습니다.
- 커밋 처리 중의 문제로 인해 큐 관리자가 실패합니다.

CSQSYSAPPL 구조

*qsg-name*CSQSYSAPPL 구조는 시스템 큐의 애플리케이션 구조입니다. 표 3은 *qsg-name*CSQSYSAPPL 구조에 정의된 기본 큐의 메시지 데이터 크기를 추정하는 방법에 대한 예를 보여줍니다.

표 24. 크기 조절에 대한 CSQSYSAPPL 사용이 표시된 표.	
<i>qsg-name</i> CSQSYSAPPL 사용	크기
SYSTEM.QSG.CHANNEL.SYNCCQ	공유 채널의 활성 인스턴스당 500바이트의 2개 메시지
SYSTEM.QSG.UR.RESOLUTION.QUEUE	2KB의 1000개 메시지

제안된 초기 구조 정의 값은 다음과 같습니다.

```
STRUCTURE NAME(qsg-nameCSQSYSAPPL)
INITSIZE(20480) /* 20 MB */
SIZE(30720) /* 30 MB */
FULLTHRESHOLD(85)
```

이 값은 공유 채널 및 그룹 복구 단위의 사용에 따라 조정할 수 있습니다.

공유 큐를 구조로 맵핑

IBM MQ에 대한 애플리케이션 구조를 정의하려면 `DEFINE CFSTRUCT` 명령을 사용하십시오. IBM MQ에 대한 구조를 정의할 때 QSG 이름 접두부를 구조 이름에 포함시키지 마십시오. 예를 들어, CFRM 정책에서 이름이 *qsg-name*APPLICATION1인 IBM MQ에 애플리케이션 구조를 정의하려면 다음 명령을 실행하십시오.

```
DEFINE CFSTRUCT(APPLICATION1)
```

큐 정의의 CFSTRUCT 속성은 큐를 구조로 맵핑하는 데 사용됩니다. 이 속성에 QSG 이름 접두부 없이 CF 구조의 이름을 지정하십시오. 예를 들어 다음 명령은 APPLICATION1 구조에 공유 큐를 정의합니다.

```
DEFINE QLOCAL(myqueue) QSGDISP(SHARED) CFSTRUCT(APPLICATION1)
```

공유 메시지 데이터 세트(SMDS) 환경 계획

SMDS 오프로딩으로 큐 공유 그룹을 사용 중인 경우 IBM MQ는 공유 메시지 데이터 세트 그룹에 연결해야 합니다. 이 주제를 사용하여 데이터 세트 요구사항 및 IBM MQ 메시지 데이터를 저장하기 위해 필요한 구성을 이해하십시오.

공유 메시지 데이터 세트(키워드 SMDS로 설명됨)는 커플링 기능 구조에 저장된 공유 메시지의 오프로드된 메시지 데이터를 저장하기 위해 큐 관리자에 사용되는 데이터 세트입니다.

참고: 구조에 대한 SMDS 데이터 세트를 정의 할 때 각 큐 관리자에 SMDS 데이터 세트가 있어야 합니다.

이러한 형식의 데이터 오프로딩이 사용될 때에는 **CFSTRUCT**에 큐 공유 그룹의 각 큐 관리자마다 하나의 데이터 세트가 있는, 공유 메시지 데이터 세트의 연관된 그룹이 필요합니다. 공유 메시지 데이터 세트 그룹은 **CFSTRUCT** 정의에서 **DSGROUP** 매개변수를 사용하여 IBM MQ에 정의됩니다. 추가 매개변수를 사용하여 데이터 세트의 확장 속성 및 사용할 버퍼 수와 같은 선택적 추가 정보를 제공할 수 있습니다.

각 큐 관리자는 소유한 데이터 세트에 쓰고, 큐 관리자를 통해 기록된 메시지의 공유 메시지 데이터를 저장하며, 그룹의 모든 데이터 세트를 읽을 수 있습니다.

구조와 연관된 각 데이터 세트에 대한 상태 및 속성을 설명하는 목록은 **CFSTRUCT** 정의의 일부로 내부적으로 유지보수되므로 각 큐 관리자는 정의를 확인하여 현재 사용 가능한 데이터 세트를 찾을 수 있습니다.

이 데이터 세트 정보는 **DISPLAY CFSTATUS TYPE(SMDS)** 명령을 사용하여 현재 상태 및 가용성을 표시하고 **DISPLAY SMDS** 명령을 사용하여 지정된 **CFSTRUCT**와 연관된 데이터 세트의 매개변수 설정을 표시하여 표시할 수 있습니다.

개별 공유 메시지 데이터 세트는 소유한 큐 관리자 이름(일반적으로 **SMDS** 키워드를 사용하여 지정)과 **CFSTRUCT** 구조 이름을 조합하여 효과적으로 식별합니다.

이 절은 다음 주제를 설명합니다.

- [DSGROUP 매개변수](#)
- [DSBLOCK 매개변수](#)
- [공유 메시지 데이터 세트 특성](#)
- [공유 메시지 데이터 세트 공간 관리](#)
- [공유 메시지 데이터 세트에 대한 액세스](#)
- [공유 메시지 데이터 세트 작성](#)
- [공유 메시지 데이터 세트 성능 및 용량 고려사항](#)
- [공유 메시지 데이터 세트 활성화](#)

이 매개변수에 대한 자세한 내용은 [DEFINE CFSTRUCT](#)를 참조하십시오.

공유 메시지 데이터 세트 관리에 대한 정보는 [공유 메시지 데이터 세트 관리](#)를 참조하십시오.

DSGROUP 매개변수

CFSTRUCT 정의의 **DSGROUP** 매개변수는 해당 구조에 대한 큰 메시지를 저장할 데이터 세트의 그룹을 식별합니다. 추가 매개변수를 사용하여 공간 할당 용도로 사용할 논리 블록 크기 및 버퍼 풀 크기와 자동 데이터 세트 확장 옵션의 값을 지정할 수 있습니다.

데이터 세트에 대한 오프로딩을 사용 가능하게 설정하려면 먼저 **DSGROUP** 매개변수를 설정해야 합니다.

- 새 **CFSTRUCT**가 **CFLEVEL (5)**에 정의되고 있고 옵션 **OFFLOAD(SMDS)**가 지정 또는 가정되면 **DSGROUP** 매개변수를 동일한 명령에 지정해야 합니다.

- CFLEVEL을 CFLEVEL (5)로 늘리기 위해 기존 CFSTRUCT를 변경하고 있으며 옵션 OFFLOAD(SMDS)가 지정 또는 가정되면 아직 설정되지 않은 경우 DSGROUP 매개변수를 동일한 명령에 지정해야 합니다.

DSBLOCK 매개변수

각 데이터 세트 내 공간은 CFSTRUCT 정의의 DSBLOCK 매개변수를 사용하여 지정된 고정 크기(보통 256KB)의 논리 블록으로 큐에 할당되며, 각 논리 블록 내 4KB(실제 블록 크기 및 제어 간격 크기에 해당) 페이지 범위로 개별 메시지에 할당됩니다. 또한 논리 블록 크기는 SMDS 버퍼 풀의 버퍼 크기와 동일한, 단일 I/O에서 읽거나 쓸 수 있는 메시지 데이터의 최대 크기도 판별합니다.

DSBLOCK 매개변수의 값이 더 커지면 개별 I/O 조작의 수를 줄여 매우 큰 메시지에 대한 성능이 개선될 수 있습니다. 하지만 값이 더 작으면 각 활성화 요청에 필요한 버퍼 스토리지의 크기가 감소합니다. DSBLOCK 매개변수의 기본값은 256KB이며 요구사항 간의 합리적인 밸런스를 제공하므로 대개 이 매개변수는 필요하지 않을 수도 있습니다.

공유 메시지 데이터 세트 특성

공유 메시지 데이터 세트는 VSAM 선형 데이터 세트(LDS)로 정의됩니다. 각 오프로드된 메시지는 데이터 세트의 하나 이상의 블록에 저장됩니다. 저장된 데이터는 가상 스토리지의 확장 형식과 같은, 커플링 기능 항목의 정보로 직접 처리됩니다. 데이터 세트 자체에 저장된 별도의 색인이나 유사한 제어 정보는 없습니다.

직접 주소 지정 설계는 한 블록에 맞는 메시지의 경우 블록을 읽거나 쓰기 위해 단일 I/O 조작만이 필요함을 의미합니다. 메시지의 범위가 둘 이상의 블록인 경우 충분한 버퍼가 사용 가능하면 경과 시간을 최소화하기 위해 각 블록에 대한 I/O 조작을 완전히 중첩시킬 수 있습니다.

공유 메시지 데이터 세트는 소수의 일반 제어 정보도 포함합니다. 이 정보는 복구 및 재시작 상태 정보를 포함한 첫 번째 페이지의 헤더와 큐 관리자 정상 종료 시 사용 가능한 블록 공간 맵을 저장하는 데 사용되는 공간 맵 체크 포인트 영역으로 구성됩니다.

공유 메시지 데이터 세트 공간 관리

용량, 성능, 작동 고려사항에 대한 백그라운드 정보로서 큐 관리자가 공유 메시지 데이터 세트의 공간을 관리하는 방식에 대한 개념을 이해하는 데 유용할 수 있습니다.

각 공유 메시지 데이터 세트의 여유 공간은 공간을 소유한 큐 관리자가 각 논리 블록 내에 사용 중인 페이지 수를 나타내는 공간 맵을 사용하여 추적합니다. 공간 맵은 데이터 세트가 열려 있는 동안 주 기억장치에서 유지보수되고 정상적으로 닫힌 경우 데이터 세트에 저장됩니다. (복구 상황에서는, 현재 사용 중인 데이터 세트 페이지를 찾기 위해 커플링 기능 구조에서 메시지를 스캔하여 공간 맵이 자동으로 다시 빌드됩니다.)

오프로드된 메시지 데이터가 있는 공유 메시지가 기록되면 큐 관리자는 각 메시지 블록의 페이지 범위를 할당합니다. 지정된 큐에 대해 부분적으로 사용된 현재 논리 블록이 있으면 큐 관리자는 이 블록의 사용 가능한 다음 페이지에서 시작하는 공간을 할당하고, 그렇지 않은 경우에는 새 논리 블록을 할당합니다. 전체 메시지가 현재 논리 블록 내에 맞지 않으면 큐 관리자는 논리 블록 끝에서 메시지 데이터를 분할하고 다음 메시지 블록에 대해 새 논리 블록을 할당합니다. 전체 메시지에 대한 공간이 할당될 때까지 이 과정을 반복합니다. 마지막 논리 블록의 사용되지 않은 공간은 큐의 새 현재 논리 블록으로 저장됩니다. 데이터 세트가 정상적으로 닫히면 현재 논리 블록의 사용되지 않은 페이지가 저장되기 전에 공간 맵으로 리턴됩니다.

오프로드된 메시지 데이터가 있는 공유 메시지를 읽은 후 삭제할 준비가 되면 큐 관리자는 소유하고 있는 큐 관리자(동일한 큐 관리자일 수 있음)가 모니터링하는 정리 목록으로 메시지에 대한 커플링 기능 항목을 전송하여 삭제 요청을 처리합니다. 항목이 이 목록에서 도달하면 소유하고 있는 큐 관리자가 항목을 읽고 삭제한 후 해제된 페이지 범위를 공간 맵으로 리턴합니다. 논리 블록의 사용된 모든 페이지가 해제되면 블록을 재사용할 수 있습니다.

공유 메시지 데이터 세트에 대한 액세스

각 공유 메시지 데이터 세트는 큐 공유 그룹의 모든 큐 관리자에 액세스 가능한 공유 직접 액세스 스토리지에 있어야 합니다.

