



IBM Software Group

채널사작및실행

Morag Hughson and Paul Clarke

hughson@uk.ibm.com , paulg_clarke@uk.ibm.com

WebSphere. software

@business on demand software

© 2003 IBM Corporation

V1.0.1

IBM Software Group | WebSphere software



사문

N
O
T
E
S

이 문서는 WebSphere MQ 및 MQSeries 에 대한 IBM 의 전문 지식과 제품 사용 중 인 교육 의 경험 과 그 들 의 환경 및 목표 에 대한 경험 을 바탕 으로 IBM 에서 제공 하는 권장 사항 이 다 . 이러한 제언 을 수락 하고 구현 하는 것은 사용 자 가 판 단 해야 하 다 . 다른 전문 인력 이 다른 약 을 들 을 수 도 있 는 다 . WebSphere MQ 에 는 다양한 의견 에 따라 달라 지 며 WebSphere MQ 가 실행 되는 특정 환경 에 따라 다양 해 는 몇 가지 영역 이 있 는 다 .

사용 자 가 WebSphere MQ 에 익숙 해 지 는 후 , 따라 할 제언 을 판 별 할 수 있 게 나 특정 환경 에 적 절 한 의견 에 도 들 을 수 있 는 다 .

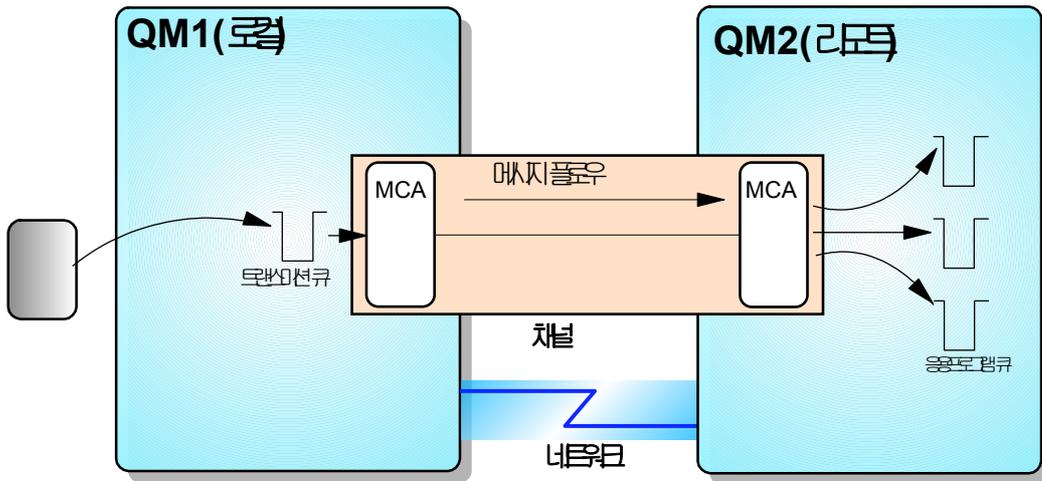
이 문서는 사 템 관 리 자 및 WebSphere MQ 네트 워 크 를 자 원 할 사용 자 를 위 한 것 이 다 . 이 문서는 다음 과 같 은 아 점 을 재 판 한 다 .

- 관리 프 로 세 스 에 알 려 져 서 제공
- 응용 프 로 그 램 의 최 대 기 용 성 재 판
- 초 차 가 능 한 공 용 적 인 실 수 방 지
- WebSphere MQ 전문 카 드 는 기 초 단 계 지원
- WebSphere MQ 프 로젝 트 의 순 조 율 은 사 작 과 성공 보 증

예 에 재 서 된 대 부 분 의 장 보 는 모든 MQSeries 및 WebSphere 버 전 과 관 련 되어 있 는 다 . 매 기 번 시 가 지 않 는 다 지 확 하 면 특정 버 전 에 적 절 한 문 서 를 장 보 하 십 사 요 .

© 2003 IBM Corporation

채널구조



채널구조 설명

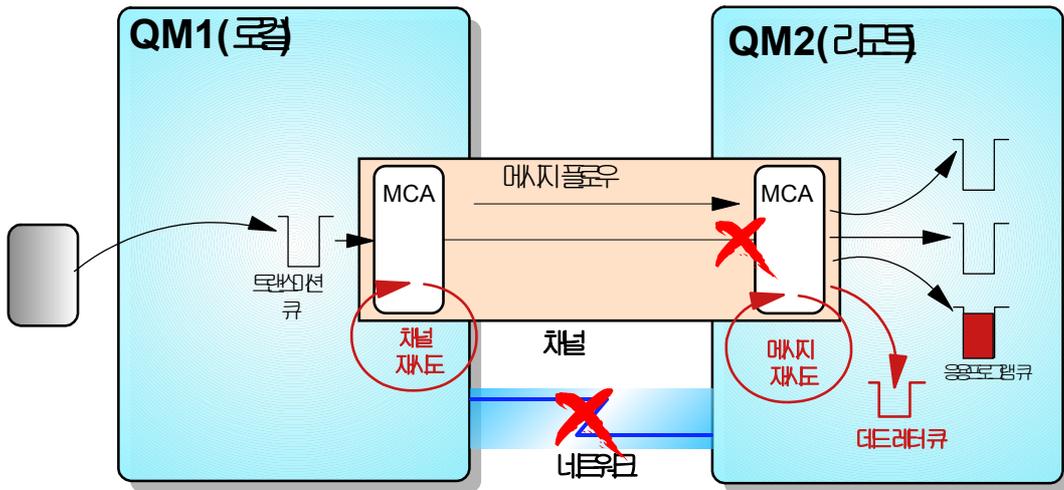
이 다이어그램은 WebSphere MQ 채널 구조를 나타냅니다. 포지셔닝 전에 걸쳐 이 다이어그램을 사용하여 여러 가지 다른 채널 주제를 보여줄 것입니다.

N 다이어그램의 주요 부분은 응용도큐가 목적지선큐에 원격 시스템에 전달될 메시지를 들 수 있음. 원격 큐 정의 또는 명세적 주소 지정 등을 통해 보여줍니다. 이 다이어그램에서 QM1은 로컬 시스템이고 QM2는 원격 시스템입니다. 목적지선큐에서 메시지를 저장하는 상 MCA는 큐로부터 메시지를 받서 채널 정의에 지정된 통신 프로토콜을 통해 원격 시스템으로 메시지를 보내고, 여기서 수신 MCA가 메시지를 받서 적절한 대상 응용도큐에 저장합니다. 이 MCA 쌍을 채널이라고 합니다.

O 별다른 이상이 없어 보이는데 잘못된 경우가 있습니까?

T
E
S

이상이있는경우



- 채널이 종료될수있음- 원격QMgr 종료
- 메시지전달불능- 큐가유한하지않거나기특참
- 네트워크장애
- 성능

이상이있는경우 설명

이그림은오늘논의케될의재미보볼수있습다 이상이발생될수있는몇가지요을간략케실퍼본후에나중에다른그림에서각각보다자세케경호할것입다

NOTES

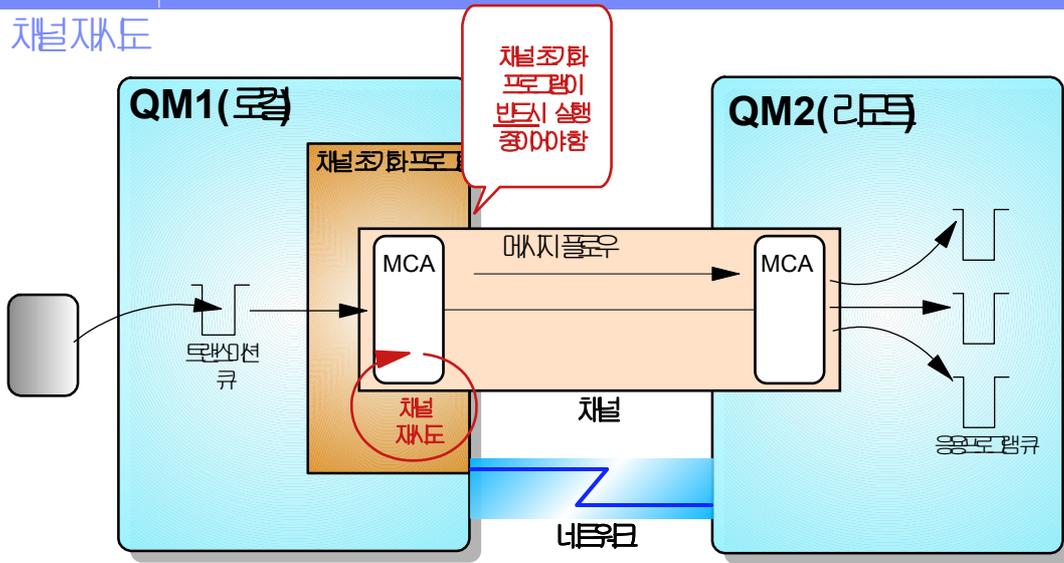
채널이종료될수있습다 예를들어 원격큐메나자가종료된것이원일수있습다 이러한유형의문제를복호화하면채널최 화프로그램을사용여채널을다시시도하면됩니다

