



IBM 软件

保持通信并行运行

Morag Hughson 和 Paul Clarke

hughson@uk.ibm.com、paulg_clarke@uk.ibm.com

WebSphere. software

@business on demand software

(C) 2003 IBM Corporation

版本1.0.1

IBM 软件 WebSphere 软件



前言

本文档由IBM 根据IBM 对WebSphere MQ 和MQSeries 的规划以及使用这些产品及其环境的经验提供。是否采纳这些建议由读者自行决定。您可能会从其他来源听到不同观点。WebSphere MQ 中有几个区域它基本各异的,可能会根据WebSphere MQ 的定期改变。

作为读者阅读本文档之前会熟悉WebSphere MQ,您可以确定要删除或可保留的特定配置选项。

本文档系管理以及任何其他支持WebSphere MQ 网络的人员使用。希望它具有下列特征:

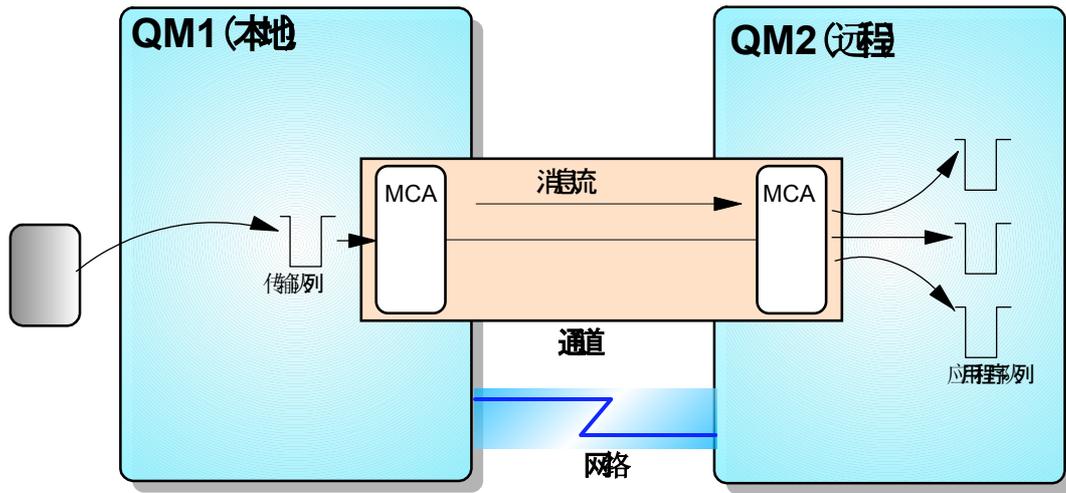
- 管理程序共致性
- 提供用野的最大可用性
- 在避免读者所犯错误方面提供帮助
- 辅助那些于WebSphere MQ 专家阶段的人员
- 通常情况保平稳并成功完成WebSphere MQ 项目

此提供的部分信息与MQSeries 和WebSphere MQ 的版本有关。请查看您的特定版本文档以确保受支持。

N
O
T
E
S

(C) 2003 IBM Corporation

通道结构



通道结构一说明

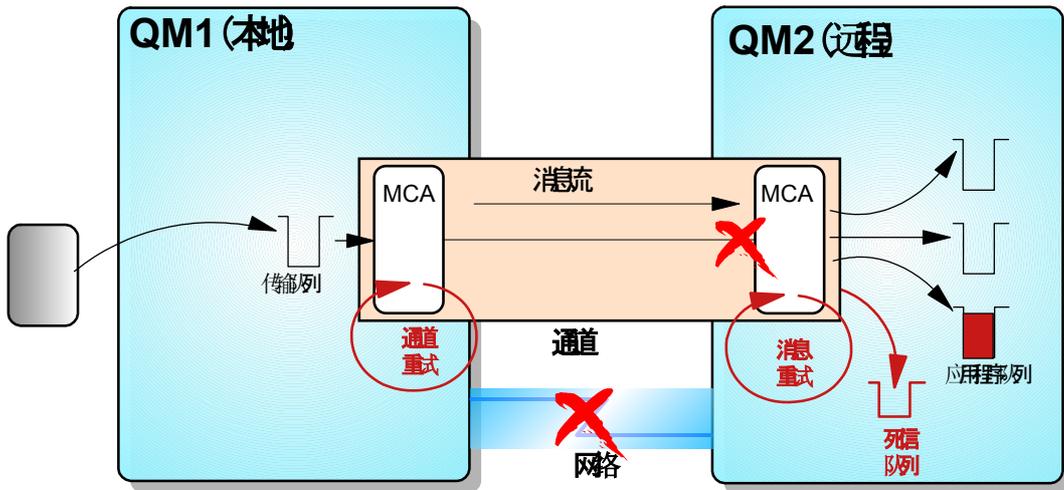
此图表示WebSphere MQ 通道结构 我在所有表中要用此表以说明不同的通道

图的主要部分显示程序可以把到达输入列的消息由程序定义或由地址传送到程序系统 在此图中QM1 是我的本地系统而QM2 是我的远程系统 为我的输入列发送MCA 使消息离开并且按定义指定的通信方式由接收MCA 接收并管理目的应用列 将它发送到程序系统 这叫MCA 称通道

这一切都很好但是会出什么错误?

N
O
T
E
S

会发生什么错误?