정상 실행 중, 각 큐 관리자는 읽기/쓰기 액세스를 위해 자체 공유 메시지 데이터 세트를 열고 읽기 전용 액세스를 위해서는 다른 큐 관리자의 활성화 공유 메시지 데이터 세트를 엽니다. 그러면 해당 큐 관리자가 저장한 메시지를 읽을 수 있습니다. 이는 각 큐 관리자 사용자 ID에 자체 공유 메시지 데이터 세트에 대한 최소 UPDATE 액세스 및 구조의 다른 모든 공유 메시지 데이터 세트에 대한 READ 액세스가 필요함을 의미합니다.

RECOVER CFSTRUCT를 사용하여 공유 메시지 데이터 세트를 복구해야 하는 경우 큐 공유 그룹의 어느 큐 관리자에서나 복구 프로세스를 실행할 수 있습니다. 복구 처리를 수행하는 데 사용할 수 있는 큐 관리자에는 복구해야 하는 모든 데이터 세트에 대한 UPDATE 액세스가 필요합니다.

공유 메시지 데이터 세트 작성

이러한 메시지 오프로딩 양식을 사용할 수 있도록 해당 **CFSTRUCT** 정의를 작성 또는 변경하기 전에 각 공유 메시지 데이터 세트를 정상적으로 작성해야 하며, 이는 **CFSTRUCT** 정의 변경이 일반적으로 즉시 적용되어 큐 관리자가 해당 구조에 지정된 공유 큐에 액세스하려 시도하는 즉시 데이터 세트가 필요하기 때문입니다. 공유 메시지 데이터 세트를 할당하고 사전 형식화하는 샘플 작업은 **SCSQPROC(CSQ4SMDS)**에 제공됩니다. **OFFLOAD(SMDS)**를 통해 **CFSTRUCT**를 사용하는 각 큐 관리자마다 공유 메시지 데이터 세트를 할당하도록 작업을 사용자 정의하고 실행해야 합니다.

큐 관리자가 오프로드 지원이 사용 가능함을 확인하고 공유 메시지 데이터 세트를 열려고 시도하지만 이 데이터 세트가 아직 작성되지 않은 경우에는 공유 메시지 데이터 세트에 사용 불가능 플래그가 지정됩니다. 그러면 큐 관리자는 데이터 세트가 작성되어 큐 관리자에게 다시 시도하도록 통지될 때까지(예를 들어, **START SMDSCONN** 명령을 사용하여) 큰 메시지를 저장할 수 없습니다.

공유 메시지 데이터 세트는 액세스 메소드 서비스 **DEFINE CLUSTER** 명령을 사용하여 **VSAM** 선형 데이터 세트로 작성됩니다. 정의는 **SHAREOPTIONS(2 3)**를 지정하여 하나의 큐 관리자가 쓰기 액세스를 위해 이를 열고 임의의 수의 큐 관리자가 동시에 이를 읽을 수 있도록 허용해야 합니다. 기본 제어 간격 크기 4KB를 사용해야 합니다. 4GB를 초과하여 데이터 세트를 확장해야 하는 경우에는 **VSAM** 확장 주소 지정 속성이 있는 **SMS** 데이터 래스를 사용하여 데이터 세트를 정의해야 합니다. 공유 메시지 데이터 세트는 **EAV (Extended Address Volume)**의 **EAS (Extended Address Space)** 파트에 상주할 수 있습니다.

각 공유 메시지 데이터 세트는 처음에 사용하기 전에 비어 있거나 2진 0으로 사전에 형식화(**CSQJUFMT** 또는 샘플 작업 **SCSQPROC(CSQ4SMDS)**와 유사한 유틸리티를 사용하여)되어 있을 수 있습니다. 열린 경우 부분적으로만 형식화되어 있거나 비어 있으면 큐 관리자가 나머지 공간을 2진 0으로 자동 형식화합니다.

공유 메시지 데이터 세트 성능 및 용량 고려사항

각 공유 메시지 데이터 세트는 동일한 시스템 내의 영역에서 소유하고 있는 큐 관리자를 통해 연관된 **CFSTRUCT**에 기록되어 있는 공유 메시지의 오프로드된 데이터를 저장하는 데 사용됩니다. 오프로드된 각 메시지는 최대 768바이트의 **CF** 스토리지를 차지하며, 이는 항목에 대해 256바이트, 헤더 및 디스크립터의 두 요소에 대해 512바이트로 구성됩니다. 각 오프로드된 메시지는 데이터 세트에서 하나 이상의 페이지(4KB 크기의 물리적 블록)에 저장됩니다.

따라서 주어진 수의 오프로드된 메시지에 필요한 데이터 세트 공간은 전체 메시지 크기(디스크립터 포함)를 4KB의 다음 배수로 반올림한 후 메시지 수를 곱하여 추정할 수 있습니다.

페이지 세트의 경우 공유 메시지 데이터 세트가 거의 다 차면 선택적으로 자동 확장될 수 있습니다. 이 자동 확장의 기본 작동은 **CFSTRUCT** 정의의 **DSEXPAND** 매개변수를 사용하여 설정할 수 있습니다. **ALTER SMDS** 명령에 **DSEXPAND** 매개변수를 사용하여 각 큐 관리자에 대해 이 설정을 대체할 수 있습니다. 데이터 세트가 90% 차고 추가 공간이 필요할 때 자동 확장이 트리거됩니다. 확장이 허용되지만 데이터 세트가 정의될 때 2차 공간 할당이 지정되지 않아서 **VSAM**에 의해 확장 시도가 거부되면 데이터 세트의 현재 크기 20%의 2차 할당을 사용하여 확장이 재시도됩니다.

확장된 주소 지정 기능 속성으로 공유 메시지 데이터 세트가 정의된 경우에는 **VSAM** 고려사항에 따라 최대 16TB 또는 59개 볼륨으로만 최대 크기가 제한됩니다. 이는 로컬 페이지 세트의 최대 크기 64GB보다 상당히 더 큽니다.

공유 메시지 데이터 세트 활성화

큐 관리자는 애플리케이션 커플링 기능 구조에 연결되면 연관된 **DSGROUP** 매개변수를 사용하여 구조 정의가 오프로딩을 지정하는지 여부를 검사합니다. 오프로딩이 지정되면 큐 관리자는 소유한 공유 메시지 데이터 세트를 할당하고 쓰기 액세스를 위해 열고 나서 다른 큐 관리자가 소유한 기존 공유 메시지 데이터 세트에 대한 읽기 액세스를 위해 엽니다.

공유 메시지 데이터 세트를 처음으로 여는 경우(큐 공유 그룹 내 활성으로 기록되기 전) 첫 페이지에 아직 올바른 헤더가 포함되어 있지 않습니다. 큐 관리자는 헤더 정보를 채워서 큐 공유 그룹, 구조 이름, 소유하고 있는 큐 관리자를 식별합니다.

헤더를 채운 후에 큐 관리자는 새 공유 메시지 데이터 세트를 활성으로 등록하고 새 데이터 세트에 대해 다른 활성 큐 관리자에게 알리는 이벤트를 브로드캐스트합니다.

큐 관리자는 공유 메시지 데이터 세트를 열 때마다 헤더 정보의 유효성을 검증하여 올바른 데이터 세트가 여전히 사용 중이며 손상되지 않았는지 확인합니다.

z/OS Db2 환경 계획

큐 공유 그룹을 사용 중인 경우 IBM MQ가 데이터 공유 그룹의 멤버인 Db2 서브시스템에 접속해야 합니다. 이 주제를 사용하면 IBM MQ 데이터를 보유하고 있는 데 사용되는 Db2 요구사항을 이해하는 데 도움이 됩니다.

IBM MQ는 이 데이터 공유 그룹에 도달하기 위해 연결된 데이터 공유 그룹의 이름 및 연결할 Db2 서브시스템(또는 Db2 그룹)의 이름을 알고 있어야 합니다. 이 이름은 CSQ6SYSP 시스템 매개변수 매크로의 QSGDATA 매개변수에 지정되어 있습니다([CSQ6SYSP 사용에 설명되어 있음](#)).

데이터 공유 그룹에서 다음을 포함하는 데 공유 Db2 테이블이 사용됩니다.

- 큐 공유 그룹에 대한 구성 정보
- IBM MQ 공유 및 그룹 오브젝트의 특성
- 오프로드된 IBM MQ 메시지 관련 데이터(선택사항)

IBM MQ는 필요한 Db2 테이블스페이스, 테이블 및 색인을 정의하기 위한 샘플 작업의 단일 세트를 제공합니다. 이러한 작업에서는 UTS(Universal Table Spaces)를 이용합니다. 이전 제품 버전에는 두 개의 작업 세트(최신 버전의 Db2에서는 더 이상 사용되지 않는 이전 유형의 테이블스페이스를 위한 작업 세트와 UTS를 위한 작업 세트)가 있었습니다.

IBM MQ는 이전 유형의 테이블스페이스에서 계속 사용할 수 있고 이는 사용자에게 기존 큐 공유 그룹이 이미 있는 경우에 적합할 수 있습니다. 그러나 새 큐 공유 그룹을 작성하는 경우 이 그룹은 UTS를 사용해야 합니다.

Db2 V12 [기능 레벨 508](#) 은 다중 테이블 테이블스페이스를 유니버설 테이블스페이스로 마이그레이션하기 위한 무중단 마이그레이션 프로세스를 제공합니다. 이 접근 방식을 사용하면 전체 큐 공유 그룹을 가동 중단하지 않고 기존 큐 공유 그룹에서 사용하는 다중 테이블 테이블스페이스를 유니버설 테이블스페이스로 마이그레이션할 수 있습니다.

Db2 V13에서 ALTER TABLESPACE문의 MOVE TABLE 옵션을 사용하십시오. 자세한 정보는 [다중 테이블 테이블스페이스에서 partition-by-growth 테이블스페이스로 테이블 이동](#) 을 참조하십시오.

기본적으로 Db2는 Db2 자원의 소유자로서 작업을 실행하는 개인의 사용자 ID를 사용합니다. 이 사용자 ID가 삭제되면 이 사용자 ID와 연관된 자원이 삭제되므로 테이블도 삭제됩니다. 개별 사용자 ID 대신 테이블을 소유한 그룹 ID 사용을 고려하십시오. JOB 카드에 GROUP=groupname 를 추가하고 SQL문앞에 SET CURRENT SQLID='groupname' 를 지정하여 이를 수행할 수 있습니다.

IBM MQ는 Db2의 RRS 첨부 기능을 사용합니다. 이는 연결하려는 Db2 그룹의 이름을 지정할 수 있음을 의미합니다. Db2 그룹 첨부 이름(특정 Db2 서브시스템이 아님) 에 연결할 때의 이점은 IBM MQ 가 해당 그룹의 멤버인 z/OS 이미지에서 사용 가능한 Db2 서브시스템에 연결(또는 다시 연결) 할 수 있다는 것입니다. 큐 공유 IBM MQ 서브시스템을 실행할 각 z/OS 이미지에서 활성인 데이터 공유 그룹의 멤버인 Db2 서브시스템이 있어야 하며 RRS가 활성 상태여야 합니다.

Db2 스토리지

대부분의 설치에서 필요한 Db2 스토리지의 양은 3390 디바이스에서 대략 20 또는 30실린더입니다. 하지만 스토리지 요구사항을 계산하려는 경우 다음 표에 Db2가 IBM MQ 데이터에 대해 필요로 하는 스토리지 양을 판별하기 위한 일부 정보가 제공됩니다. 이 표는 각 Db2 행의 길이와 관련 Db2 테이블에서 각 행이 추가 또는 삭제되는 시기를 설명합니다. *Db2 for z/OS* 설치 안내서의 Db2 테이블 및 해당 인덱스에 대한 스페이스 요구사항 계산에 대한 정보와 함께 이 정보를 사용하십시오.