메시지전달이불능해질수있습다 예를들어 지정된큐가유한하지않은경우 응용도 램 큐이거나큐가기특참경와같이이시스템문제일수있습다 이러한유형의문제가발생한후에개호하면데드레터큐를사용거나메시지재도가을사용하면됩니다

네트워크장애로인해공을당할수있습다 장애유형에따라채널이이를발호복호하는방법은다함다

성능이문제마될수있습다 따라서몇가지성능고려함을실퍼보았습다

채널재도



```

DEF CHL ....
SHORTRTY(10) SHORTTMR(60)
LONGRTY(999 999 999) LONGTMR(1200)
    
```

채널재도- 설명

채널 작동 여부를 확인하기 위한 디버그 메시지를 전송하는 WebSphere MQ가 채널을 다시 시작하도록 하기 위한 것이다. 이 메시지를 채널 재도요청한다. 채널재도는 송자 서버 또는 클라이언트 송자와 같은 채널 송신 유형에서 사용할 수 있다.

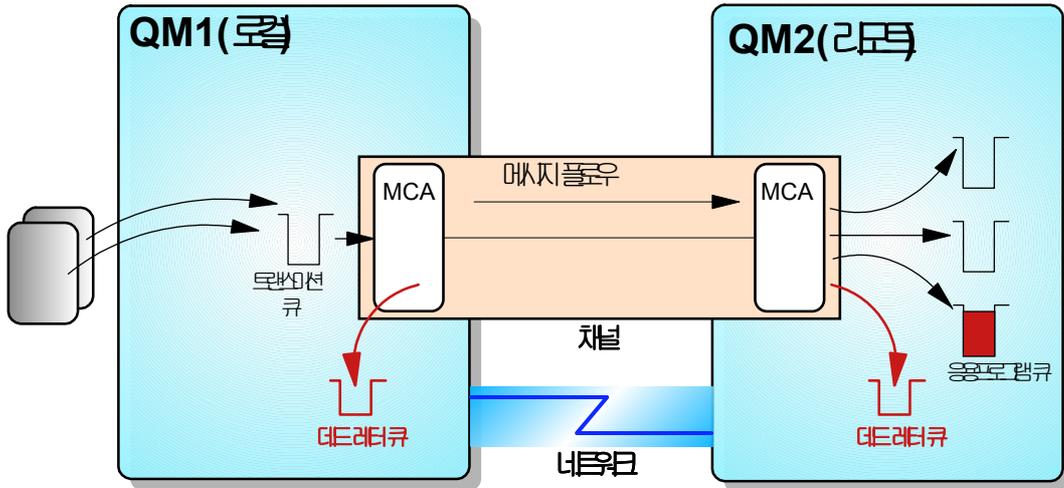
모든 송자 채널은 적절한 재도요청으로 구성되어 있으며 재도요청 메시지를 전송하기 위해서는 채널코화 프로그램이 반드시 실행되어야 한다.

재도요청은 내부요청을 해결할 수 있도록 충분한 시간을 보장해야 하며 예를 들어 2분 동안의 재도는 충분하지 않을 수 있다. 사용자 사용할 수 있는 재도요청은 짧은 재도요청과 긴 재도요청의 두 가지 유형이 있다. 짧은 재도요청은 재도요청이 적은 경우에 사용된다. 즉, 일종의 짧은 응답이 뒤를 이어서 다시 연결한다. 긴 재도요청은 자주 발생하지 않는 재도요청이 많은 경우에 사용된다. 즉, 채널이 종료된 후 이 단축이 작은 응답이 아니라 재도요청하는 경우에 대해 한다.

메시지 전송으로 전송되지 않고 재도요청이 완료된 경우 채널을 다시 시작하려면 수동 개입이 필요하다. 또한 긴 재도요청에 지정된 수는 일종의 작동 중지 상황을 감할 수 있을 정도로 충분한 값이어야 한다는 점을 숙지하십시오.

NOTES

데드레터큐



- 예를 들어, 다음과 같은 메시지의 저장 영역
 - ▶ 주소가 잘못 지정된 메시지
 - ▶ 권한이 없는 메시지
 - ▶ 너무 큰 메시지

데드레터큐 설명

큐에 메시지가 있어서 데드레터큐의 존재는 상당히 중요하다. 데드레터큐가 없으면 유효하지 않은 큐의 지정으로 인해 발생하는 응용도 램의 간헐한 프로래밍 오류(응용도 채널이 작동 중)될 수 있습니다. 여러 개의 다른 응용도 램이 동일한 채널을 이용하고 있으면 분명히 이용할 수 없는 상황입니다.

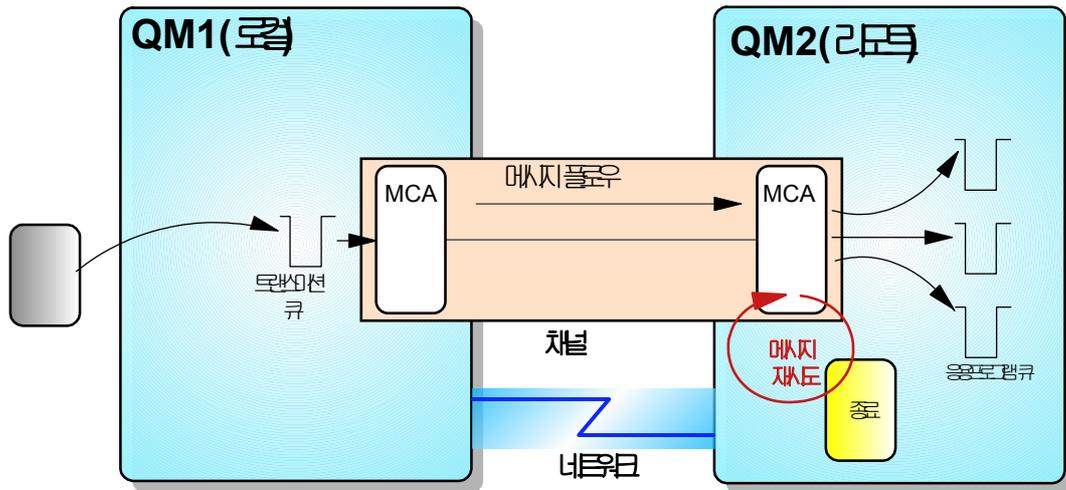
N
O
T
E
S

DLQ는 채널의 어느 한 쪽에서 사용할 수 있는 점을 유의하십시오. 순방향의 DLQ는 특정 메시지가 채널을 통해 전송되기에 너무 큰 경우 사용된다. 메시지 크기가 목표지선 큐에서 사용할 수 있는 정도도 채널에 저장된 메시지 크기를 초과하면 이러한 상태가 발생한다. 수직의 DLQ는 메시지를 대상 큐에 넣을 수 없는 경우에 사용된다. 해당되는 원인은 여러 가지일 수 있으며 일부는 알 수 없다. 메시지 재도매(번역 채널에 대해 장영) 및 메시지 재도중은 이러한 상태를 처리하기 위한 것이다. 메시지 재도매에 관해서는 곧 살펴 보겠습니다.