- 通道阻塞 - 远程QMgr 结束
- 消息不能传递 - 队列失效或已满
- 网络故障
- 性能

会发生什么错误? - 说明

此部分是我今天讨论的内容。我将要查看会发生错误的不同情况。然后，我将后部分子查看每一种情况更多的信息。

通道阻塞 例如，因为程序/管理限制。要从此类的失败中恢复，您可以使用通道管理器来重试这些通道。

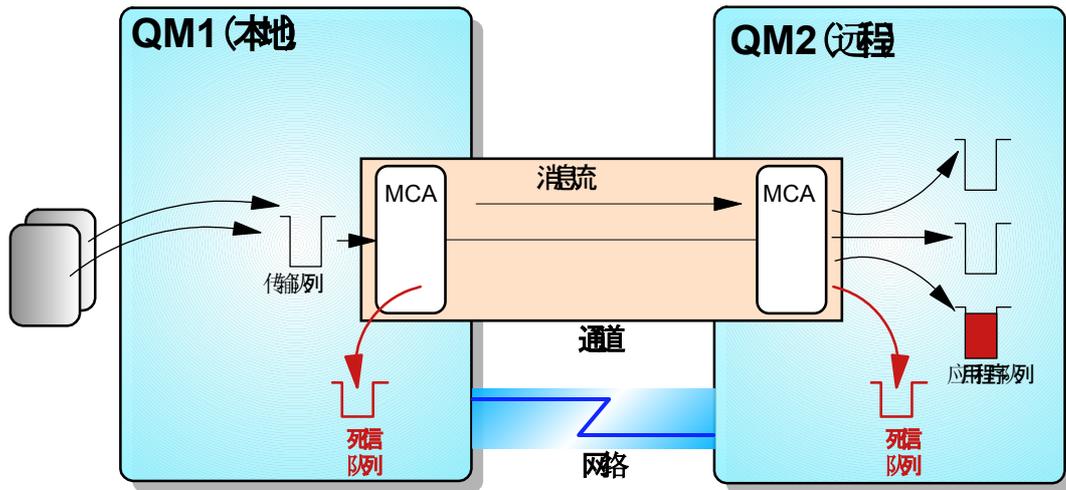
消息不能传递 这可能是因为应用程序错误，例如指定的队列无效，或因系统问题，例如队列已满。要在此类的问题后继续运行，您可以使用重试队列，或者您可以使用消息重试。

您可能会遭遇网络故障，而且根据故障的类型，您可通过不同方式对此故障恢复。

性能是个因素，因此我也考虑某些性能因素。

N
O
T
E
S

死队列



■ 如下例的消息存区域一

- ▶ 错误址
- ▶ 未数
- ▶ 太大

死队列—说明

强烈建议管理具有死队列。如果没有死队列，应用程序可能无法启动，编程错误足以使通道机，如果不同的应用程序使用同一通道，这当然能接受。

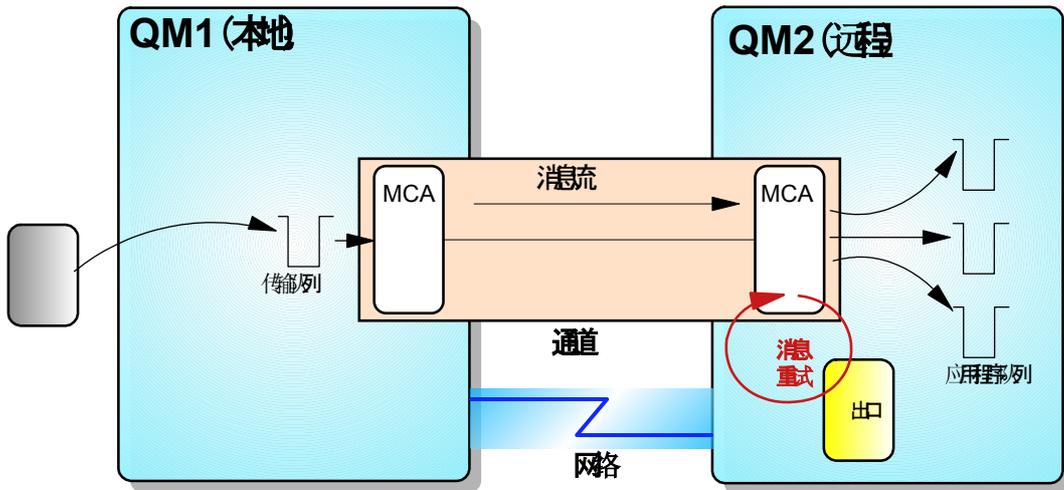
请注意可能在通道的任意一端用DLQ。在发送端，消息太大而要发送而超过通道机，那么将使用DLQ。如果能力超出了为通道定义的消息大小，那么传输队列内容，它也会进入这种情况。在接收端，如果无法将消息(PUT)到目标队列，那么将使用DLQ。众多原因有些可能模糊的，会进入这种情况。设置了消息参数为通道定义的，和消息出口来，理这些情况，我将该看能重式。

应该意DLQ的使用，在通道的任意一端，会影响消息的流动。如果以某些事物中成的，能存在一些问题，那么此消息将被DLQ，并且该消息将断。

某些情形，应用程序要求是消息，但我将其发送的，同时，这些应用程序要求DLQ不存在，没有DLQ，的消息是通道的，坏消息将阻止通道，阻止任何应用程序接收其消息。由此原因，强烈建议安装具有DLQ，并且多个应用程序以理解，接收此消息。此修改并非，并且不同的，可使用技术，例如，将识别用作，数器，具有，队列，或，用，能，分，组。