표 25. Db2 스토리지 요구사항 계획			
Db2 테이블 이름	행 길이	다음 경우에 행이 추가됨:	다음 경우에 행이 삭제됨:
CSQ.ADMIN_B_QSG	252바이트	CSQ5POSG 유틸리티의 ADD QSG 기능으로 큐 공유 그룹이 테이블에 추가됩니다.	CSQ5POSG 유틸리티의 REMOVE QSG 기능으로 큐 공유 그룹이 테이블에서 제거됩니다. (이 큐 공유 그룹에 관련된 모든 행은 큐 공유 그룹 레코드가 삭제될 때 다른 모든 Db2 테이블에서 자동으로 삭제됩니다.)
CSQ.ADMIN_B_QMGR	최대 3828 바이트	CSQ5POSG 유틸리티의 ADD QMGR 기능으로 큐 관리자가 테이블에 추가됩니다.	CSQ5POSG 유틸리티의 REMOVE QMGR 기능으로 큐 관리자가 테이블에서 제거됩니다.
CSQ.ADMIN_B_STRUCTURE	1454 바이트	큐 공유 그룹 내 이전의 알 수 없는 구조의 이름을 지정하는 QSGDISP(SHARED) 속성을 지정하여 첫 번째 로컬 큐 정의가 정의됩니다.	큐 공유 그룹 내 구조의 이름을 지정하는 QSGDISP(SHARED) 속성을 지정하여 마지막 로컬 큐 정의가 삭제됩니다.
CSQ.ADMIN_B_SCST	342바이트	공유 채널이 시작됩니다.	공유 채널이 비활성화됩니다.
CSQ.ADMIN_B_SSKT	254바이트	NPMSPEED(NORMAL) 속성이 있는 공유 채널이 시작됩니다.	NPMSPEED(NORMAL) 속성이 있는 공유 채널이 비활성화됩니다.
CSQ.ADMIN_B_STRBACKUP	514바이트	CSQ.ADMIN_B_STRUCTURE 테이블에 새 행이 추가됩니다. 더미 입력 항목을 덮어쓰는 BACKUP CFSTRUCT 명령이 실행될 때까지 각 입력 항목은 더미 입력 항목입니다.	CSQ.ADMIN_B_STRUCTURE 테이블에서 행이 삭제됩니다.
CSQ.OBJ_B_AUTHINFO	3400 바이트	QSGDISP(GROUP)를 포함한 인증 정보 오브젝트가 정의됩니다.	QSGDISP(GROUP)를 포함한 인증 정보 오브젝트가 삭제됩니다.
CSQ.OBJ_B_QUEUE	최대 3707 바이트	<ul style="list-style-type: none"> QSGDISP(GROUP) 속성이 있는 큐가 정의됩니다. QSGDISP(SHARED) 속성이 있는 큐가 정의됩니다. DEFTYPE(SHAREDYN) 속성이 있는 모델 큐가 열립니다. 	<ul style="list-style-type: none"> QSGDISP(GROUP) 속성이 있는 큐가 삭제됩니다. QSGDISP(SHARED) 속성이 있는 큐가 삭제됩니다. DEFTYPE(SHAREDYN) 속성이 있는 동적 큐는 DELETE 옵션으로 닫힙니다.
CSQ.OBJ_B_NAMELIST	최대 15127 바이트	QSGDISP(GROUP) 속성이 있는 이름 목록이 정의됩니다.	QSGDISP(GROUP) 속성이 있는 이름 목록이 삭제됩니다.
CSQ.OBJ_B_CHANNEL	최대 14127 바이트	QSGDISP(GROUP) 속성이 있는 채널이 정의됩니다.	QSGDISP(GROUP) 속성이 있는 채널이 삭제됩니다.
CSQ.OBJ_B_STGCLASS	최대 2865 바이트	QSGDISP(GROUP) 속성이 있는 스토리지 클래스가 정의됩니다.	QSGDISP(GROUP) 속성 클래스가 있는 스토리지 클래스가 삭제됩니다.
CSQ.OBJ_B_PROCESS	최대 3347 바이트	QSGDISP(GROUP) 속성이 있는 프로세스가 정의됩니다.	QSGDISP(GROUP) 속성이 있는 프로세스가 삭제됩니다.

표 25. Db2 스토리지 요구사항 계획 (계속)			
Db2 테이블 이름	행 길이	다음 경우에 행이 추가됨:	다음 경우에 행이 삭제됨:
CSQ.OBJ_B_TOPIC	최대 14520 바이트	QSGDISP(GROUP) 속성이 있는 토픽 오브젝트가 정의됩니다.	QSGDISP(GROUP) 속성이 있는 토픽 오브젝트가 삭제됩니다.
CSQ.EXTEND_B_QMGR	430바이트 미만	CSQ5POSG 유틸리티의 ADD QMGR 기능으로 큐 관리자가 테이블에 추가됩니다.	CSQ5POSG 유틸리티의 REMOVE QMGR 기능으로 큐 관리자가 테이블에서 제거됩니다.
CSQ.ADMIN_B_MESSAGES	87바이트	대형 메시지 PUT의 경우(BLOB당 1)	대형 메시지 GET의 경우(BLOB당 1)
CSQ.ADMIN_MSGS_BAUX1 CSQ.ADMIN_MSGS_BAUX2 CSQ.ADMIN_MSGS_BAUX3 CSQ.ADMIN_MSGS_BAUX4		이 4개의 테이블은 메시지의 각 BLOB에 대한 4개의 테이블 중 하나에 추가된 대형 메시지의 메시지 페이로드(payload)를 포함합니다. BLOBS 길이는 최대 511KB이므로 메시지 크기가 711KB를 초과하면 이 메시지에 대해 다중 BLOB가 있게 됩니다.	

크기가 63KB를 초과하는 많은 수의 공유 큐 메시지를 사용하면 IBM MQ 시스템에 상당한 성능 제한이 있을 수 있습니다. 자세한 정보는 SupportPac MP16, Capacity Planning and Tuning for IBM MQ for z/OS([IBM MQ 및 기타 프로젝트 영역용 SupportPacs](#)) 을 참조하십시오.

z/OS 백업 및 복구 계획

사이트의 백업 및 복구 프로시저 개발은 비용이 높고 시간이 오래 걸리는 데이터 손실을 방지하는 데 중요한 역할을 합니다. IBM MQ는 시스템 실패 후 큐와 메시지를 모두 현재 상태로 복구하기 위한 수단을 제공합니다.

이 주제에는 다음 절이 포함되어 있습니다.

- [171 페이지의 『복구 프로시저』](#)
- [172 페이지의 『백업 및 복구 팁』](#)
- [174 페이지의 『페이지 세트 복구』](#)
- [175 페이지의 『CF 구조 복구』](#)
- [175 페이지의 『특정 복구 목표 달성』](#)
- [176 페이지의 『기타 제품의 백업 고려사항』](#)
- [177 페이지의 『복구 및 CICS』](#)
- [177 페이지의 『복구 및 IMS』](#)
- [177 페이지의 『대체 사이트에서 복구 준비』](#)
- [177 페이지의 『큐 관리자 백업 활동 예』](#)

복구 프로시저

IBM MQ에 대한 다음 프로시저를 개발하십시오.

- 복구점 작성.
- 페이지 세트 백업.
- CF 구조 백업.
- 페이지 세트 복구.
- 공간 부족 조건(IBM MQ 로그 및 페이지 세트) 복구.

- CF 구조 복구.

이에 대한 정보는 [IBM MQ for z/OS 관리의 내용](#)을 참조하십시오.

사이트에서 다음에 사용하는 프로시저를 숙지하십시오.

- 하드웨어 또는 전원 실패 복구.
- z/OS 컴포넌트 실패 복구.
- 오프사이트 복구를 사용하여 사이트 인터럽트 복구.

z/OS 백업 및 복구 팁

이 주제를 사용하여 일부 백업 및 복구 태스크를 이해하십시오.

큐 관리자 재시작 프로세스는 로그 정보를 페이지 세트에 적용하여 데이터를 일치 상태로 복구합니다. 페이지 세트가 손상되었거나 사용 불가능한 경우 페이지 세트의 백업 사본을 사용하여(모든 로그가 사용 가능한 경우) 문제를 해결할 수 있습니다. 로그 데이터 세트가 손상되거나 사용 불가능하면 완전히 복구되지 않을 수도 있습니다.

다음 사항을 고려하십시오.

- 정기적으로 백업 사본을 작성하십시오.
- 필요할 수 있는 아카이브 로그는 제거하지 마십시오.
- 페이지 세트 연관으로 DDname을 변경하지 마십시오.

주기적으로 백업 사본을 작성하십시오.

복구점은 IBM MQ 페이지 세트의 백업 사본 세트 및 이 페이지 세트를 복구하는 데 필요한 해당 로그 데이터 세트를 설명하는 용어입니다. 이 백업 사본은 페이지 세트 유실(예: 페이지 세트 I/O 오류) 발생 시 잠재적인 재시작 지점을 제공합니다. 이 백업 사본을 사용하여 큐 관리자를 재시작할 경우 IBM MQ의 데이터는 이 사본이 작성된 시점까지 일치합니다. 이 시점부터 모든 로그가 사용 가능하면 IBM MQ를 장애 지점까지 복구할 수 있습니다.

보다 최신 백업 사본일수록 IBM MQ가 페이지 세트의 데이터를 더 빠르게 복구할 수 있습니다. 페이지 세트의 복구는 사용 가능한 모든 필수 로그 데이터 세트에 따라 다릅니다.

복구를 계획할 때에는 백업 사본을 작성할 빈도 및 유지할 전체 백업 순환 수를 판별해야 합니다. 이 값을 통해 IBM MQ 복구를 위한 페이지 세트의 백업 사본 및 로그 데이터 세트를 보관할 기간을 알 수 있습니다.

백업 사본의 작성 빈도를 결정할 때에는 페이지 세트를 복구하는 데 필요한 시간을 고려하십시오. 필요한 시간은 다음에 따라 판별됩니다.

- 순회할 로그의 양.
- 운영자가 아카이브 테이프 볼륨을 마운트 및 제거하는 데 걸리는 시간.
- 복구에 필요한 로그 부분을 읽는 데 걸리는 시간.
- 변경된 페이지를 재처리하는 데 필요한 시간.
- 백업 사본에 사용된 스토리지 매체.
- 백업 사본 작성 및 복원에 사용된 메소드.

일반적으로 백업 사본을 더 자주 작성할수록 복구 시간은 줄어들지만 사본 작성에 소요되는 시간은 늘어납니다.

각 큐 관리자마다 다음의 백업 사본을 작성해야 합니다.

- 아카이브 로그 데이터 세트
- 아카이브 시에 작성된 BSDS 사본
- 페이지 세트
- 오브젝트 정의
- CF 구조

백업 사본의 손실 또는 손상 위험을 줄이려면 다음을 고려하십시오.

- 백업 사본을 원래 사본과 다른 스토리지 볼륨에 저장.

- 백업 사본을 원래 사본과 다른 사이트에 저장.
- 각 페이지 세트 백업의 최소 두 개의 사본을 작성하고 단일 로깅 또는 단일 BSDS를 사용 중인 경우 아카이브 로그 및 BSDS의 두 개 사본. 이중 로깅 또는 BSDS를 사용 중인 경우에는 아카이브 로그나 BSDS 모두의 단일 사본을 작성하십시오.

IBM MQ를 프로덕션 환경으로 이동시키기 전에 백업 프로시저를 완전히 테스트하고 문서화하십시오.

페이지 세트 백업

페이지 세트를 정기적으로 백업해야 합니다. 어떤 엔터프라이즈는 하루에 두 번 페이지 세트를 백업하기도 합니다.

백업을 사용하여 복구하려면 활성 및 아카이브 로그가 필요합니다. 큐 관리자가 실행 중일 때 백업을 수행한 경우 4개의 체크포인트로 되돌아가기 위해 충분한 로그 데이터가 필요합니다.

ADRDSU FastReplication을 사용하여 페이지 세트를 백업할 수 있고 큐 관리자가 활성인 동안 이를 수행할 수 있습니다. 참고로 스토리지 풀에 충분한 공간이 있는지 확인해야 합니다.

오브젝트 정의 백업

오브젝트 정의의 백업 사본을 작성하십시오. 이를 위해 유틸리티 프로그램 COMMAND 함수의 MAKEDEF 기능을 사용하십시오(CSQUTIL의 COMMAND 기능 사용에 설명됨).