DLQ의 사용 채널의 어느 쪽에 있지 관계없이)은 메시지 전달 순서에 영향을 줄 수 있는 점을 알아야 한다. 일부 순서에 속하는 메시지에 문제가 있으면 이 메시지는 DLQ에 넣어 메시지 순서가 변경된다.

일부 노린 응용도 램은 메시지를 항상 전송된 순서와 동일한 순서로 전달하도록 요구한다. 이러한 응용도 램은 DLQ를 사용하지 말 것을 요구한다. DLQ가 없으면 채널에 있는 줄지 못한 메시지로 인해 채널이 종료되고 따라서 응용도 램이 메시지를 받지 못하게 된다. 이러한 이유로 인해 현재 DLQ를 포함시키고 때때로 메시지가 순서에 관계없이 전달되는 경우에 대응할 수 있도록 응용도 램을 수정할 것을 권장한다. 이러한 수정은 어렵지 않으며 상반 ID를 기본으로 사용하거나 2차 큐를 만들거나 메시지 그룹을 사용하는 등과 같은 다양한 방법을 채택할 수 있습니다.

메시지재도



```
DEF CHL ....
MRRTY(10) MRTMR(1000)
MREXIT(msg-retry-exit-name) MRDATA()
```

메시지재도- 설명

메시지재도사용특히 더드래터큐를사용하지않고실행하는경우를생해보십시오

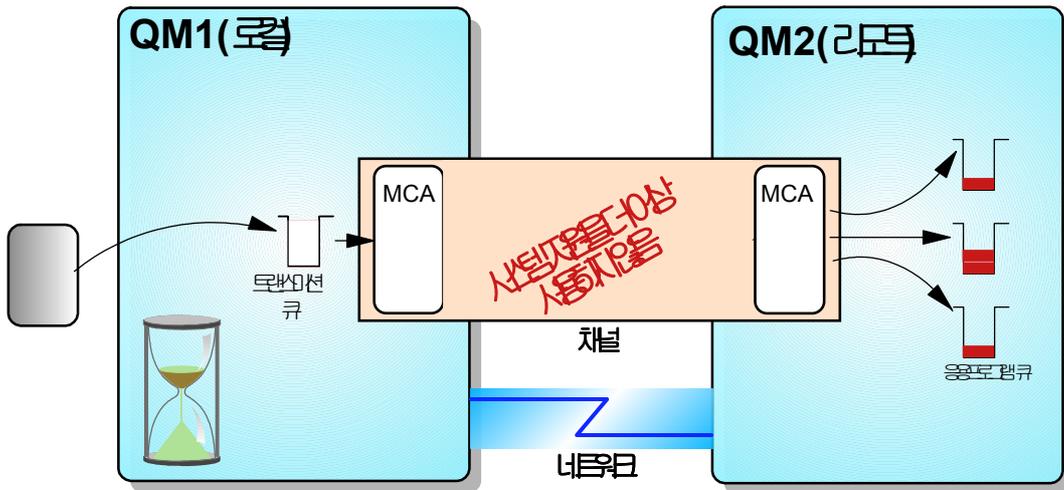
메시지재도메커니즘은중요와함께사용되거나중요없이사용할수있습니다.중요가없으면수신MCA가간헐타실파한날(put)를간헐메시지
 시도합니다.재도사이의시간과재도횟수에대한많은채널장애있습니다.중요에서용될경우이숫자전달도지만양위값으로변환할수있습니다.
 중요메시지재도메커니즘과함께사용면실파를보다인정적으로채할수있습니다.예를들어,대량큐에메시지를넣을수없는경우방향을재지정할수
 있습니다.

중요사용각메시지를버탕으로재도값을내릴수있습니다.

z/OS에서는현재메시지재도가지원되지않습니다.

N
O
T
E
S

연결기격력



DEF CHL DISCONT(6000)

- **모니터링과 결합**



연결기격력- 설명

연결기격력 또는 DISCONT는 보낼 메시지가 없어도 채널을 계속 실행시켜야 하는 시간을 제어하는데 사용하는 송신 유형 채널에 대한 채널 속성이다.

사용자가 설정하는 값은 몇 가지 기준에 따라 다르다.

- 가져오는 채널 수 및 시스템 자원의 중요도
- 네트워크의 신뢰도
- 통신 세션의 비용

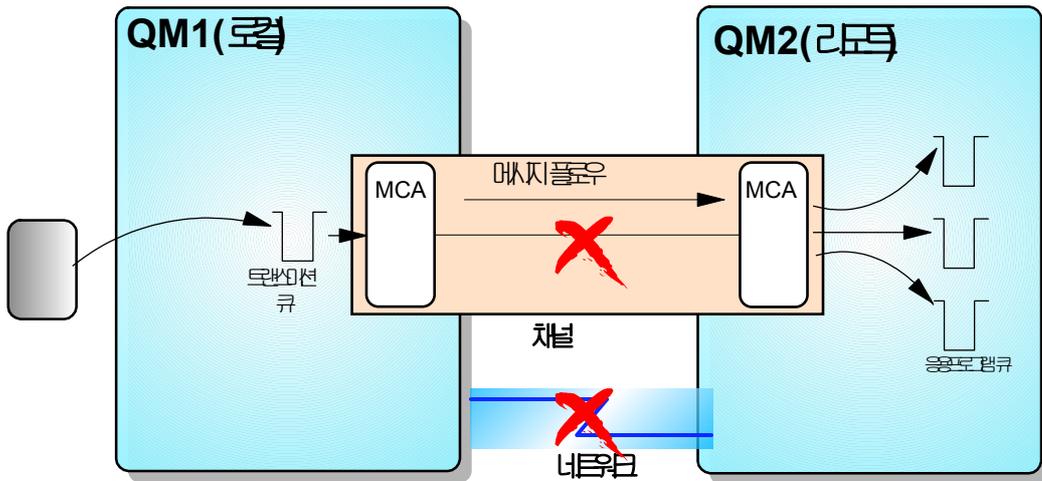
이 값은 상대적이며 상대 메시지의 평균 시간보다 길어야 한다. 즉 상대에게 전달할 작업이 없을 때 채널의 연결이 끊어짐을 의미한다.

채널의 연결기격력 사용을 채널 트리거링(triggering)과 결합하여 채널을 다시 시작해야 한다. 이러한 방식으로 수할 작업이 있으면 채널이 자동으로 시작되고 수할 작업이 더 이상 없으면 채널이 자동으로 종료된다.

**N
O
T
E
S**



네트워크장애



- 허브트
- TCP/IP Keepalive
- MCA 채택



네트워크장애- 설명

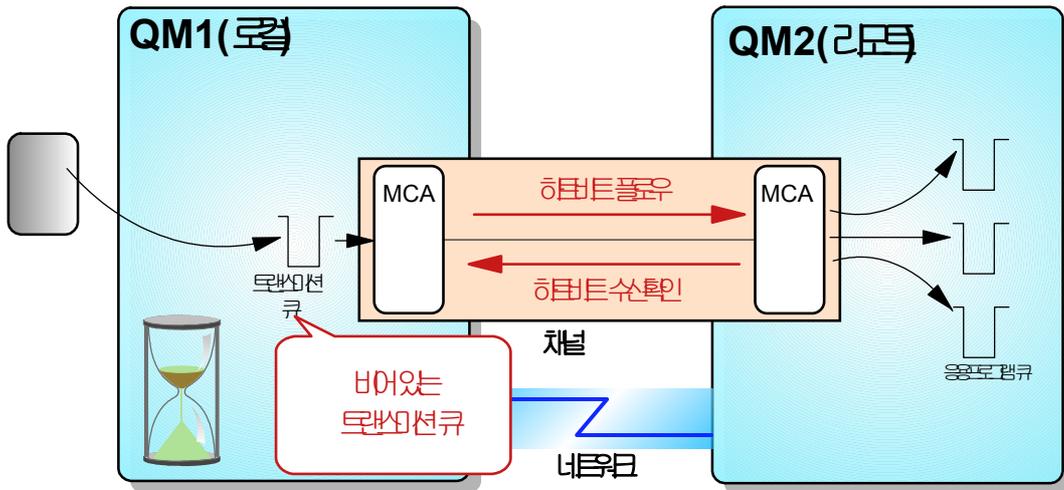
채널에 보던 네트워크 장애는 채널을 종료합니다. 네트워크 장애가 채널에 보고되지 않으면 채널이 종료되지 않고 네트워크 장애를 감지할 수 없습니다. 채널은 낮은 레벨의 통신 기능에 의존하여 네트워크 작동 중지를 포함합니다. 예를 들어 RECEIVER 채널의 휴식 상태가 rec 호출에 달려 있습니다. rec 호출이 도출되지 않으면 채널은 상대방의 연결이 끊긴 것을 알 수 없습니다. MCA 사용에 있어서야 온점중하는 통신 프로토콜이 항상 올바른 시간에 오류를 반환하는 것은 아니므로 네트워크 실패로 실패한 점을 적절하게 인식하는 것입니다. 일반적으로 LU6.2가 TCP/IP 보다 이 방향에서 뛰어난 (보다 복잡한 프로토콜).