N
O
T
E
S

消息重试



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

消息重试一说明

请慎用消息重试特别是您在运行不具有队列时

有没有出口都可以使用消息重试 如果没有出口接收MCA

仅在每个时间间隔失败进行重试从通道失败重试之间的间隔重试次数 如果在有出口的情况下使用

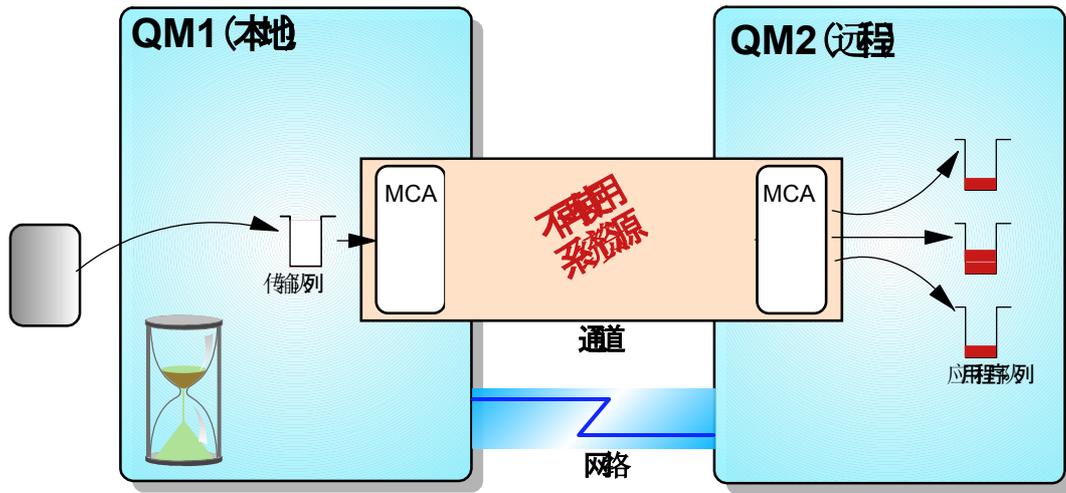
出口传递这些消息是可以将它们更改为任何值 使用消息重试机制的出口允许故障更替处理 例如如果不能将消息传送到目标队列您可以重试它们

对出口使用消息重试以消息为基础进行重试判定

当前在z/OS 上不支持消息重试

N
O
T
E
S

断开连接间隔



DEF CHL ... DISCINT(6000)

- 与接收



断开连接间隔—说明

断开连接间隔DISCINT 是发送型通道的属性，用于控制通道应保持多长时间（即使没有消息要送）。

您设置取决于以下条件：

- 您有多少通道及系统资源的重要度
- 您网络的负载
- 通道的负载

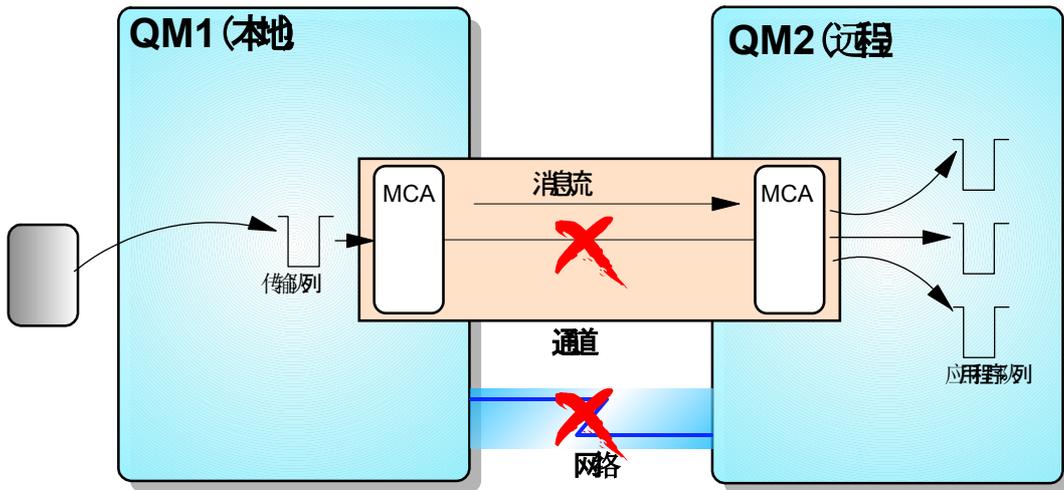
理想值应该小于接收方的平均时间，这意味着没有工作接收方的通道断开连接。

应将您通道的断开连接间隔值与通道接收起来一起设置为通道。送当该通道有等待工作，它自启动，并且它不再等待工作，将自结束。

N
O
T
E
S



网络故障



- 脉动器
- TCP/IP Keepalive
- 采用MCA

网络故障一说明

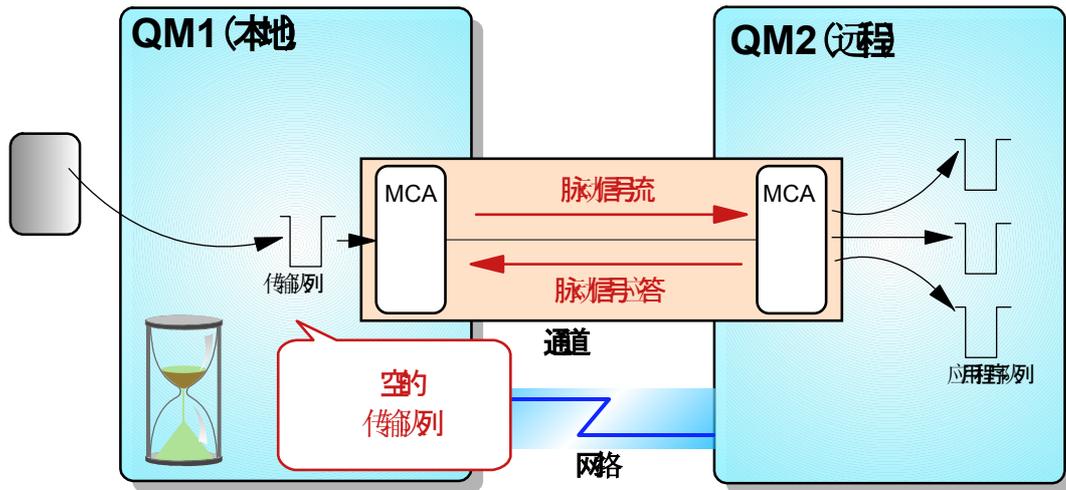
报告通道的任何故障引起通道关闭。如果网络故障被报告，通道将关闭。通道不能知道故障。