큐 관리자 데이터 세트의 백업 사본을 작성할 때마다 이를 수행하고 최신 버전을 유지해야 합니다.

커플링 기능 구조 백업

큐 공유 그룹을 설정한 경우 사용하지 않더라도 CF 구조의 주기적 백업을 수행해야 합니다. 이를 위해 IBM MQ `BACKUP CFSTRUCT` 명령을 사용하십시오. 이 명령은 RECOVER(YES) 속성으로 정의된 CF 구조에만 사용할 수 있습니다. 지속 공유 메시지에 대한 CF 입력 항목이 공유 메시지 데이터 세트(SMDS) 또는 Db2에 저장되어 있는 오프로드된 메시지 데이터를 참조하는 경우 오프로드된 데이터가 검색되고 CF 입력 항목과 함께 백업됩니다. 공유 메시지 데이터 세트는 별도로 백업해서는 안 됩니다.

CF 구조를 복원하는 데 걸리는 시간을 최소화하려면 매시간, 모든 CF 구조의 백업을 작성할 것을 권장합니다.

단일 큐 관리자의 모든 CF 구조 백업을 수행할 수 있으며 이는 로그 사용의 증가를 단일 큐 관리자로 제한하는 장점이 있습니다. 또는 큐 공유 그룹의 모든 큐 관리자에 대한 백업을 수행할 수 있으며, 이는 큐 공유 그룹에 워크로드를 분산시키는 장점이 있습니다. 어느 전략을 사용하든, IBM MQ는 백업을 찾고 큐 공유 그룹의 큐 관리자에서 RECOVER CFSTRUCT를 수행할 수 있습니다. CF 구조를 복구하려면 큐 공유 그룹의 모든 큐 관리자 로그에 액세스해야 합니다.

메시지 보안 정책 백업

Advanced Message Security를 사용하여 메시지 보안 정책의 백업을 작성 중인 경우 메시지 보안 정책 유틸리티(CSQOUTIL)를 사용하여 백업을 작성해서 -export 매개변수와 함께 `dspmqspl`을 실행한 후 EXPORT DD로 출력되는 정책 정의를 저장하십시오.

큐 관리자 데이터 세트의 백업 사본을 작성할 때마다 메시지 보안 정책의 백업을 작성하고 최신 버전을 유지해야 합니다.

필요할 수 있는 아카이브 로그를 제거하지 마십시오.

IBM MQ는 재시작 중 아카이브 로그를 사용해야 할 수 있습니다. 시스템이 완전히 복원할 수 있도록 충분한 아카이브 로그를 보관해야 합니다. IBM MQ는 아카이브 로그를 사용하여 복원된 백업 사본에서 페이지 세트를 복구할 수 있습니다. 아카이브 로그를 제거한 경우에는 IBM MQ가 페이지 세트를 현재 상태로 복원할 수 없습니다. 아카이브 로그를 제거하는 시기 및 방식은 아카이브 로그 데이터 세트 제거에 설명되어 있습니다.

`/cpf DIS USAGE TYPE(ALL)` 명령을 사용하여 큐 관리자의 페이지 세트 및 큐 공유 그룹의 구조를 복구해야 하는 로그 RBA 및 로그 범위 순서 번호(LRSN)를 표시할 수 있습니다. 그런 다음 로그 맵 인쇄 유틸리티(CSQJU004)를 사용하여 큐 관리자에 대한 부트스트랩 데이터 세트(BSDS) 정보를 인쇄해서 로그 RBA를 포함한 로그를 찾을 수 있습니다.

CF 구조의 경우에는 큐 공유 그룹의 각 큐 관리자에서 CSQJU004 유틸리티를 실행하여 LRSN을 포함한 로그를 찾아야 합니다. 페이지 세트와 구조를 복구하려면 이 로그 및 이후의 로그가 필요합니다.

DDname을 페이지 세트 연관으로 변경하지 마십시오.

IBM MQ는 페이지 세트 번호 00을 DDname CSQP0000과, 페이지 세트 번호 01을 DDname CSQP0001과 연관시키는 방식으로 최대 CSQP0099까지 페이지 세트 번호를 연관시킵니다. IBM MQ는 페이지 세트가 연관된 DDname에 기초하여 페이지 세트에 대한 복구 로그 레코드를 씁니다. 이러한 이유로 PSID DDname과 이미 연관된 페이지 세트는 이동시키면 안됩니다.

페이지 세트 복구

이 주제를 사용하여 페이지 세트의 복구 시기 및 재시작 시간을 최소화하는 방법에 관련된 요인을 이해하십시오.

복구 전략의 핵심 요인은 큐 관리자 가동 중단을 용인할 수 있는 시간에 관련됩니다. 총 가동 중단 시간에는 백업에서 페이지 세트를 복구하거나 비정상 종료 후 큐 관리자를 재시작하는 데 걸리는 시간이 포함될 수 있습니다. 재시작 시간에 영향을 미치는 요인으로는 페이지 세트를 백업하는 빈도 및 체크포인트 사이에 로그에 기록되는 데이터의 양이 있습니다.

비정상 종료 후의 재시작 시간을 최소화하려면 시스템이 다시 시작될 때 최대 두 개의 활성 로그가 사용되도록 작업 단위를 짧게 유지하십시오. 예를 들어, IBM MQ 애플리케이션을 설계 중인 경우 첫 번째 동기점 MQI 호출과 커밋점 사이의 대기 간격이 긴 MQGET 호출을 피하십시오. 지속 기간이 긴 작업 단위를 초래할 수 있기 때문입니다. 긴 작업 단위의 또 다른 일반적 원인은 채널 시작기의 5분이 넘는 배치 간격입니다.

`DISPLAY THREAD` 명령을 사용하여 작업 단위의 RBA를 표시하고 이전 문제를 해결하십시오.

페이지 세트를 얼마나 자주 백업해야 합니까?

상당히 짧은 복구 시간이 필요한 경우 빈번한 페이지 세트 백업은 필수적입니다. 페이지 세트가 아주 작거나 해당 페이지 세트에 큐에 대한 소수의 활동이 있는 경우에도 이는 필수적입니다.

페이지 세트에 지속 메시지를 사용하면 백업 빈도가 일 단위가 아닌 시 단위여야 합니다. 이는 페이지 세트 0에 대한 경우이기도 합니다.

대략적인 백업 빈도를 계산하려면 대상 총 복구 시간을 판별하는 것으로 시작하십시오. 다음으로 구성됩니다.

1. 문제에 반응하는 데 걸린 시간.
2. 페이지 세트 백업 사본을 복원하는 데 걸린 시간.

SnapShot 백업/복원을 사용하는 경우에는 이 태스크 수행에 걸리는 시간이 몇 초입니다. SnapShot에 대한 정보는 *DFSMSdss Storage Administration Guide*를 참조하십시오.

3. 페이지 세트 복구에 필요한 추가 시간을 포함하여 큐 관리자를 재시작하는 데 필요한 시간.

이 시간은 페이지 세트가 마지막으로 백업된 이후에 활성 및 아카이브 로그에서 읽어야 하는 로그 데이터의 양에 상당히 의존합니다. 손상된 페이지 세트와 직접 연관된 데이터 외에 이러한 모든 로그 데이터를 읽어야 합니다.

참고: 퍼지 백업(작업 단위가 활성 상태일 때 로그 및 페이지 세트의 스냅샷을 찍는)을 사용할 때에는 최대 세 개의 추가 체크포인트를 읽어야 하므로 이로 인해 하나 이상의 추가 로그를 읽어야 할 수 있습니다.

페이지 세트 복구를 허용할 기간을 결정할 때 고려해야 할 요인은 다음과 같습니다.

- 정상 처리 중 활성 로그에 데이터가 기록되는 비율은 메시지 비율 외에 시스템에 메시지가 도달하는 방식에 따라 다릅니다.

채널을 통해 수신 또는 송신된 메시지는 로컬로 생성 및 검색된 메시지보다 데이터 로깅을 더 많이 유발합니다.

- 아카이브 및 활성 로그에서 데이터를 읽을 수 있는 비율.

로그를 읽을 때 달성 가능한 데이터 속도는 사용된 디바이스 및 특정 DASD 서브시스템의 총 로드에서 달라집니다.

대부분의 테이프 장치의 경우 블록 크기가 큰 아카이브된 로그에 대한 데이터 속도가 더 빠를 수 있습니다. 하지만 아카이브 로그가 복구에 필요한 경우에는 활성 로그의 모든 데이터도 읽어야 합니다.

z/OS CF 구조 복구

이 주제를 사용하여 CF 구조에 대한 복구 프로세스를 이해하십시오.

큐 공유 그룹에서 RECOVER CFSTRUCT 명령을 처리할 최소 하나의 큐 관리자가 활성화되어 있어야 합니다. CF 구조 복구는 이미 활성화된 큐 관리자가 복구를 수행하기 때문에 큐 관리자 재시작 시간에 영향을 미치지 않습니다.

복구 프로세스는 RECOVER CFSTRUCT 명령이 관리하는 두 개의 논리적 단계로 이루어집니다.

1. 백업 찾기 및 복원.
2. CF 구조를 사용한 큐 공유 그룹의 모든 큐 관리자의 로그에서 CF 구조에 보유된 지속 메시지에 대한 모든 로그된 업데이트 병합 및 백업 변경사항 적용.

두 번째 단계는 많은 데이터를 읽어야 할 수 있으므로 시간이 더 오래 걸릴 수 있습니다. 백업을 자주 하는 경우나 여러 CF 구조를 동시에 복구하는 경우 또는 두 경우 모두 소요되는 시간을 줄일 수 있습니다.

복구를 수행 중인 큐 관리자는 Db2 및 부트스트랩 데이터 세트에서 데이터를 사용하여 다른 모든 큐 관리자의 로그에서 관련 백업을 찾습니다. 큐 관리자는 마지막 백업 바로 전부터 실패 시점까지 큐 공유 그룹 전체에 걸쳐 올바른 시간 순서로 이 백업을 재실행합니다.

CF 구조를 복구하는 데 걸리는 시간은 재현해야 하는 복구 로그 데이터의 양(그 다음으로 백업 빈도)에 따라 다릅니다. 즉, 큐 관리자의 로그에 쓴 시간만큼 읽는 시간이 소요됩니다. 따라서 예를 들어, 6개의 큐 관리자를 포함한 큐 공유 그룹이 있는 경우 로그 활동 1시간을 재현하는 데 6시간이 걸릴 수 있습니다. 일반적으로 읽기는 대량으로 수행할 수 있고 다른 큐 관리자의 로그를 병렬로 읽을 수 있기 때문에 시간이 덜 소요됩니다. 시작점으로서 CF 구조를 매시간 백업할 것을 권장합니다.

실패한 CF 구조가 있는 동안에 모든 큐 관리자가 비공유 큐 및 다른 CF 구조의 큐로 계속해서 작업할 수 있습니다. 관리 구조도 실패한 경우 RECOVER CFSTRUCT 명령을 발행하려면 큐 공유 그룹의 최소 하나의 큐 관리자를 시작해야 합니다.

CF 구조 백업은 상당한 로그 쓰기 용량을 요구할 수 있으므로 백업을 수행하는 큐 관리자에 큰 로드를 유발할 수 있습니다. 백업을 수행할 낮은 로드 큐 관리자를 선택하고 번잡한 시스템의 경우 큐 공유 그룹에 추가 큐 관리자를 추가해서 이 큐 관리자가 전용으로 독점적 백업을 수행하게 하십시오.

z/OS 특정 복구 목표 달성

백업 빈도를 조정해서 특정 복구 목표 시간을 달성하는 방법에 대한 지침으로 이 주제를 사용하십시오.