MCA의 다양한 향상으로 상당히 좋은 상황으로 개선했습니다. 다음 몇 개의 그림에서 이 중 알를 살펴 보겠습니다.

NOTES



허브트



DEF CHL HBINT(300)

버퍼해제

큐달기

허브트- 설명

허브트간격 또는 HBINT는 채널의 송신기가 보낼 메시지가 없을 때 수신자가 양호한 상태인지 점검해야 하는 빈도를 제어하는 채널 속성이다. 값은 채널의 양쪽에서 초단위로 지정되며 양쪽은 채널 시작 시 사용할 적절한 값을 조정한다. 이 값은 지정된 두 개의 간격 중 반보다 낮은 간격이다. 한쪽 또는 양쪽이 0(허브트없음)을 지정하면 이 값은 반보다 가장 낮은 값이므로 이 값을 선택한다.

N
O
T
E
S

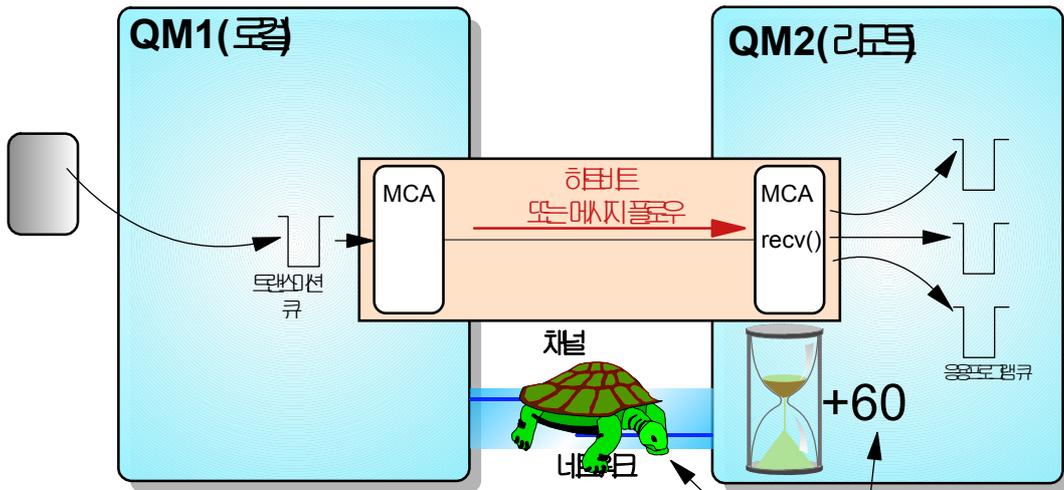
가장 낮은 채널은 허브트 간격이 300 초와 5 초로 지정된다. 네트워크에 대한 영향을 최소화하기 위해 상대적으로 높은 값을 가장 낮은 값으로 선택한다. 실제로 낮은 값을 들어 대부분의 경우 60 초 정도가 적당하다.

이 허브트는 사실 수신 MCA가 계속 사용 가능하지 점검하기 위해 수신 MCA에 보내는 짧은 메시지이다(또는 MQGET 중 CLINTCONN). 최하 허브트 간격은 동인 채널에 다른 활동이 없는 경우에만 메시지를 보낸다. 이러한 메시지는 두 가지 용도를 해한다. 송신 MCA는 MQGET에 의존하지 않고 네트워크로부터 즉각적인 오류 리턴을 받을 수 있다. 또한 수신 MCA가 응답을 보내기 전에 큐에 저장된 정보와 같은 일부 채널을 수신할 수 있도록 해준다.

고에 채널은 메시지를 주적으로 전송하지 않으므로 채널은 지정 개시 및 쿨을 유지하지 않도록 선택한다. 따라서 허브트 시에 채널은 불필요한 지정 버퍼 및 메시지 양을 해하고 개시 쿨을 감소한다.

그러므로 성능 면에서 허브트 값이 너무 작지 않아야 한다. 그러나 낮은 허브트 값은 채널이 계속 사용 중일 때 쿨 네트워크 용을 발생시키지 않는다는 점을 참조하십시오. 채널을 통해 2 초마다 메시지를 보내는 응용 프로그램을 생각해 보십시오. 보낼 실제 메시지가 항상 있으므로 낮은 허브트 값 3 초보다 허브트 값 지정되지 않는다.

수신대기시간



```
DEF CHL .... HBINT(300)
MQRCVBLKTO: +60
RCVTIME=+60
```

네트워크 대기 시간 반영



수신대기시간 - 설명

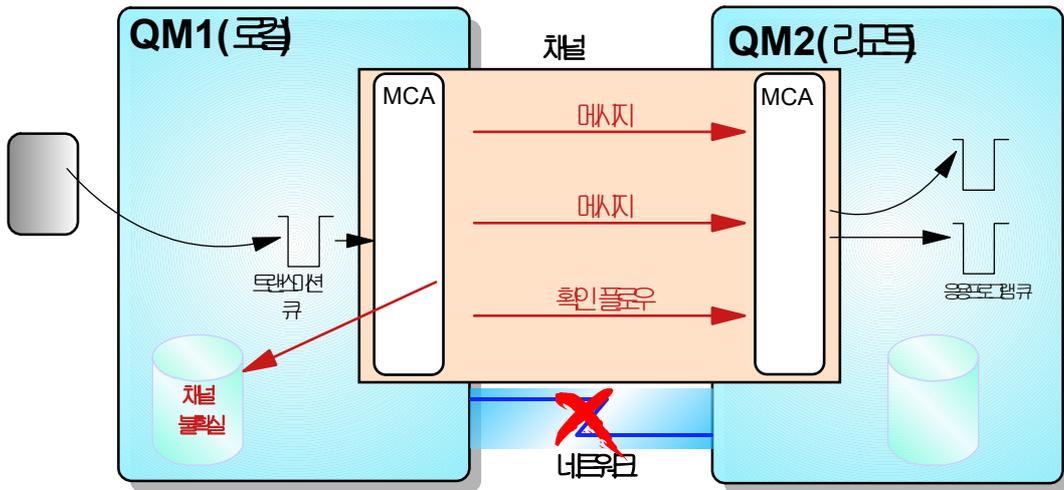
NOTES

이미언환과 마찬가지로 채널의 송측에서 허턴트 메시지를 정적으로 보지 못하게 하는 허턴트 간격은 양쪽이 조정된 값(로양쪽) 값을 알고 있습니다. 허턴트가 없으면 메시지를 전달할 수 없으므로 채널의 수신자 측은 채널을 통해 다음 데이터 조각이 도착하는 시를 예측할 수 없습니다. 그러나 허턴트를 사용하면 수신자는 허턴트 간격 내에 메시지가 도착했는지 또는 허턴트가 도착했는지 알 수 있습니다. 도착하지 않은 경우 통신 세션이 종료된 것입니다. 이를 TCP/IP 네트워크를 통한 연결을 기준으로 사용합니다. CLNTCONN 채널의 경우 MQI 명령에 대한 응답을 기다리는 동안에도 시간을 사용하지 않습니다(SVRCONN이 허턴트의 클라이언트 변형을 지원하지 않는 z/OS에 있는 경우 제외).