对于任何网络，中断通道的低级通信是不可靠的。例如，交换机通道的停止与recv调用有关。如果recv调用返回，则通道将尝试重新连接。MCA的困难之一是网络已经失效，但发现是由于comms协议返回错误。通常，此LU6.2比TCP/IP（它是更好的协议）容易得多。

对MCA的修改已经很大程度上改善了状况。我将通过接下来的几部分查看其中的部分。

N
O
T
E
S

脉搏号



DEF CHL HBINT(300)

空缓冲区 ✓

关闭队列 ✓

脉搏号—说明

在没有发送脉搏号或HBINT 是空通道的发送多久检查器是否已经的通道性 该在通道的端指定且它最合适以通道上使用 此是指的个值最频繁的间隔 如果一端端指定了, 这意味着没有脉搏号那么此值作为最频繁的可能是个选择

缺省情况下通道的脉搏号定义为300 秒5 分钟。选择相对的值缺省值因此网络只有微小影响 实际值的值比方说60 秒在多数情况下

此脉搏号是发送接收MCA (或MQGET 期的CLNTCONN)

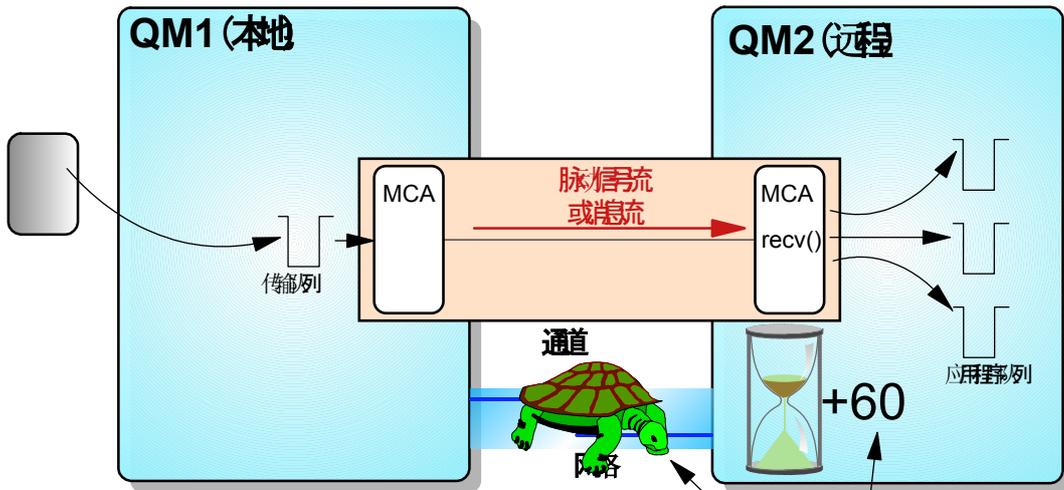
以它作为可用的信息, 仅当脉搏号在通道上没有活动时发送 这能解决两个问题发送MCA 现在可以即插入网络返回而是向其MQGET; 并且它也使接收器MCA 执行一些逻辑以管理及影响之检查是否止

另外因为通道不停地发送因此通道并不保存其数据和队列高速缓存 因此在脉搏号期时通道也不需任何缓存缓冲区消息区域并任何高速缓存的列

因此为了能保持脉搏号不过大, 但是请注意, 脉搏号开始起没有是通道的被网络流量保持 请考虑将通道向发送通道的用途, 即使是3 秒的脉搏号也不会超过脉搏号因此是有一个好的需要送

N
O
T
E
S

接收等待时间



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

反网络等待时间



接收等待时间一说明

我已提及通道的传递周期性发送脉冲信号。脉冲信号是一个商值，因网络延迟是长。如果没有脉冲信号因为延迟是随机的。因此通道的接收端会等待一部分时间到达通道。然而有了脉冲信号接收端会知道在脉冲信号间隔是总是脉冲信号到达。如果这样，那么通道会一直等待。这用作TCP/IP 网络故障的基础。对于CLNTCONN 通道等待到MQI 命令的响应用也是。对于SVRCONN 在不支持脉冲信号容器的z/OS 上除外。