큐 관리자 복구 및 xx초 이내의 정상 시동 시간 외 재시작 처리 완료와 같이 달성할 특정 복구 목표가 있는 경우 다음 계산을 사용하여 백업 빈도(시간)를 추정할 수 있습니다.

$$\text{Backup frequency (in hours)} = \frac{\text{Required restart time (in secs)} * \text{System recovery log read rate (in MB/sec)}}{\text{Application log write rate (in MB/hour)}}$$

참고: 다음에 제공된 예는 페이지 세트를 자주 백업해야 할 필요성을 강조표시합니다. 계산 시 대부분의 로그 활동이 많은 수의 지속 메시지에서 파생된다고 가정합니다. 하지만 로그 활동량이 쉽게 계산되지 않는 경우가 있습니다. 예를 들어, 큐 공유 그룹 환경에서는 기타 자원 외에 공유 큐가 업데이트되는 작업 단위로 인해 UOW 레코드가 IBM MQ 로그에 기록될 수 있습니다. 이러한 이유로 공식 (A)의 애플리케이션 로그 쓰기 속도는 IBM MQ 로그가 채워지는 관찰된 속도로부터만 정확하게 도출될 수 있습니다.

예를 들어, IBM MQ MQI clients가 1초에 100개의 지속 메시지를 총 로드로 생성하는 시스템을 가정해 보십시오. 이 경우 모든 메시지가 로컬에 생성됩니다.

각 메시지의 사용자 길이가 1KB이면 매시간 로그되는 데이터의 양은 대략 다음과 같습니다.

$$100 * (1 + 1.3) \text{ KB} * 3600 = \text{approximately } 800 \text{ MB}$$

where
 100 = the message rate a second
 (1 + 1.3) KB = the amount of data logged for each 1 KB of persistent messages

전체 목표 복구 시간 75분을 고려하십시오. 문제점에 반응하고 페이지 세트 백업 사본을 복원하는 데 15분을 허용한 경우 큐 관리자 복구 및 재시작은 공식 (A)를 적용하여 60분(3600초) 이내에 완료되어야 합니다. 복구 속도가 대략 초당 2.7MB인 RVA2-T82 DASD에 필요한 모든 로그 데이터가 있다고 가정하면 페이지 세트 백업 빈도는 최소 다음 주기여야 합니다.

$$3600 \text{ seconds} * 2.7 \text{ MB a second} / 800 \text{ MB an hour} = 12.15 \text{ hours}$$

IBM MQ 애플리케이션 일이 대략 12시간 지속되면 매일 하나의 백업이 적절합니다. 하지만 애플리케이션 일이 24시간 지속되는 경우에는 매일 두 개의 백업이 더 적절합니다.

또 다른 예는 모든 메시지가 요청-응답 애플리케이션에 대한 메시지인(즉, 수신자 채널에서 지속 메시지가 수신되고 지속 응답 메시지가 생성되어 송신자 채널로 송신됨) 프로덕션 시스템일 수 있습니다.

이 예에서는 아카이브된 배치 크기가 1이므로 모든 메시지마다 배치가 하나씩 있습니다. 1초에 50개의 요청 응답이 있으면 총 로드는 초당 100개의 지속 메시지입니다. 각 메시지의 길이가 1KB이면 매시간 로그되는 데이터의 양은 대략 다음과 같습니다.

$$50((2 * (1+1.3) \text{ KB}) + 1.4 \text{ KB} + 2.5 \text{ KB}) * 3600 = \text{approximately } 1500 \text{ MB}$$

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

큐 관리자 복구 및 30분(1800초) 내 재시작을 달성하려면 필요한 모든 로그 데이터가 RVA2-T82 DASD에 있다고 가정할 때, 페이지 세트 백업을 최소 다음 주기로 이행해야 합니다.

$$1800 \text{ seconds} * 2.7 \text{ MB a second} / 1500 \text{ MB an hour} = 3.24 \text{ hours}$$

백업 빈도의 주기적 검토

시간당 MB의 관점에서 IBM MQ 로그 사용을 모니터링하십시오. 주기적으로 이 검사를 수행하고 필요에 따라 페이지 세트 백업 주기를 수정하십시오.

기타 제품의 백업 고려사항

IBM MQ를 CICS 또는 IMS와 함께 사용 중이면 이들 제품의 백업 전략에 대한 영향도 고려해야 합니다. DFHSM(Data Facility Hierarchical Storage Manager)은 데이터 스토리지를 관리하며 IBM MQ에 사용되는 스토리지와 상호작용할 수 있습니다.

DFHSM을 통한 백업 및 복구

DFHSM(Data Facility Hierarchical Storage Manager)은 시스템의 스토리지 디바이스 사이에 자동 공간 가용성 및 데이터 가용성 관리를 수행합니다. 사용할 경우 이 관리자가 IBM MQ 스토리지로(부터) 데이터를 자동으로 이동시키는 점을 알고 있어야 합니다.

DFHSM은 최근 대체 스토리지에 사용되지 않은 데이터 세트를 이동시켜서 DASD 공간을 효율적으로 관리합니다. 새 데이터 세트나 변경된 데이터 세트를 테이프 또는 DASD 백업 볼륨에 자동으로 복사하여 데이터를 복구 가능하게 하기도 합니다. 데이터 세트를 삭제하거나 다른 디바이스로 이동시킬 수 있습니다. 매일 지정된 시간에 조작이 발생하며 데이터 세트를 삭제 또는 이동시키기 전에 사전 판별된 기간 동안 보관하도록 허용합니다.

모든 DFHSM 조작을 수동으로 수행할 수도 있습니다. DFHSM에 대한 자세한 정보는 [z/OS DFSMS 제품 문서](#)를 참조하십시오. DFHSM을 IBM MQ와 함께 사용할 경우 DFHSM이 다음을 수행함에 유의하십시오.

- 카탈로그화된 데이터 세트를 사용합니다.
- 페이지 세트 및 로그에 대해 작용합니다.
- VSAM 데이터 세트를 지원합니다.

복구 및 CICS

CICS 자원의 복구는 IBM MQ의 영향을 받지 않습니다. CICS 는 IBM MQ 를 비CICS 자원 (또는 외부 자원 관리자) 으로 인식하고, CICS 자원 관리자 인터페이스 (RMI) 를 사용하여 동기점 조정 요청에 IBM MQ 를 참가자로 포함합니다. CICS 복구 및 CICS 자원 관리자 인터페이스에 대한 자세한 정보는 [CICS 제품 문서](#)를 참조하십시오.

복구 및 IMS

IMS는 IBM MQ를 외부 시스템 및 동기점 조정의 참여자로 인식합니다. 외부 서브시스템 자원에 대한 IMS 복구는 [IMS 제품 문서](#)에 설명되어 있습니다.

대체 사이트에서 복구 준비

IBM MQ 컴퓨팅 센터 전체가 손실된 경우 복구 사이트의 다른 IBM MQ 시스템에서 복구할 수 있습니다.

복구 사이트에서 IBM MQ 시스템을 복구하려면 페이지 세트와 로그를 정기적으로 백업해야 합니다. 모든 데이터 복구 조작에서 재해 복구의 목표는 데이터, 워크로드 처리(업데이트), 시간을 가능한 거의 잃지 않는 것입니다.

복구 사이트에서:

- 복구 IBM MQ 큐 관리자는 반드시 손실된 큐 관리자와 이름이 같아야 합니다.
- 복구 큐 관리자에 사용되는 시스템 매개변수 모듈에 손실된 큐 관리자와 동일한 매개변수가 포함되었는지 확인하십시오.

자세한 정보는 [IBM MQ for z/OS 관리 및 IBM MQ for z/OS 문제점 해결](#) 의 내용을 참조하십시오.

큐 관리자 백업 활동 예

이 주제는 큐 관리자 백업 활동의 예를 보여줍니다.

큐 관리자 백업 전략을 계획할 때 주요 고려사항은 올바른 양의 로그 데이터 보유입니다. [로그 관리](#)에는 큐 관리자의 시스템 복구 RBA를 참조하여 필요한 로그 데이터 세트를 판별하는 방법이 설명되어 있습니다. IBM MQ는 다음에 대한 정보를 사용하여 시스템 복구 RBA를 판별합니다.

- 현재 활성 작업 단위.
- 버퍼 풀에서 디스크로 아직 비우지 않은 페이지 세트 업데이트.
- CF 구조 백업 및 이 큐 관리자의 로그에 이를 사용한 복구 조작에 필요한 정보가 포함되었는지 여부.

매체 복원을 수행하려면 충분한 로그 데이터를 보유해야 합니다. 시스템 복구 RBA가 시간이 지남에 따라 증가하는 동안 보유해야 하는 로그 데이터의 양은 후속 백업이 수행될 때 감소하기만 합니다. CF 구조 백업은 IBM MQ에서 관리되므로 시스템 복구 RBA를 보고할 때 고려됩니다. 이는 사실상, 보유해야 하는 로그 데이터의 양이 페이지 세트 백업이 수행될 때 감소함을 의미합니다.

178 페이지의 그림 43은 큐 공유 그룹의 멤버인 큐 관리자의 백업 활동, 복구 RBA가 각 백업에서 변화하는 방식, 보유해야 하는 로그 데이터의 양에 영향을 미치는 방식을 보여줍니다. 예에서 큐 관리자는 로컬 및 공유 자원 즉, 페이지 세트와 두 개의 CF 구조, STRUCTURE1 및 STRUCTURE2를 사용합니다.

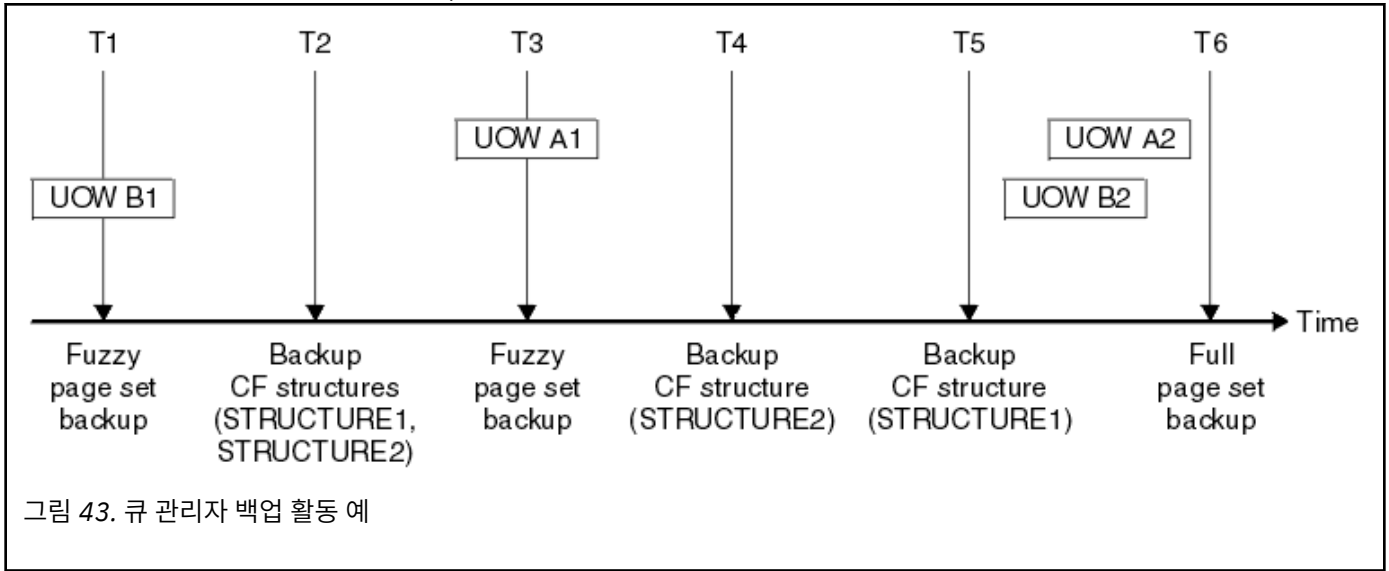


그림 43. 큐 관리자 백업 활동 예

각 시점에 발생하는 사항은 다음과 같습니다.

시점 T1

페이지 세트 백업 및 복구 방법에 설명된 대로 페이지 세트의 퍼지 백업이 작성됩니다.

큐 관리자의 시스템 복구 RBA가 다음의 최저입니다.