수신자는 통신 장애가 있는 것으로 가정하기 전에 기본적으로 조정된 허턴트 간격이 60초 미만인 경우 허턴트 간격이 두 배 간격 이내 또는 조정된 허턴트 간격이 60초 이상인 경우 조정된 허턴트 간격을 넘어 60초 내에 데이터가 수신되지 않으면 실제로 시간 종료됩니다. 이 규정은 실제로 환경 변수 MQRCVBLKTO를 변경해서 분산 풀링에서 네트워크 대기 시간을 반영하여 조정할 수 있습니다. 허턴트에 따라 시간 종료 값을 계산할 때 사용할 최소값을 설정하는 데 사용할 수 있는 또 다른 환경 변수 MQRCVBLKMIN이 있습니다. XPARM 속성 RCVTIME를 사용하여 시간 종료 값을 규정요 RCVTMIN을 사용하여 최소값을 설정하여 z/OS의 V5.3.1에서 적용을 설정할 수 있습니다.

따라서 작은 값은 장애를 보다 빠르게 발견하지만 채널이 활발하게 되지 않은 경우 네트워크 통신 및 CPU 자원 측면에서 비용이 많이 들 수도 있습니다.

전달확인



- 양쪽 디스크에 기록된 동화 데이터
 - ▶ 네트워크 장애 이후 자동 화해용
 - ▶ 불확실 일괄처리에 있는 메시지는 클러스터링 알고리즘으로 재발할 수 없음



전달확인 설명

NOTES

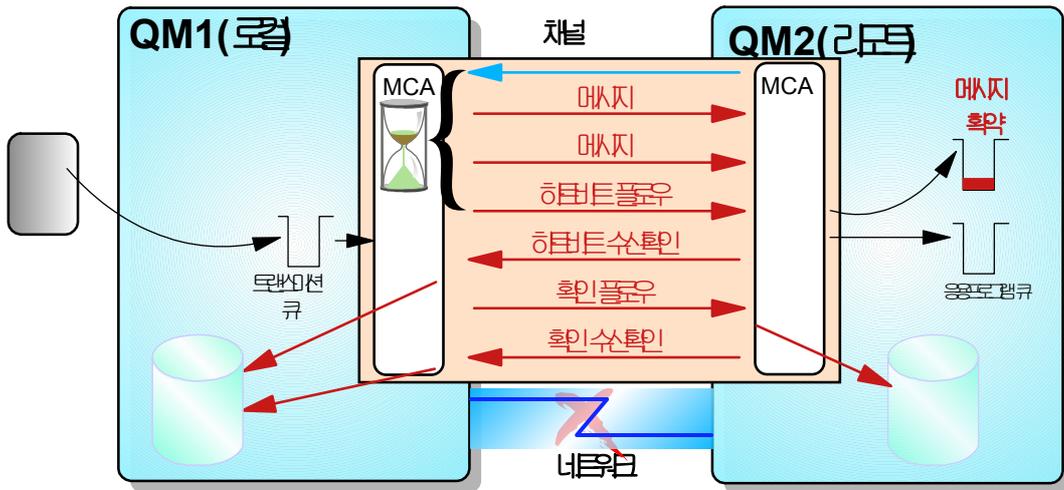
예에서 메시지 일괄처리를 전하는 채널의 양쪽을 볼 수 있습니다. MCA가 일괄처리 끝에 있는 디스크에 데이터를 기록하는 방법을 보십시오. 복구 가능한 일괄처리의 경우에만 이렇게 수행됩니다. 기록된 데이터에는 상에서 모든 메시지를 감시기 위해 사용한 목적지의 목적지 ID가 들어 있습니다. 상에서 상대방에게 확인 필로우를 받았으면 응답을 받을 때까지 불확실 상태입니다. 즉 상에서 채널 메시지가 정상적으로 전달되었는지를 확인할 수 없습니다. 이 단계 중 동기 상태에 있는 경우 채널은 불확실 상태로 종료됩니다. 상대방에게 다시 연결할 때 데이터 저장소에서 상대방과 불확실 상태에 있었음을 확실히 되도록 다른 채널에 마지막 메시지 일괄처리를 전달 여부를 묻습니다. 상대방이 보낸 응답을 사용해서 채널은 목표지선 큐에서 메시지의 확인 또는 확실히 종료할 수 있습니다.

이동화 데이터는 DIS CHSTATUS(*) SAVED 명령을 실행하면 볼 수 있습니다. 포함된 채널의 양쪽에서 동일해야 합니다.

채널이 불확실한 상태에 있을 때 채널이 다시 시작될 경우 자동으로 불확실 상태를 해결하는 점을 참고하십시오. 그러나 동일한 상태와 대립하고 있는 경우에만 가능합니다. 채널 속성이 변조되거나 다른 큐에 대해 P 주소를 인계했거나 동일한 목적지선 큐에서 메시지를 전송하는 다른 채널이 시작될 경우 채널은 즉시 종료되며 다른 큐에 대해 계속 불확실 상태를 나타내는 메시지가 포함됩니다. 사용자는 윌트 큐에 대해 방향을 지정하여 채널을 시작하거나 RESOLVE CHANNEL 명령을 실행하여 불확실 상태를 수동으로 해결해야 합니다. 이 경우 사용자는 DIS CHS(*) SAVED의 출력을 사용하여 윌트 조치 COMMIT 또는 BACKOUT이 선택되었는지 확인해야 하는 점을 유의하십시오.



일괄처리허브트



DEF CHL BATCHHB(500)

- 마지막통신이후시간이간격보다큰경우허브트플류가전송됨

내류큐인정성변명



일괄처리허브트- 설명

불활성상태로 종료된 채널의 문제점은 쿼리시 링알고리즘을 통해 확인할 수 있습니다. 쿼리시 링 채널이 비정상적으로 종료된 경우 이 채널이 전송하려고 했던 메시지가 다른 채널 및 대상 클러스터 큐의 다른 인스턴스로 재전송됩니다. 그러나 송신자 채널이 수신자 채널의 수신여부가 불확실한 메시지 일괄처리일부인 경우 재전송할 수 없습니다.

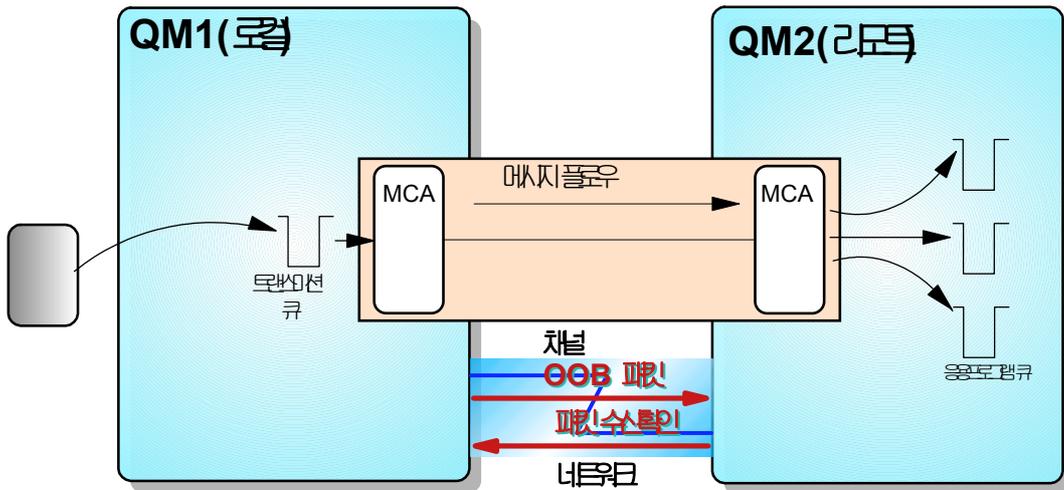
불활성상태에서 채널이 종료될 가능성을 줄이기 위해 채널 프로그래머가 플류가 추가되었습니다. 이 플류를 일괄처리허브트라고 합니다. 이 플류는 일괄처리에 대한 메시지가 전송된 후 채널이 불활성상태로 재전송되고 확인 플류를 보내기 전에 전송됩니다. 채널의 수신자로부터 응답을 받으면 일괄처리를 완료할 수 있음을 상당히 확인할 수 있습니다. 이제 수신자 채널이 종료되고 불활성상태에 머무를 가능성이 매우 적습니다. 일괄처리허브트 플류를 사용하지 않으면 이러한 가능성은 훨씬 커집니다. (상대방으로부터 마지막 플류를 받은 이후 시간까지).