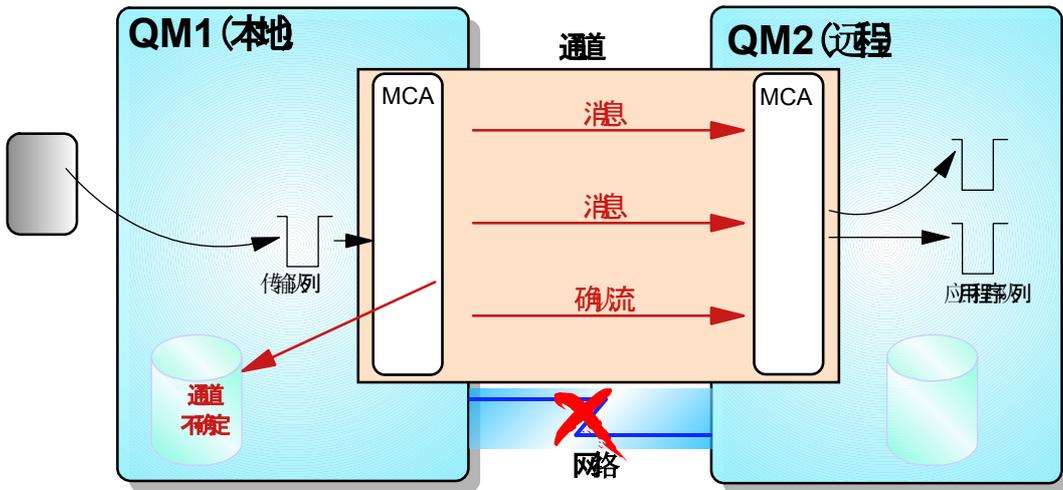
接收器实际在以下情况超时：如果商脉冲信号间隔于60 秒在网络脉冲信号间隔内接收到数据，或者大于或等于60 秒超过了商脉冲信号间隔60 秒。缺省情况下假定存在通道故障。在分布式上通过更改环境变量MQRCVBLKTO。

实际上可以调整限制以反映网络等待时间。当网络脉冲信号间隔的时候也有另一个环境变量MQRCVBLKMIN 允许您看最小使用值。通过用XPARM 属性RCVTIME 限制超时以及RCVTMIN 来设置。值可以在V5.3.1 的z/OS 上置为。

因此，如果您的通道负载小值更快地检测到故障，但是也会带来更多的网络流量和CPU 资源。

N
O
T
E
S

确传递



- 在两端看到相同数据
 - 网络故障后重新同步
 - 群集算法重新分配不确定处理消息

确传递一说明

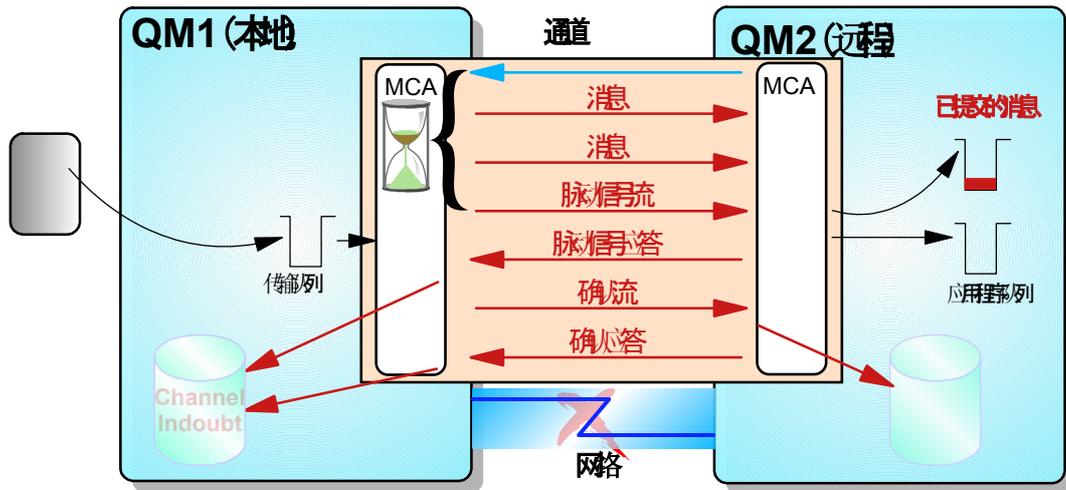
在我们可以看送处理消息的端端。注意在端端中MCA的消息和数据是磁的。任何从的端端进行操作。所有数据包含在送消息。消息的标识一旦送方对其合作发出确认流在接收端之前是不确定。换句话说发方通道能确定是否成功传递。如果在此段在通道端那通道等来不确定。当它重新接其合作时。将数据存其合作的不确定。因将向其通道是最后一批消息已传递。使用合作方的答通道可以决定是否交回其传输列的消息。

通过发出DIS CHSTATUS(*) SAVED 命令可以查看同端数据。通端端的示应该相同。

请注意如果当通道不。它重新启动它将解决不确定。然而它在同一合作时可以进行此操作。如果通道生更改或不同的管理器接管了IP 地址或服务器同一传输列的通道。那通道立即结束并有信息示不同的管理。当然是不确定。用。必须将其向正确的管理。的通道。或发出RESOLVE CHANNEL 命令解决不确定。请注意在此情况用。应该来自DIS CHS(*) SAVED 的出来。解进行正确操作。COMMIT 或BACKOUT。

N
O
T
E
S

批处理消息



DEF CHL BATCHHB(500)