- 현재 백업되고 있는 페이지 세트의 복구 RBA.
- CF 애플리케이션 구조를 복구하는 데 필요한 최저 복구 RBA. 이는 이전에 작성된 STRUCTURE1 및 STRUCTURE2의 백업 복구에 관련됩니다.
- 큐 관리자 내에서 가장 오래된 현재 활성화된 작업 단위(UOWB1)에 대한 복구 RBA.

퍼지 백업 프로세스의 일부인 DISPLAY USAGE 명령이 발행한 메시지를 통해 이 시점에 대한 시스템 복구 RBA가 제공됩니다.

시점 T2

CF 구조의 백업이 작성됩니다. CF 구조 STRUCTURE1이 먼저 백업되고 그 다음에 STRUCTURE2가 백업됩니다.

T1에 시스템 복구 RBA에서 판별된 동일한 데이터가 T1에 수행된 페이지 세트 백업을 사용한 복구에 여전히 필요하기 때문에 보유해야 하는 로그 데이터의 양은 변경되지 않습니다.

시점 T3

또 다른 퍼지 백업이 작성됩니다.

큐 관리자의 시스템 복구 RBA가 다음의 최저입니다.

- 현재 백업되고 있는 페이지 세트의 복구 RBA.
- STRUCTURE1이 STRUCTURE2 전에 백업되었기 때문에 CF 구조 STRUCTURE1를 복구하는 데 필요한 최저 복구 RBA.
- 큐 관리자 내에서 가장 오래된 현재 활성화된 작업 단위(UOWA1)에 대한 복구 RBA.

퍼지 백업 프로세스의 일부인 DISPLAY USAGE 명령이 발행한 메시지를 통해 이 시점에 대한 시스템 복구 RBA가 제공됩니다.

이제 새 시스템 복구 RBA를 통해 판별된 대로 보유되는 로그 데이터를 줄일 수 있습니다.

시점 T4

CF 구조 STRUCTURE2의 백업이 수행됩니다. 가장 오래된 필수 CF 구조 백업의 복구를 위한 복구 RBA가 시점 T2에 백업된 CF 구조 STRUCTURE1의 백업에 관련됩니다.

이 CF 구조 백업의 작성은 보유해야 하는 로그 데이터의 양에 영향을 미치지 않습니다.

시점 T5

CF 구조 STRUCTURE1의 백업이 수행됩니다. 가장 오래된 필수 CF 구조 백업의 복구를 위한 복구 RBA가 이제 시점 T4에 백업된 CF 구조 STRUCTURE2의 복구에 관련됩니다.

이 CF 구조 백업의 작성은 보유해야 하는 로그 데이터의 양에 영향을 미치지 않습니다.

시점 T6

페이지 세트 백업 및 복구 방법에 설명된 대로 페이지 세트의 전체 백업이 수행됩니다.

큐 관리자의 시스템 복구 RBA가 다음의 최저입니다.

- 현재 백업되고 있는 페이지 세트의 복구 RBA.
- CF 구조를 복구하는 데 필요한 최저 복구 RBA. CF 구조 STRUCTURE2의 복구에 관련됩니다.
- 큐 관리자 내에서 가장 오래된 현재 활성화된 작업 단위에 대한 복구 RBA. 이 경우 현재 작업 단위는 없습니다.

전체 백업 프로세스의 일부인 DISPLAY USAGE 명령이 발행한 메시지를 통해 이 시점에 대한 시스템 복구 RBA가 제공됩니다.

전체 백업과 연관된 시스템 복구 RBA가 보다 최신이므로 보유되는 로그 데이터를 줄일 수 있습니다.

z/OS

z/OS UNIX 환경 계획

IBM MQ 큐 관리자, 채널 시작기 및 mqweb 서버 내의 특정 프로세스는 정상 처리를 위해 z/OS UNIX System Services (z/OS UNIX) 를 사용합니다.

z/OS UNIX에 액세스할 수 있으려면 큐 관리자와 채널 시작기에서 시작한 태스크 사용자 ID에 UID가 정의된 OMVS 세그먼트가 있어야 합니다. 사용자 ID는 z/OS UNIX에 대한 특수 권한이 없어도 됩니다.

참고: 큐 관리자 및 채널 시작기가 z/OS UNIX 기능을 사용하더라도 (예를 들어, TCP/IP 서비스와 인터페이스하기 위해) z/OS UNIX 파일 시스템에서 IBM MQ 설치 디렉토리의 콘텐츠에 액세스할 필요가 없습니다. 결과적으로 큐 관리자와 채널 시작기는 z/OS UNIX 파일 시스템의 경로를 지정하기 위한 구성이 필요하지 않습니다.

IBM MQ Console 및 REST API를 호스트하는 mqweb 서버는 z/OS UNIX 파일 시스템의 IBM MQ 설치 디렉토리에 있는 파일을 사용합니다. 또한 구성 및 로그 파일과 같은 데이터를 저장하는 데 사용되는 기타 파일 시스템에 대한 액세스가 필요합니다. mqweb 시작된 태스크 JCL은 이러한 z/OS UNIX 파일 시스템을 참조하도록 사용자 정의해야 합니다.

z/OS UNIX 파일 시스템에 있는 IBM MQ 디렉토리의 콘텐츠는 IBM MQ에 연결하는 애플리케이션에서도 사용됩니다. 예를 들어, IBM MQ classes for Java 또는 IBM MQ classes for JMS 인터페이스를 사용하는 애플리케이션입니다.

관련 구성 지시사항은 다음 주제를 참조하십시오.

- [IBM MQ classes for Java에 관련된 환경 변수](#)
- [IBM MQ classes for Java 라이브러리](#)
- [환경 변수 설정](#)
- [JNI\(Java Native Interface\) 라이브러리](#)

z/OS

Advanced Message Security 계획

TLS(또는 SSL)는 네트워크에서 플로우되는 메시지를 암호화하고 보호하는 데 사용될 수 있지만 큐에 있는("저장된") 메시지는 보호하지 않습니다. AMS(Advanced Message Security)는 의도된 메시지 수신인만 해당 메시지를 읽을 수 있도록 메시지를 처음 큐에 넣을 때부터 가져올 때까지 메시지를 보호합니다. 메시지는 PUT 처리 동안 암호화 및 서명되고 GET 처리 동안에는 보호되지 않습니다.

AMS는 다른 방법으로 메시지를 보호하도록 구성될 수 있습니다.

1. 메시지가 서명될 수 있습니다. 메시지는 일반 텍스트이지만 체크섬이 있으며 체크섬은 서명됩니다. 이렇게 하면 메시지 콘텐츠의 모든 변경사항이 감지될 수 있습니다. 서명된 콘텐츠에서 데이터를 서명한 사람을 식별할 수 있습니다.
2. 메시지를 암호화할 수 있습니다. 복호화 키가 없으면 콘텐츠가 사용자에게 표시되지 않습니다. 복호화 키는 각 수신인을 위해 암호화됩니다.

3. 메시지를 암호화하고 서명할 수 있습니다. 암호화 키는 각 수신인을 위해 암호화되고 서명을 통해 메시지를 전송한 사람을 식별할 수 있습니다.

암호화 및 서명은 디지털 서명과 키 링을 사용합니다.

데이터를 클라이언트 채널에 넣기 전에 데이터를 보호하도록 클라이언트에서 AMS를 사용하도록 설정할 수 있습니다. 보호된 메시지는 리모트 큐 관리자로 전송될 수 있으며 이러한 메시지를 처리하기 위해 리모트 큐 관리자를 구성해야 합니다.

AMS 설정

AMS 주소 공간은 AMS 작업 수행에 사용됩니다. 여기에는 키 링 및 인증서에 액세스하고 사용을 보호하기 위한 추가 보안 설정이 있습니다.

큐에 대한 보안 정책을 정의하기 위해 유틸리티 프로그램(CSQOUTIL)을 사용하여 어떤 큐를 보호할지를 구성합니다.

AMS를 설정한 후

메시지를 넣고 메시지를 가져올 사람을 위해 디지털 인증서와 키 링을 설정해야 합니다.

Alice라는 z/OS 사용자가 Bob에게 메시지를 전송해야 하는 경우 AMS에는 Bob을 위한 공용 인증서 사본이 필요합니다.

Bob이 Alice의 메시지를 처리하려면 AMS에 Alice를 위한 공용 인증서 또는 Alice가 사용한 동일한 인증 기관 인증서가 필요합니다.



주의: 다음을 수행해야 합니다.

- 큐에 넣거나 가져올 수 있는 사람을 유의하여 계획
- 사람들 및 해당 인증서 이름을 식별.

실수하기 쉽고 문제점을 해결하기 어려울 수 있습니다.

관련 개념

130 페이지의 『큐 관리자에 대한 계획』

큐 관리자를 설정하는 경우 사용자의 계획이 큐 관리자의 확대를 허용해야 큐 관리자가 사용자 엔터프라이즈의 요구사항에 충족할 수 있습니다.

z/OS Managed File Transfer 계획

이 절을 z/OS에서 MFT(Managed File Transfer)를 실행하도록 시스템을 설정하는 방법에 대한 지침으로 사용하십시오.

z/OS Managed File Transfer 계획 - 하드웨어 및 소프트웨어 요구사항

이 주제를 z/OS에서 MFT(Managed File Transfer)를 실행하도록 시스템에서 하드웨어 및 소프트웨어 요구사항을 설정하는 방법에 대한 지침으로 사용하십시오.

소프트웨어 요구사항

Managed File Transfer는 Java로 작성되며 일부 셸 스크립트와 JCL을 사용하여 프로그램을 구성하고 운영할 수 있습니다.

중요사항: Managed File Transfer를 구성하려면 z/OS UNIX System Services (z/OS UNIX)에 익숙해야 합니다. 예를 들면, 다음과 같습니다.

- /u/userID/myfile.txt와 같은 이름의 파일 디렉토리 구조
- z/OS UNIX 명령. 예를 들어 다음과 같습니다.
 - cd(디렉토리 변경)
 - ls(목록)

- chmod(파일 권한 변경)
- chown(파일 또는 디렉토리에 액세스할 수 있는 파일 소유권 또는 그룹 변경)

MFT를 구성하고 실행할 수 있으려면 z/OS UNIX에 다음 제품이 필요합니다.

1. Java(예: /java/java80_bit64_GA/J8.0_64/ 디렉토리)
2. IBM MQ 9.3.0(예: /mqm/V9R3M0 디렉토리)
3. 상태 및 실행 기록에 Db2를 사용하려는 경우 예를 들어 /db2/db2v10/jdbc/libs 디렉토리에 Db2 JDBC 라이브러리를 설치해야 합니다.

제품 등록

시작 시 Managed File Transfer는 sys1.parmlib(IFAPRDxx) 연결에서 등록을 확인합니다. 다음 코드는 MFT 등록 방법의 예입니다.

```
PRODUCT OWNER('IBM CORP')
NAME('WS MQ FILE TRANS')
ID(5655-MFT)
VERSION(*) RELEASE(*) MOD(*)
FEATURENAME('WS MQ FILE TRANS')
STATE(ENABLED)
```

디스크 공간

IBM MQ for z/OS 프로그램 디렉토리에는 Managed File Transfer에 대한 DASD 및 zFS 스토리지 요구사항이 기재되어 있습니다. IBM MQ for z/OS의 프로그램 디렉토리에 대한 다운로드 링크는 [IBM MQ 9.3 제품 문서 및 프로그램 디렉토리의 PDF 파일을 참조하십시오.](#)

z/OS Managed File Transfer 계획 - 토폴로지

이 주제를 z/OS에서 MFT(Managed File Transfer)를 실행하기 위해 시스템에 필요한 토폴로지에 대한 지침으로 사용하십시오.

Managed File Transfer 큐 관리자

IBM MQ Managed File Transfer 토폴로지는 다음으로 구성됩니다.

에이전트 및 이와 연관된 큐 관리자

에이전트는 해당 에이전트 큐 관리자에서 호스트되는 시스템 큐를 사용하여 상태 정보를 유지보수하고 작업 요청을 수신합니다.