이 일괄처리허브트 플류가 전송되었는지 여부를 판별하기 위해 상대방으로부터 플류를 마지막으로 받은 시간을 확인할 수 있습니다. 이 시간이 채널에서 BatchHeartbeat 또는 BATCHHB 매개변수에 지정된 간격보다 상당히 길면 플류를 보낼 것이고 그렇지 않으면 보낼지 않습니다. 따라서 지정된 간격은 내류큐의 인정을 반영하는 것입니다. 이 간격은 알 세드 단위로 지정합니다.

NOTES



TCP/IP Keepalive



▪ QM.INI **KEEPALIVE=YES**

▪ XPARM **TCPKEEP=YES**

DEF CHL ... KAIN(T(360) (z/OS만)



TCP/IP Keepalive - 설명

TCP/IP에는 KeepAlive라고 하는 자체의 허트비트 프로토콜도 있어서 네트워킹 상태를 안할 수 있습니다. KeepAlive의 사용이 권장됩니다. z/OS에서 채널 매개변수로 구성할 수 있으며 다른 채널이 특정 시간 종료 값을 가지거나 다른 허트비트 메시지를 사용할 수 있습니다. 이 채널 매개변수는 KeepAlive를 사용하지 전에 TCP/IP에서 on으로 설정할 것을 계속 요구합니다.

모든 플랫폼에서는 동일한 qm.ini 파일에서 WebSphere MQ의 큐에 저가점으로 구성할 수 있습니다. 이 경우 더 머릿글에서 사용되는 실제 값은 시스템의 전체 TCP/IP 스택에 대해 설정됩니다. 알바 값은 기본적으로 두 시간 이후에 네트워킹 상태를 갱신하는 것입니다. 이 값은 WebSphere MQ 환경의 경우 너무 높을 수도 있지만 동일한 시스템의 다른 응용 프로그램이 KeepAlive를 사용하려 할 수도 있으므로 값은 이러한 여러 응용 프로그램의 요구 사항을 절충한 것이어야 합니다.

KeepAlive를 구성하는 방법을 알면 TCP/IP 매개변수를 알아야 합니다. AIX의 경우 "no" 명령을 사용하십시오. HP-UX의 경우 "netttune" 명령을 사용하십시오. Windows NT의 경우 레지스트리를 편집해야 합니다. z/OS에서는 TCP/IP PROFILE 데이터 세트를 갱신하고 TCPCONFIG 섹션에서 INTERVAL 매개변수를 추가 또는 변경하십시오.

주 z/OS에서는 TCP/IP APAR PQ75195가 권장됩니다.

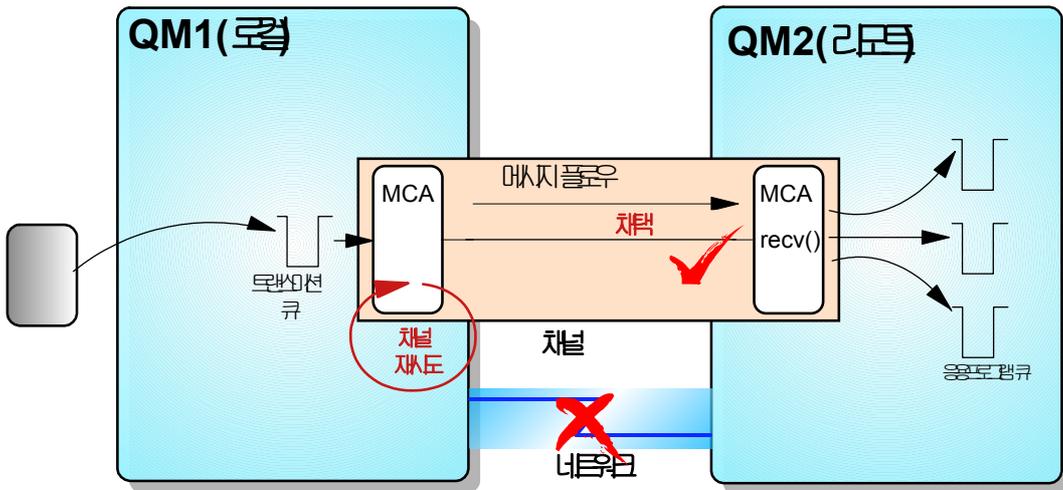
허트비트 및 수신 대기 시간 종료 사용면 양쪽이 두 응용을 모두 지원할 때 KeepAlive가 필요하지 않게 되지만 KeepAlive를 계속 사용할 수도 있습니다. 허트비트 및 종료 없는 애플리케이션을 모두 지원하지 않는 WebSphere MQ 구현과 통합할 때 KeepAlive를 계속 사용해야 합니다.

클라이언트 허트비트 사용 가능한 경우에도 MQGET 중재만 사용하므로 KeepAlive는 특히 SVRCONN 채널에 권장됩니다.

NOTES



MCA 채택



- QM.INI **AdoptNewMCA, AdoptNewMCATimeout, AdoptNewMCACheck**
- XPARM **ADOPTMCA, ADOPTCHK**

MCA 채택- 설명

예를 들어 통신 오류로 인해 채널이 실패하면 채널의 양쪽이 동시에 실패를 발견하는 보장 없습니다. 채널의 송신측은 소켓을 통해 메시지를 보내려고 시도하는 쪽이므로 실패를 먼저 발견할 가능성이 높습니다. 따라서 송신측은 대상 큐메저에 다시 연결하려고 시도할 것입니다. 대상 시스템에 연결이 들어오면 큐메저는 이 이름로 이 위치에서 실행 중인 채널 이미 있음을 확인합니다. 결국 새 연결은 거부됩니다.

NOTES

AdoptNewMCA를 구성하면 새 연결이 도착했으며 동일한 네임과 주소와 동일한 큐메저에서 실행 중인 이름의 채널 이미 있는 경우 기존 채널을 종료하고 새 인스턴스로 바로 큐메저에 재채널할 수 있습니다. 다시 연결해서 새 연결에서 채널 인스턴스를 채택해야 합니다.

QM.INI의 채널 선택에 있는 다양한 구성 매개변수 또는 z/OS에서는 XPARMS)을 통해 채택 작동에 관한 변경할 수 있습니다. 채택을 자동화하려면 이러한 방식으로 채택할 수 있는 채널 유형의 목록을 설정하십시오.

AdoptNewMCA=SVR,SDR,RCVR,CLUSRCVR,ALL,FASTPATH

ADOPTMCA=YES/NO(z/OS에서)

'채택된 채널의 세밀한 종료와 즉각적인 종료 시도의 대기 시간을 설정하십시오.

AdoptNewMCATimeout=60

분할 채널 이름은 알해야 하지만 주된 네임과 주소와 동일한 리모 큐메저 이름도 동일하도록 지정할 수 있습니다. 그러면 네임에 있는 다른 시스템의 잘못된 연결 이미 실행 중인 채널을 채택하지 못합니다. 채택된 연결과 새 연결 시도에 알해야 하는 속성을 설정하십시오.