- 如果自最后一次通信的时间大于时间间隔则发出脉谱消息

反网络稳定性



批处理消息—说明

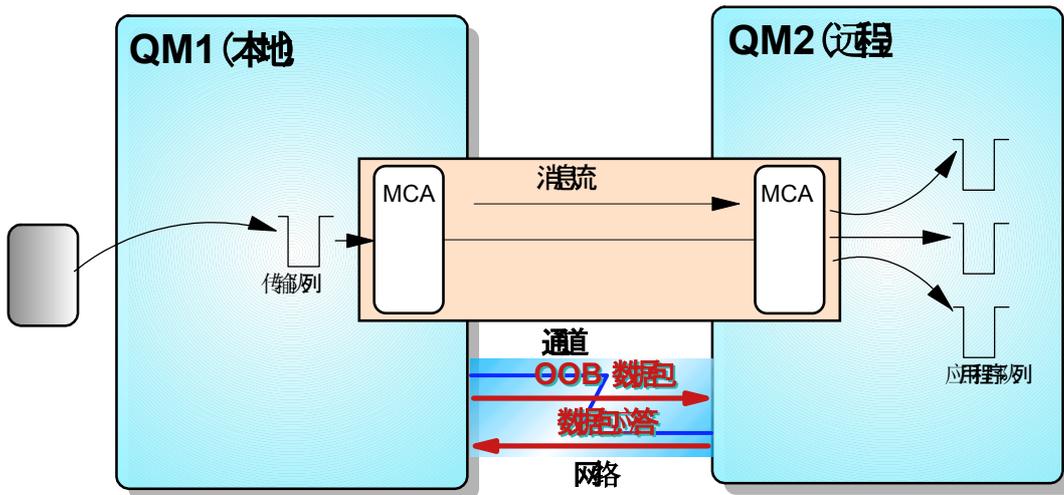
结束不确定性的最佳算法是看，如果集通道正常，则由通道发送的任何数据重新分配另一个通道且清除队列中的另一个实例。然而，如果它是发送通道，则清除通道是否已经接收的批处理消息部分将无法重新分配它们。

要确认通道在不确定中结束的可能性已被清除，则通道以，此流对批处理消息，在发送批处理消息后，只在通道本身标记为不确定时发送流。由于已从通道的接收端接收，我可以完成我能完成批处理。现在有一个开放的窗口，其批处理消息可以结束，但更可能停留在不确定状态。如果没有批处理消息，此窗口会关闭，从队列接收一个流，直到现在为止。

要确定批处理消息是否已发送，我要上一次我的队列接收的时间。如果比时间通道的BatchHeartbeat 或BATCHHB 参数中的时间间隔早，那我将发送流，否则不送，因此指定的时间间隔是网络稳定性，因此时间间隔是确定。

N
O
T
E
S

TCP/IP Keepalive



- QM.INI **KEEPALIVE=YES**
- XPARM **TCPKEEP=YES**
DEF CHL ... KAIN(T360) (仅z/OS)

TCP/IP Keepalive 一说明

TCP/IP 也有自己的术语称为KeepAlive, 允其网络做健康用KeepAlive。

仅在z/OS上可以配置不同通道保持或禁用系统的通道参数。此通道参数需要KeepAlive 在可以切换到TCP/IP。

在有些上可以在WebSphere MQ 的列管理基础上在qm.ini 文件中配置。在此情况下, 器和数使用的值是机器的整个TCP/IP 堆栈的。典型是在每2 小时后网络做健康。这能WebSphere MQ 环境做大了, 但同一堆栈的其它程序可能要用KeepAlive, 因此可能提些不同用要求的表。

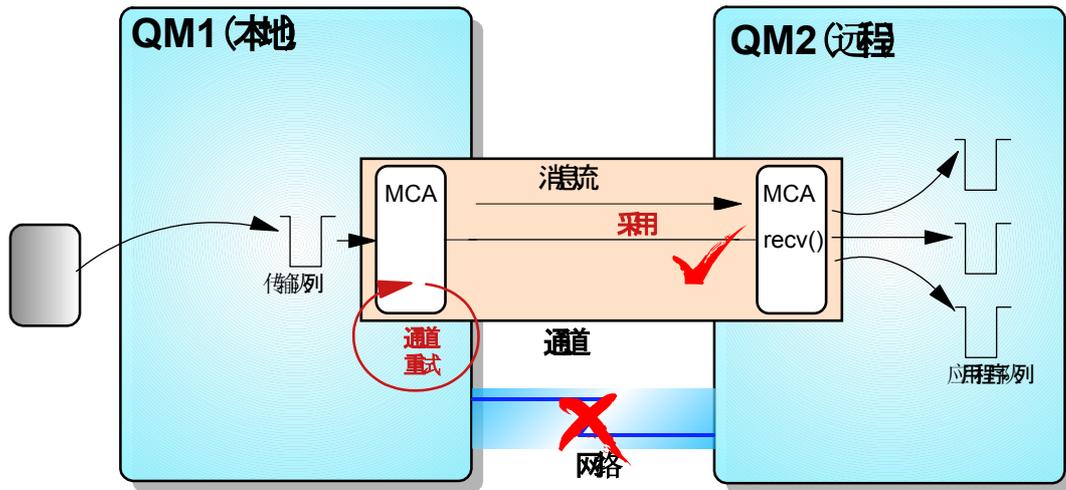
您需要调参的TCP/IP 手册找到何置KeepAlive。对于AIX, 请用no"命令。对于HP-UX, 请用netttune"命令。对于Windows NT, 您需要调参表。在z/OS上请参您的TCP/IP PROFILE 数据集并更改TCPCONFIG 部分的INTERVAL 参数。注建在z/OS上请用TCP/IP APAR PQ75195。

当网络不选时, 脉冲看使再健康等超。除了对KeepAlive 的, 然而KeepAlive 也然可以用。当不支持看和分表的WebSphere MQ 实例通时您应勿然使用KeepAlive。

因则客户端看不用仍对SVRCONN 通道健康用keepalive, 因此仅在MQGET 期使用。

NOTES

采用MCA



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

采用MCA 一淘

如果通道(例如通信, 不保和通端同时检测到故障, 经常是通道的远端先检测到故障, 因为它是将消息发送给远端, 因此要发出的是远端重新连接目标管理器, 当远端管理器已有多名来自本通道的运行因新的连接拒绝