명령 큐 관리자

MFT 토폴로지에 대한 게이트웨이 역할을 하며 송신자 및 수신자 채널 또는 클러스터링을 통해 에이전트 큐 관리자에 연결됩니다. 특정 명령이 실행되면 명령 큐 관리자에 직접 연결하여 지정된 에이전트에 메시지를 전송합니다. 이 메시지는 IBM MQ 네트워크를 통해 에이전트 큐 관리자로 라우팅되며, 여기서 에이전트에 의해 선택되어 처리됩니다.

조정 큐 관리자

전체 토폴로지에 대한 지식이 있는 중앙 허브입니다. 조정 큐 관리자는 송신자 및 수신자 채널을 통해 또는 클러스터링을 사용하여 토폴로지의 모든 에이전트 큐 관리자에 연결됩니다. 에이전트는 정기적으로 상태 정보를 조정 큐 관리자에 발행하고 여기에 해당 전송 템플릿을 저장합니다.

단일 큐 관리자가 토폴로지 내에서 여러 역할을 수행할 수 있습니다. 예를 들어, 동일한 큐 관리자를 토폴로지에 대한 조정 큐 관리자 및 명령 큐 관리자 둘 다로 구성할 수 있습니다.

여러 큐 관리자를 사용하는 경우 큐 관리자 사이의 채널을 설정해야 합니다. 클러스터링을 사용하거나 포인트-투-포인트 연결을 사용하여 수행할 수 있습니다.

IBM MQ Managed File Transfer for z/OS를 사용하는 경우 토폴로지 내에서 다양한 역할에 사용할 큐 관리자를 판별할 때 고려해야 할 여러 사항이 있습니다.

에이전트 큐 관리자

IBM MQ Managed File Transfer for z/OS 에이전트에 대한 에이전트 큐 관리자가 z/OS에서 실행 중이어야 합니다.

다음과 같은 경우

- 에이전트가 IBM MQ 9.1 이상에서 Managed File Transfer for z/OS를 실행 중입니다.
- 또한 에이전트 큐 관리자에 Advanced VUE(IBM MQ Advanced for z/OS Value Unit Edition)에 대한 라이선스가 부여되었습니다.

에이전트는 CLIENT 전송을 사용하여 큐 관리자에 연결할 수 있습니다.



그림 44. z/OS의 MFT 9.1 에이전트는 큐 관리자에 고급 VUE에 대한 라이선스가 있다고 가정하여 CLIENT 전송을 사용하여 큐 관리자에 연결할 수 있습니다.

다음과 같은 경우

- 에이전트가 IBM MQ 9.0 이하에서 Managed File Transfer for z/OS를 실행 중입니다.
- 또는 에이전트 큐 관리자가 IBM MQ 9.0 이상에서 Managed File Transfer for z/OS를 실행 중이고 에이전트 큐 관리자에 MFT, IBM MQ Advanced for z/OS 또는 Advanced VUE에 대한 라이선스가 부여되었습니다.

에이전트는 BINDINGS 전송을 사용하여 큐 관리자에 연결해야 합니다.



그림 45. MFT 또는 IBM MQ Advanced에 대해 라이선스가 부여된 에이전트 큐 관리자가 있는 z/OS 및 9.1 에이전트의 MFT 9.0 에이전트는 BINDINGS 전송을 사용하여 연결해야 합니다.

명령 큐 관리자

MFT 명령 및 프로세스와 큐 관리자의 연결 조합 주제에서 Managed File Transfer 토폴로지에 대한 명령 큐 관리자에 연결하는 모든 명령을 보여줍니다.

참고: z/OS에서 이러한 명령을 실행할 때는 명령 큐 관리자도 z/OS에 있어야 합니다.

명령 큐 관리자에 Advanced VUE에 대한 라이선스가 부여된 경우, 명령은 CLIENT 전송을 사용하여 큐 관리자에 연결할 수 있습니다. 그렇지 않은 경우에는 명령이 BINDINGS 전송을 사용하여 명령 큐 관리자에 연결해야 합니다.

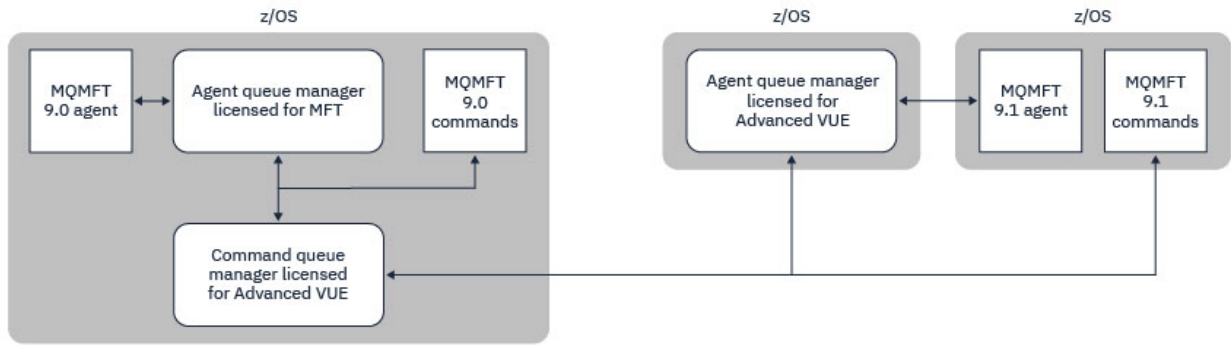


그림 46. 명령은 MFT 토폴로지의 명령 큐 관리자에 연결합니다. z/OS에서 이러한 명령을 실행할 때는 명령 큐 관리자도 z/OS에 있어야 합니다.

조정 큐 관리자

IBM MQ Managed File Transfer for z/OS 에이전트는 조정 큐 관리자가 z/OS에서 실행 중이거나 멀티플랫폼에서 실행 중인 토폴로지의 일부일 수 있습니다.

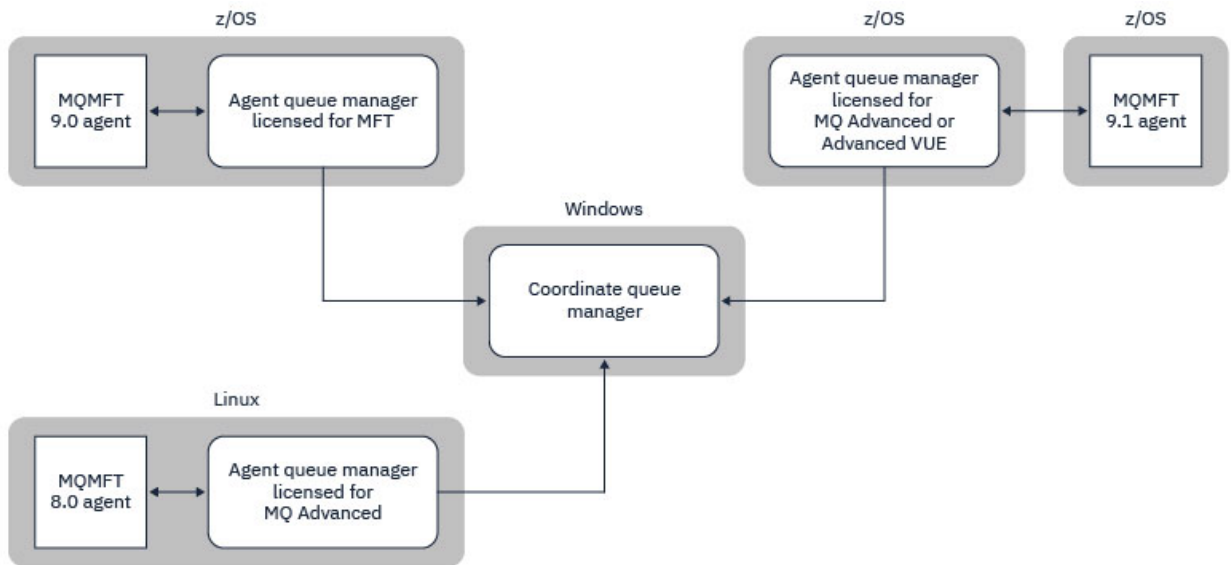


그림 47. z/OS에서 실행 중인 MFT 에이전트는 조정 큐 관리자가 IBM MQ 멀티플랫폼에서 실행 중인 MFT 토폴로지의 일부일 수 있습니다.

MFT 명령 및 프로세스와 큐 관리자의 연결 조합 주제에서 Managed File Transfer 토폴로지에 대한 조정 큐 관리자에 연결하는 명령을 보여줍니다. z/OS에서 이러한 명령을 실행하고 다른 플랫폼에서 실행 중인 조정 큐 관리자에 연결하도록 할 수 있습니다.

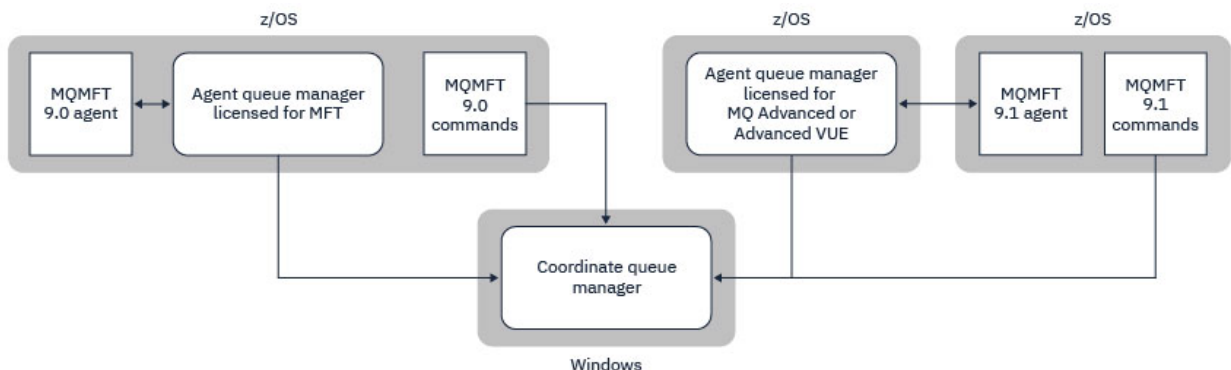


그림 48. **ftelListAgents**와 같은 특정 명령은 MFT 토폴로지에 대한 조정 큐 관리자에 직접 연결합니다.

필요한 에이전트 수

에이전트는 데이터 전송 작업을 수행하므로, 데이터 전송을 요청할 때 에이전트의 이름을 지정합니다.

기본적으로 에이전트는 25개의 전송 요청과 25개의 수신 요청을 동시에 처리할 수 있습니다. 이러한 프로세스를 구성할 수 있습니다. 자세한 정보는 [z/OS의 Managed File Transfer 구성 옵션의 내용](#)을 참조하십시오.

에이전트가 사용 중이면 작업을 큐에 넣습니다. 요청을 처리하는 데 걸리는 시간은 전송할 데이터 크기, 네트워크 대역폭, 네트워크 지연 등의 여러 요인에 따라 달라집니다.

여러 에이전트가 작업을 병렬로 처리하게 할 수 있습니다.

에이전트가 액세스할 수 있는 자원도 제어할 수 있으므로 일부 에이전트가 데이터의 제한된 서브세트에 대해서만 작업하게도 할 수 있습니다.

우선순위가 서로 다른 요청을 처리하려는 경우 여러 에이전트를 사용하고 워크로드 관리자를 사용하여 작업의 우선순위를 설정할 수 있습니다.

에이전트 실행

일반적으로 에이전트는 오래 실행되는 프로세스입니다. 프로세스는 배치로 실행되는 작업 또는 시작된 태스크로 제출할 수 있습니다.

Managed File Transfer 계획 - 보안 고려사항

이 주제를 z/OS에서 MFT(Managed File Transfer)를 실행하기 위해 시스템에 필요한 보안 고려사항에 대한 지침으로 사용하십시오.