AdoptNewMCACheck=QM,ADDRESS,NAME,ALL

ADOPTCHK=NONE/QMNAME/NETADDR/ALL(z/OS에서)

상

- 생각해볼몇가지질문
 - ▶ 도플이전체일괄처리를조절할까?
 - ▶ 요구되는응답시간은?
 - ▶ 일괄처리를전송하는데소요되는시간은?
 - ▶ 채널의시작시간은?
- 사용자의환경에따른답
 - ▶ 응용프로그램사용량
 - ▶ 운영체제하웨어
 - ▶ 네트워킹대역폭
 - ▶ 메시징기

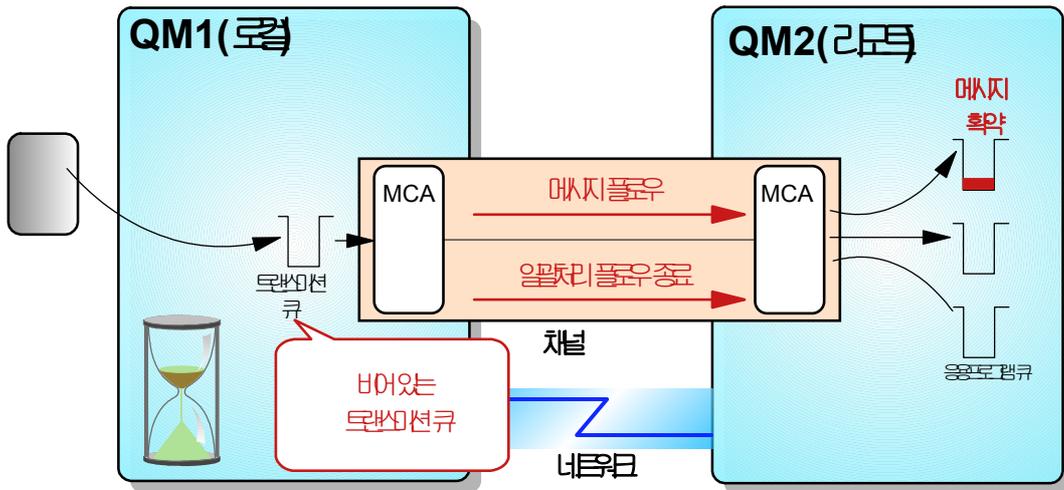


상구성

- **BatchInterval**을사용하여일괄처리계속열기
 - ▶ 도플이전체일괄처리를전송하는것보다약간낮은경우
 - ▶ 대기시간이증될수있지만전체처리는행됨
- 채널을계속보존하기위해**DisconnectInterval**을충분히높게설정
 - ▶ 채널이시작중지재시작되는횟수가줄어듦
 - ▶ 네트워킹"활성세션"의요금을부하하거나**MaxChannels** 한계에도달하지않은경우
- 세가지사안요
- 신뢰버안딩사용
- 용량계획안내서참조
 - ▶ SupportPac MP00
 - ▶ SupportPac MP16



일괄처리간격



- 메시지 일괄처리의 최소 수명

DEF CHL ... BATCHINT(100)

▶ 복구 가능한 메시지는 BATCHINT 값이지만 즉시 도착함



NOTES

일괄처리간격- 설명

복구 가능한 메시지 일괄처리는 목적지선 큐 버어있으면 즉시 종료된다. 따라서 호전 전 및 디스크 할당이 필요 이상 많지 않을 수 있다. 세가지 자속인 메시지를 목적지선 큐에 넣는 응용도 램을 생해 보았을 때 두 번째 메시지를 넣기 전에 채널이 첫 번째 메시지를 감할 수 있는 좋은 기회이다. 결국 채널은 일괄처리를 완료하고 두 번째 및 세 번째 메시지는 실제로 두 번째 일괄처리로 들어간다.

일괄처리에 예를 들어 BATCHINT(100)과 같이 작은 자연 수를 추가하면 절의 디스크 할당으로도 세가지 메시지를 모두 첫 번째 일괄처리에서 보낼 수 있다. 최소한 100ms 가 자연수로 메시지는 전달하는데 보다 많은 시간이 걸린다. BATCHINT는 비자속인 메시지

NPMSPEED(FAST)에 영향을 주지 않는다. 이러한 메시지를 송점 바깥에서 넣기 때문이다. 보통 응용도 램에서 데이터베이스 큐는 시간이 중요한 비자속 메시지로 보며, 데이터베이스 갱신은 시간이 비교적 중요하지 않은 자속 메시지로 보낸다.

BATCHINT의 전체적인 효과값은 MQ 메시지 전달 네튜워크 및 메시지 전달 프로세스에 따라 매우 다르다. 시스템에 있는 채널 수가 많을수록 전체적인 이점이 증한다.

BATCHINT의 이점이 많았을 때 성능을 개선하는 데 도움이 되는 방법은 효과적인 일괄처리 코드를 확인하는 것이다. DISPLAY CHSTATUS를 실행하고 메시지 수를 일괄처리 수로 나타내시오. 이 수가 매우 낮으면 (즉, 1과 2 사이) 채널은 하나 또는 두가지 메시지 각에 대해 획득해야 하는데 이는 비효율적이며 BATCHINT의 이점을 볼 수 없다. 모든 메시지 비자속 메시지인 경우 BATCHINT가 전혀 도움이 되지 않으므로 어떤 하의 파일 뿐이다.

또한 획득해야 할 것은 채널 프로세스의 디스크 사용량이다. 디스크 사용량이 많은 경우 BATCHINT가 이 비용을 줄고 전체 채널을 향상시킬 좋은 기회이다.



시나리오1

- 사전로드된 트랜잭션 큐의 다양 전송
 - ▶ 큰BatchSize 사용 가능값500면적당
 - 매우 많은 메시지를 사용하지 않는 경우 자원 낭비
 - ▶ BatchInt는0이어야함



시나리오1 - 설명

첫 번째 성능 조정 시나리오는 사전로드된 트랜잭션 큐에서 메시지를 다양 전송하는 것입니다.

여기에서는 큰 일괄 처리로 메시지를 보내어 메시지당 일괄 처리 중료 비용을 줄이는 것이 목표입니다. 따라서 큰 일괄 처리 크기를 사용해야 하며 (여기에서는 가능값 500면적당) 일괄 처리 간격은 0으로 설정해야 합니다. 모든 메시지는 트랜잭션 큐에 사전로드되어 시나리오에서는 일괄 처리를 열어 놓을 필요가 없습니다.

성능 균형은 내부 큐 연결이 도중에 끊어진 경우 전체 일괄 처리를 자원 낭비 비용과 비교하여 메시지당 일괄 처리 중료 비용을 줄기 위해 하나의 일괄 처리에서 보내야 하는 메시지 수와 관련된 것입니다. 매우 많은 메시지가 도착하는 경우 일괄 처리 크기를 50보다 작게 하여 균형을 유지해야 합니다.

N
O
T
E
S

시나리오2

■ 지연된 채에 대한 소량 전송

- ▶ 정확한 숫자는 도착률에 따라 다름
- ▶ 높은 일괄처리 수행 속도
- ▶ 전송하기 전에 BatchInt를 적절한 지연값으로 설정하여 1분
- ▶ 크거나 유크지 않아야 함 그렇지 않으면 오래 실행되는 UoW가 오래 실행됨



© 2003 IBM Corporation

시나리오2 - 설명

두 번째 성능 조정 시나리오는 지연된 채에 대해 메시지를 소량 전송하는 것입니다.

여기서는 메시지의 합적인 일괄처리 크에이르는 것이 목표이므로 메시지당 일괄처리 종료비용을 줄게 됩니다. 이 방법은 일괄처리 간격을 사용하는 중의 방법입니다. 메시지는 채널이 채워할 수 있는 것보다 약간 느리게 도착합니다. 약간 오래 일괄처리를 열어서 더 많은 메시지를 함께 일괄처리할 수 있습니다.

상능 균형은 복구 가능한 메시지가 일괄처리가 완료될 때까지 즉, 최한 일괄처리 기억이류에 대상 큐에서 확보되지 않으므로 전달하는 데 시간이 오래 걸린다는 사실과 비교하여 하나의 일괄처리에 보낼 수 있는 메시지 수에 대한 것입니다.

전달 시간이 중요하지 않은 경우 오래 실행되는 작업 단위를 생성할 위험이 있으므로 일괄처리 간격을 너무 크게 설정하지 않도록 주의하십시오.

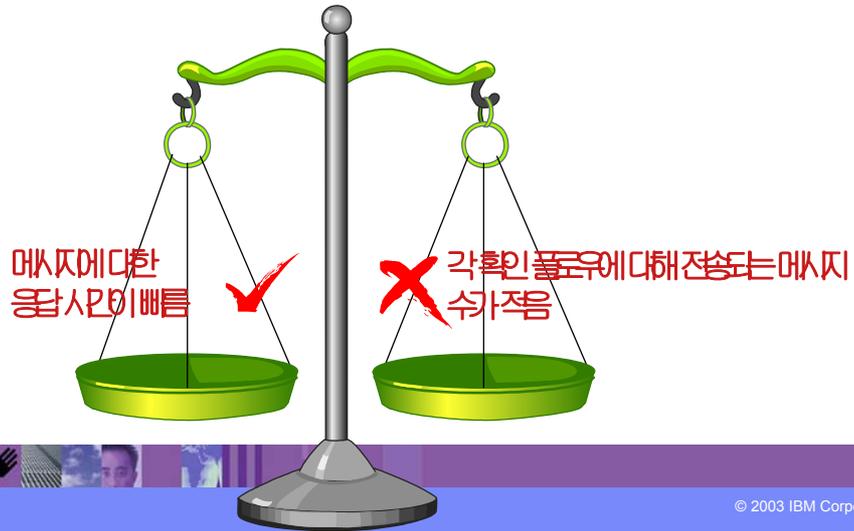
N
O
T
E
S

© 2003 IBM Corporation

시나리오3

■ 동기요청응답

- ▶ 응답시간이 중요한 경우 BatchInt를 사용하지 않을 수도 있음
- ▶ 자취적 부하가 낮은 경우 BatchSize를 작게 설정
- ▶ 비자취적 메시지 플로우의 경우 NPMSPEED(FAST) 및 0이 아닌 BatchInt 사용



시나리오3 - 설명

세 번째 성능 조정 시나리오는 응답시간이 중요한 동기요청응답 모델입니다.

시나리오2와 비슷한 상황이지만 이번에는 하나의 알플라에 있는 많은 메시지보다 빠른 응답시간이 균형에 중요합니다.

N 자취적 메시지 부하가 낮으면 보다 작은 알플라 크기 또는 동일 알플라 크기를 전혀 사용하지 않거나 메시지가 사면에 따라 사용 가능한지 확인하기 위해 몇 초 정도의 매우 낮은 값을 사용해야 할 것입니다.

O 비자취적 메시지가 있는 경우 빠른 채널을 통해 전송되는 비자취적 메시지는 즉시 사용 가능하므로 비자취적 메시지 속도 또는 NPMSPEED FAST를 사용하여 0이 아닌 알플라 크기를 설정해야 합니다. 알플라 크기 값은 계속 사용되지만 응답시간에 영향을 주지 않습니다.

몇 가지 수를 예로 들겠습니다.

T 예상 부하가 초당 30개의 비자취적 메시지인 경우 BatchInt=2000(2초)을 설정하고 효과적인 알플라 크기가 50이므로 BatchSize=50을 설정하십시오.

E
S

신뢰바인딩

- 단일 프로세스
 - ▶ MCA와 로컬 큐메저에 대한 단일 프로세스
 - ▶ MCA MQI 명령에 대한 PC 요해가 없음
 - 비차적 메시지의 경우 중요

- 큐메저 ini 파일에 구성됨
 - ▶ 채널 스캔자

MQIBINDTYPE=FASTPATH

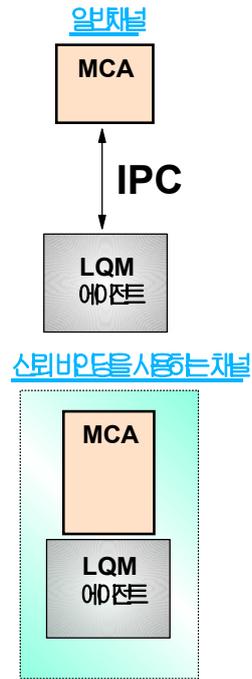
- 채널 중단을 테스트하는 경우 사용되지 않음
- 다음 명령 실행 시 주의

STOP CHL() MODE(FORCE)

(V5.3 이전)

STOP CHL() MODE(TERMINATE)

V5.3



신뢰바인딩- 설명

NOTES

가볍고 분산된 제품 세트에서 모든 큐메저 작업은 실제로 에이전트로 하는 응용 프로그램과 별도의 프로세스에서 수행한다. 각 API 호출은 일종의 IPC 메커니즘을 통해 에이전트 프로세스로 보낸다. 이러한 작업은 차단 상태에서 수행된다. 응용 프로그램은 큐메저 지원 또는 메모에 직접 액세스할 수 없다. 그러나 각 MQI 호출은 프로세스 스위치로 성능에 확실히 영향을 준다. 비차적 메시지의 경우 호출 비용의 약 60% 정도가 프로세스 스위치에 있다.

모든 응용 프로그램은 신뢰 모드에서 실행하도록 선택할 수 있다. 즉 큐메저가 응용 프로그램에 따라 직접 실행할 수 있음을 의미한다. 일반적으로 응용 프로그램이 모드에서 실행하지 않는 것이 좋다. 그러나 채널은 제품 코어로 빠른 경로로 신뢰 응용 프로그램으로 실행하는 것이 적절하다. 정상 응용 프로그램으로 실행하는 것이 가장 빠르지만 QM.INI 파일의 채널 스캔자에서 MQIBindType=FASTPATH를 설정하면 모든 채널을 신뢰하는 것으로 실행한다.

채널 중단을 사용할 때에는 주의해야 한다. 채널 중단은 MCA에 따라 실행되므로 중단은 이제 큐메저 에이전트와 동일한 메모리 공간에 있다. 이 모드에서 채널 중단을 실행하기 전에 철저하게 테스트해야 한다.

V5.3 이전 릴리스에서 STOP MODE(FORCE) 명령을 사용할 때에도 주의해야 하며 V5.3에서 STOP MODE(TERMINATE) 명령을 사용할 때 주의해야 한다. 둘 중 어느 경우든 채널이 종료될 때까지 기다려야 완전히 수행되지 않을 수도 있으므로 큐메저가 손상되거나 정지될 수 있다.



메세지수집조

채널	설명
HBINT	허브트고역
BATCHSZ	일괄처리에는최대메시수
BATCHINT	메시지일괄처리최소수명
DISCINT	채널이종료되면경계이하는고역
SHORTRTY, SHORTTMR LONGRTY, LONGTMR	채널재도계수및타이머
MRRTY, MRTMR MREXIT, MRDATA	메시지재도계수및타이머종료z/OS에서는해당항없음

QM.INI - 채널스캔자	XPARMS	설명
AdoptNewMCA	ADOPTMCA	채널유형채택
AdoptNewMCATimeout		정지시간종료채택
AdoptNewMCACheck	ADOPTCHK	메세지수집조채택
KeepAlive	TCPKEEP	KeepAlive 사용
MQIBindType		채널FASTPATH 비인딩사용

환경변수	XPARMS	설명
MQRVCBLKTO	RCVTIME	수신대기시간종료
MQRVCBLKMIN	RCVTMIN	수신대기최소시간종료



메세지수집조 설명

이모듈은 성능 및 복구 용에 영향을 주는 채널의 속성을 나열한 것입니다.

일부 속성은 실제 채널 정의 일부이고 다른 속성은 INI 파일과 같은 다른 위치에서 설정됩니다.

N
O
T
E
S

채널 정의에 있는 메세지수집에 대한 자세한 정보는 InterCommunication 서적에서 찾을 수 있습니다.

분산 플랫폼에서 QM.INI 파일에 있는 메세지수집은 채널 스캔에 있으며 Windows에서는 레지스트리에서 구현됩니다.

키: HKEY_LOCAL_MACHINE/SOFTWARE/IBM/MQSeries/QueueManager/<QMNAME>

또는 MQSeries 서브시스템

채널 최적화 프로그램 메세지수집은 분산 플랫폼의 QM.INI 파일에서 찾을 수 있는 동일한 메세지수집을 z/OS에서 지원합니다. XPARMS는 CSQ6CHIP 매크로를 사용하여 발됩니다.