N
O
T
E
S

通过采用AdoptNewMCA, 您可以告诉管理器如果新连接并且已经存在自网络地址来自本管理器具有名称的通道在, 那么有通道该名称由此新连接替换, 即说通道列应该由新连接采用。

QM.INI 的channels 节中的各参数或情况或在z/OS 上使用(XPARMS)允许可行稍更改, 您可以在方式采用通道列表时采用

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

ADOPTMCA=YES/NO (在z/OS 上)

设置合也来采用通道立即来它之的等待时间隔

AdoptNewMCATimeout=60

显然通道必须配

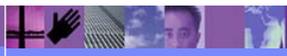
但是您可以指定网地址及程序管理器名称也必须, 阻止了来自网上不同机器的任意采用已运行通道, 必须采用新连接之的属性置/配

AdoptNewMCACheck=QM,ADDRESS,NAME,ALL

ADOPTCHK=NONE/QMNAME/NETADDR/ALL (在z/OS 上)

性能

- **考虑某些问题**
 - ▶ 什么是输入队列到达速度?
 - ▶ 什么是需要的响应时间?
 - ▶ 批处理多长?
 - ▶ 通道的吞吐量是什么?
- **回答取决于您的环境**
 - ▶ 应用程序使用情况
 - ▶ 操作系统硬件
 - ▶ 网络带宽
 - ▶ 消息大小

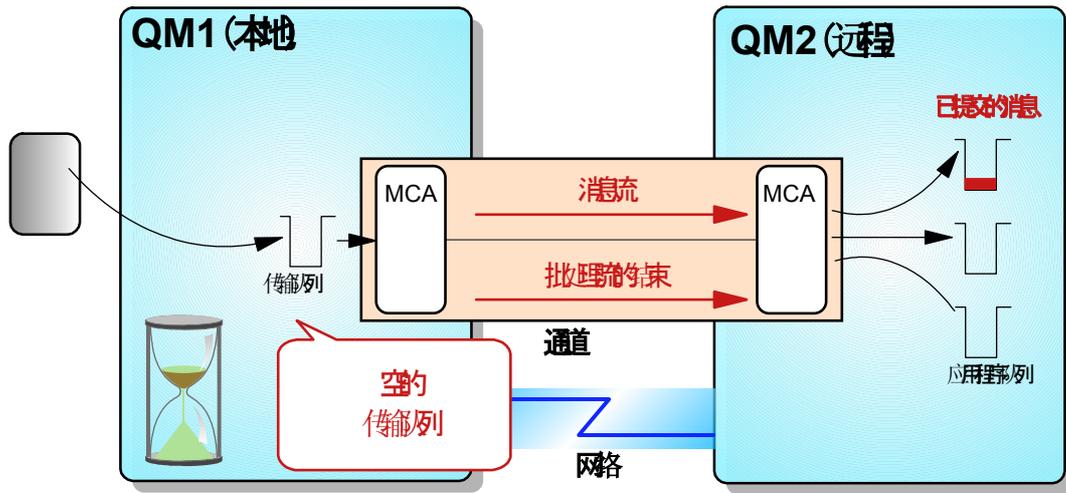


性能设置

- **使用BatchInterval 保持批处理开**
 - ▶ 如果到达速度很慢才发送完成的批处理
 - ▶ 可能增加等待时间但是提高了总的吞吐量
- **将DisconnectInterval 设置为足够高以保持通道保持活动**
 - ▶ 减少通道自动停止重新连接的次数
 - ▶ 防止网络由活动会话主机堵塞正在达到MaxChannels 限制
- **三种方案**
- **使用绑定**
- **请参阅Capacity Planning Guides**
 - ▶ SupportPac MP00
 - ▶ SupportPac MP16



批处理间隔



■ 批处理消息的生存期

DEF CHL BATCHINT(100)

▶ 可复消息会比BATCHINT 早达



批处理间隔一说明

N
O
T
E
S

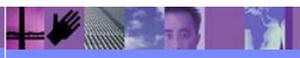
一旦传输列空可复的批处理消息约束, 这会导致需要更多传输列和更多活动。请考虑3 条消息到传输列的应用。这是通过第 3 条消息从第 1 条消息的机会, 因为通道完成批处理第 1 和第 2 条消息, 然后将第 2 批处理

消息的延迟到批处理比方说BATCHINT(100), 那所有 3 条消息发送第一个批处理并阻塞的延迟为半。请这尽管存在至少 100ms 的延迟, 但是可能显示出来较长时间来传递, 因为非持续性消息NPMSPEED(FAST) 是在向点的因此BATCHINT 对这些不起作用。通常, 应用将紧急数据查询作为非持续性消息发送, 并将不那么紧急数据再作为持续性消息发送。

BATCHINT 的效果值主要取决于MQ 消息传输网络和消息键文件。根据的通道越多潜在省越多。

一种帮助能BATCHINT 更出好的方法是看有效的批处理大小。发出DISPLAY CHSTATUS, 并将该批处理删除。如果批处理大小在 1 和 2 之间, 那么通道必须每 1 个或 2 个消息一次发送效率低且它受益于BATCHINT。这只是一指示, 因为当所有消息非持续性时 BATCHINT 无任何帮助。

要看另一件事是通道传输使用情况。如果您能可释放那么这是一个机会BATCHINT 将减少成本并提高吞吐量。



方案1

■ 成批传送的传输列

- ▶ 使用的BatchSize (缺省为50 是较好的,
 - 除使用异常的消息重新传输成本
- ▶ BatchInt 应该为0



方案1 一测

我的第一个调整是成批传送的传输列的消息

您的目标是批量传送从而减少每条消息处理成本, 所以当使用的批理大小(缺省为50 是较好的), 而批理间隔应当置为0。在这个方案中保持批理打开所有消息都是装入传输列的

您的省悟是在一次批理发送多少条消息以取每条消息处理成本, 以及用重新传输批理去网络通信会免部分通信成本, 如果您转移异常的消息那么此研究可能要用比50 更大的批理大小来完成

N
O
T
E
S



方案2

■ 为延迟的使用流式传送

- ▶ 取决于到达速率峰值
- ▶ 尝试更大的批理
- ▶ 设置BatchInt 为输出可接受延迟例如1分钟
- ▶ 但不要设置得太大否则会导致运行的工作单元



(C) 2003 IBM Corporation



方案2 一说明

我的第二个调整方案是延迟的使用流式消息传送

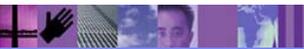
您的目标是管理消息的批理大小

从而减少批理条数带来的成本 这是批理间隔的常用消息以通常处理慢速到达 保持批理间隔稍小以便更多消息一起批理

您的首要是在一次批理中发送多少消息根据具体情况 可变的延迟可能会需要时间来传递因为在批理完成之后即至少在批理间隔) 这些消息才会被送至目标队列

如果随时间来出小心不要把批理间隔设置得太小 因为这会生成过多的工作单元的

N
O
T
E
S



(C) 2003 IBM Corporation

方案3

■ 同步请求

- ▶ 当响应时间是关键时可能不想用BatchInt
- ▶ 如果吞吐量考虑较小的BatchSize
- ▶ 对于非实时流使用NPMSPEED(FAST) 和零BatchInt



方案3 一说明

我的第三个性能调整方案是同步请求类型，其中响应时间是关键

这类似于方案2，除了这次我们偏向于快速的响应时间而不是批处理中更多的消息。

如果您的吞吐量可能考虑使用较小的批处理大小且不使用任何批处理间隔或在几秒钟内则使用相当的值以确保在适当方式可用

如果您有非实时流则当使用非实时速度或NPMSPEED 为FAST, 并设置零批处理间隔
因为快速通过传输的非实时消息要用的 仍使用批处理大小值但不影响响应时间。

要出些理解

使用非实时速度30 非实时流秒设置BatchInt=2000(2 秒)和BatchSize=50 将50 作为有效的批处理大小。

N
O
T
E
S



委托

- 单个进程
 - ▶ 用MCA 和本地代理器程序
 - ▶ 无IPC 供MCA MQI 命令稍
 - 对性能有意义
- 在代理器ini 文件配置
 - ▶ Channels 节

MQIBINDTYPE=FASTPATH

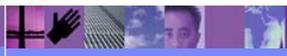
- 如果测试通道出口则不要使用
- 发出小心

STOP CHL() MODE(FORCE)

(V5.3 之前)

STOP CHL() MODE(TERMINATE)

V5.3



委托—说明

缺省情况下在产品的分布式集合所有代理器操作是独立称为代理器的程序的每个API 调用是由一些PC 机的代理器发送的。这是由完成原因 应用程序不直接为代理器内存。然而因为每个MQI 调用是一个进程对它产生影响。对性能总大约 60% 的消耗是在代理器中。

**N
O
T
E
S**

所有应用程序都运行在委托方式中。这意味着代理器与应用程序内运行。我建议一般应用程序不要运行在此方式。然而通道产品代码 因此合作快速委托应用程序运行。缺省情况下它们以常规方式运行但通过QM.INI 文件的Channels 节中置 MQIBindType=FASTPATH,所有通道委托运行。

应当意通道出口。因为通道出口与MCA 内运行所出口。现在将与代理器代理器同时内存中。在此方式运行之前应当测试通道出口。

还应当使用V5.3 之前注册的STOP MODE(FORCE) 命令在V5.3 上应当使用STOP MODE(TERMINATE) 命令。这两个命令可能因通道被死清除完全执行而没有被代理器。



参 考 用

通道	描述
HBINT	脉冲间隔
BATCHSZ	批处理最大消息数
BATCHINT	消息批处理最大周期
DISCINT	通道关闭的间隔
SHORTRTY、SHORTTMR LONGRTY、LONGTMR	通道关闭和重启
MRRTY、MRTMR MREXIT、MRDATA	消息数和重启出口不在 z/OS 上

QM.INI -Channel 节	XPARMS	描述
AdoptNewMCA	ADOPTMCA	采用通道型
AdoptNewMCATimeout		采用等待超时
AdoptNewMCACheck	ADOPTCHK	采用参数检查
KeepAlive	TCPKEEP	使用Keepalive
MQIBindType		通道FASTPATH 绑定

环境变量	XPARMS	描述
MQRCVBLKTO	RCVTIME	接收等待超时
MQRCVBLKMIN	RCVTMIN	接收等待最短超时



参 考 用 一 说 明

此表列出那些影响性能和能力的通道属性

一些属性是通道的部分而另一些是其他地方例如 INI 文件的设置

有通道的属性完整信息可查阅 InterCommunication 一书

在分布式 QM.INI 文件中的那些参数都在 Channels 节中或在 Windows 上它都在注册表中

键 HKEY_LOCAL_MACHINE\SOFTWARE\IBM\MQSeries\QueueManager/<QMNAME>

或用 MQSeries 服务

通道程序参数在 z/OS 上提供一些相同的参数而在分布式的 QM.INI 文件中找到 XPARMS 是用 CSQ6CHIP 宏键的

N
O
T
E
S