보안

MFT 구성 및 MFT 운영에 사용할 사용자 ID를 식별해야 합니다.

전송하는 파일 또는 큐, MFT에 전송 요청을 제출할 사용자 ID를 식별해야 합니다.

에이전트와 로거를 사용자 정의할 때 MFT 서비스를 실행하거나 MFT 관리를 수행할 수 있는 사용자 그룹을 지정합니다.

MFT 사용자 정의를 시작하기 전에 이 그룹을 설정해야 합니다. MFT는 IBM MQ 큐를 사용하므로 큐 관리자에서 보안을 사용으로 설정한 경우 MFT에는 다음 자원에 액세스해야 합니다.

표 26. MQADMIN 자원 클래스	
이름	필요한 액세스
QUEUE.SYSTEM.FTE.EVENT.agent_name	업데이트
QUEUE.SYSTEM.FTE.COMMAND.agent_name	업데이트
CONTEXT.SYSTEM.FTE.COMMAND.agent_name	업데이트
QUEUE.SYSTEM.FTE.STATE.agent_name	업데이트
QUEUE.SYSTEM.FTE.DATA.agent_name	업데이트
QUEUE.SYSTEM.FTE.REPLY.agent_name	업데이트
QUEUE.SYSTEM.FTE.AUTHAGT1.agent_name	업데이트
QUEUE.SYSTEM.FTE.AUTHTRN1.agent_name	업데이트
QUEUE.SYSTEM.FTE.AUTHOPS1.agent_name	업데이트
QUEUE.SYSTEM.FTE.AUTHSCH1.agent_name	업데이트
QUEUE.SYSTEM.FTE.AUTHMON1.agent_name	업데이트
QUEUE.SYSTEM.FTE.AUTHADM1.agent_name	업데이트

표 27. MQQUEUE 자원 클래스	
이름	필요한 액세스
SYSTEM.FTE.AUTHAGT1.agent_name	업데이트
SYSTEM.FTE.AUTHTRN1.agent_name	업데이트
SYSTEM.FTE.AUTHOPS1.agent_name	업데이트
SYSTEM.FTE.AUTHSCH1.agent_name	업데이트
SYSTEM.FTE.AUTHMON1.agent_name	업데이트

전송을 요청하는 사용자가 액세스할 수 있는 파일 시스템의 부분을 판별하는 데 사용자 샌드박스를 사용할 수 있습니다.

사용자 샌드박스를 사용하려면 제한하려는 에이전트의 *agent.properties* 파일에 `userSandboxes=true`문을 추가하고 적절한 값을 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name/UserSandboxes.xml`에 추가하십시오.

자세한 정보는 [사용자 샌드박스](#)에 대한 작업을 참조하십시오.

이 사용자 ID는 `UserSandboxes.xml` 파일에 구성되어 있습니다.

이 XML 파일에는 사용자 ID 또는 사용자 ID* 및 사용할 수 있거나(포함) 사용할 수 없는(제외) 자원 목록 등의 정보가 있습니다. 특정 자원에 액세스할 수 있는 특정 사용자 ID를 정의해야 합니다. 예를 들어, 다음과 같습니다.

표 28. 특정 자원에 대한 액세스 권한이 있는 예제 사용자 ID			
사용자 ID	액세스	포함 또는 제외	자원
Admin*	읽기	포함	/home/user/**
Admin*	읽기	제외	/home/user/private/**
Sysprog	읽기	포함	/home/user/**
Admin*	읽기	포함	Application.reply.queue

참고:

1. `type=queue`가 지정된 경우 자원은 큐 이름이거나 `queue@qmgr`입니다.
2. 자원이 `//`로 시작될 경우 자원은 데이터 세트입니다. 그렇지 않을 경우 자원은 z/OS UNIX의 파일입니다.
3. 사용자 ID는 MQMD 구조의 사용자 ID이므로 실제로 메시지를 넣는 사용자 ID를 나타내지 않을 수 있습니다.
4. 로컬 큐 관리자에 대한 요청의 경우 `MQADMIN CONTEXT.*`를 사용할 수 있습니다. 이 값을 설정할 수 있는 사용자를 제한합니다.
5. 리모트 큐 관리자를 통해 수신되는 요청의 경우 MQMD 구조에서 승인되지 않은 사용자 ID가 설정되지 않도록 분산 큐 관리자에 보안이 사용된다고 가정해야 합니다.
6. Linux 시스템에서 사용자 ID `SYSPROG1`은 z/OS의 보안 검사용과 동일한 사용자 ID `SYSPROG1`입니다.

z/OS z/OS 에서 IBM MQ Console 및 REST API 사용 계획

IBM MQ Console 및 REST API는 `mqweb`으로 알려진 WebSphere Liberty(Liberty) 서버에서 실행되는 애플리케이션입니다. `mqweb` 서버는 시작된 태스크로서 실행됩니다. IBM MQ Console에서는 웹 브라우저를 큐 관리자 관리에 사용할 수 있습니다. REST API는 큐 관리자를 관리하고 메시징을 수행하도록 애플리케이션에 대한 단순한 프로그래밍 방식의 인터페이스를 제공합니다.

설치 및 구성 파일

z/OS UNIX(z/OS UNIX System Services)에서 mqweb 서버를 실행하는 데 필요한 파일을 설치하는 IBM MQ for z/OS UNIX System Services Web Components 기능을 설치해야 합니다. mqweb 서버를 구성하고 관리하려면 z/OS UNIX에 대해 잘 알고 있어야 합니다.

IBM MQ for z/OS UNIX System Services Components 설치에 대한 정보는 [IBM MQ for z/OS Program Directory PDF](#) 파일을 참조하십시오.

z/OS UNIX의 IBM MQ 파일은 mqweb 서버의 올바른 조작에 필요한 다양한 속성 세트와 함께 설치됩니다. 한 시스템에 IBM MQ z/OS UNIX를 설치하고 다른 시스템에서 IBM MQ를 실행하는 경우와 같이 IBM MQ 설치 파일을 복사해야 하는 경우에는 설치 중에 작성된 IBM MQ ZFS를 복사하고 대상에서 이를 읽기 전용으로 마운트해야 합니다. 다른 방법으로 파일을 복사하면 일부 파일 속성이 손실될 수 있습니다.

mqweb 서버를 작성할 때는 사용자가 Liberty 사용자 디렉토리의 위치를 결정하고 이를 작성해야 합니다. 이 디렉토리에는 구성 및 로그 파일이 포함되어 있으며 위치는 /var/mqm/mqweb과 유사할 수 있습니다.

여러 레벨의 큐 관리자에서 IBM MQ Console 및 REST API 사용

REST API 는 REST API를 실행하는 mqweb 서버와 동일한 버전, 릴리스 및 수정 (VRM) 에서 실행되는 큐 관리자 와만 직접 상호작용할 수 있습니다. 예를 들어, IBM MQ 9.3.0 REST API 는 IBM MQ 9.3.0에 있는 로컬 큐 관리자 와만 직접 상호작용할 수 있으며 IBM MQ 9.2.5 REST API 는 IBM MQ 9.2.5에 있는 로컬 큐 관리자 와만 직접 상호작용할 수 있습니다.

REST API 를 사용하면 게이트웨이 큐 관리자를 구성하여 mqweb 서버와 다른 버전의 큐 관리자를 관리할 수 있습니다. 하지만 게이트웨이 큐 관리자 로 사용할 하나 이상의 큐 관리자는 mqweb 서버와 동일한 버전이어야 합니다. 자세한 정보는 [REST API를 사용한 원격 관리](#)를 참조하십시오.

IBM MQ Console 를 사용하여 IBM MQ Console와 동일한 버전에서 실행되는 로컬 큐 관리자를 관리할 수 있습니다. **V 9.3.0** IBM MQ 9.3.0부터 IBM MQ Console 를 사용하여 원격 시스템에서 실행 중이거나 IBM MQ Console에 대한 다른 버전에서 실행 중인 큐 관리자를 관리할 수도 있습니다. 자세한 정보는 [IBM MQ Console에 리모트 큐 관리자 추가](#)의 내용을 참조하십시오.

마이그레이션

큐 관리자가 한 개뿐인 경우, mqweb 서버를 하나의 시작된 태스크로서 실행하고 큐 관리자를 마이그레이션할 때 사용하는 라이브러리를 변경할 수 있습니다.

큐 관리자가 둘 이상인 경우, 마이그레이션 중에 여러 이름의 시작된 태스크를 사용하여 서로 다른 버전에서 mqweb 서버를 시작할 수 있습니다. 이러한 이름은 원하는 이름이 될 수 있습니다. 예를 들어, 시작된 태스크 MQWB0910를 사용하여 IBM MQ 9.1.0 mqweb 서버를 시작하고 시작된 태스크 MQWB0905를 사용하여 IBM MQ 9.0.5 mqweb 서버를 시작할 수 있습니다.

그런 다음 큐 관리자를 하나의 버전에서 후속 버전으로 마이그레이션하면 큐 관리자가 후속 버전의 mqweb 서버에서 사용 가능하게 되며 이전 버전의 mqweb 서버에서는 더 이상 사용할 수 없게 됩니다.

모든 큐 관리자를 후속 버전으로 마이그레이션한 후 이전 버전의 mqweb 서버를 삭제할 수 있습니다.

HTTP 포트

mqweb 서버는 HTTP에 최대 두 개의 포트를 사용합니다.

- HTTPS용 포트. 기본값은 9443입니다.
- HTTP용 포트. HTTP는 기본적으로 사용할 수 없지만, 사용 가능한 경우 기본값은 9080입니다.

기본 포트 값이 사용 중인 경우 다른 포트를 할당해야 합니다. 둘 이상의 IBM MQ 버전에서 둘 이상의 mqweb 서버가 동시에 실행 중인 경우 각 버전에 개별 포트를 할당해야 합니다. mqweb 서버가 사용하는 포트 설정에 대한 자세한 정보는 [HTTP 및 HTTPS 포트 구성](#)을 참조하십시오.

다음 TSO 명령을 사용하여 포트에 대한 정보를 표시할 수 있습니다.

```
NETSTAT TCP tcpip (PORT portNumber)
```

여기서 *tcpip*는 TCP/IP 주소 공간의 이름이고 *portNumber*는 정보를 표시할 포트 번호를 지정합니다.

보안 - mqweb 서버 시작

mqweb 서버 사용자 ID는 특정 권한이 필요합니다. 자세한 정보는 [mqweb 서버의 시작된 태스크 사용자 ID에 필요한 권한](#)을 참조하십시오.

보안 - IBM MQ Console 및 REST API 사용

IBM MQ Console 및 REST API를 사용하는 경우 구성된 레지스트리에 포함된 사용자로서 인증해야 합니다. 이러한 사용자에게 사용자가 수행할 수 있는 조치를 판별하는 특정 역할이 지정됩니다. 예를 들어, messaging REST API를 사용하려면 사용자에게 MQWebUser 역할이 지정되어야 합니다. IBM MQ Console 및 REST API에 사용 가능한 역할 및 이러한 역할에 부여된 액세스 권한에 대한 자세한 정보는 [IBM MQ Console 및 REST API에서의 역할을 참조하십시오](#).

IBM MQ Console 및 REST API의 보안 구성에 대한 자세한 정보는 [IBM MQ Console 및 REST API 보안](#)을 참조하십시오.

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 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

07326

서울특별시 영등포구
국제금융로 10, 3IFC
한국 아이.비.엠 주식회사
U.S.A.

2바이트(DBCS) 정보에 관한 라이선스 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 라이선스 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 라이선스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이선스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 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 추정

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 라이선스:

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 소프트웨어로 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 프로그램을 작성하여 WebSphere MQ서비스를 얻을 수 있도록 하는 계획된 프로그래밍 인터페이스에 대한 정보가 포함되어 있습니다.

그러나 본 정보에는 진단, 수정 및 성능 조정 정보도 포함되어 있습니다. 진단, 수정 및 성능 조정 정보는 응용프로그램 소프트웨어의 디버거를 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: