

IBM MQ培训

基本概念，常用命令，日常运维

IBM MQ产品

是**IBM**的一款商业消息中间产品，适用于分布式计算环境或异构系统之中。消息队列技术是分布式应用间交换信息的一种技术。消息队列可驻留在内存或磁盘上,队列存储消息直到它们被应用程序读走。通过消息队列，应用程序可独立地执行--它们不需要知道彼此的位置、或在继续执行前不需要等待接收程序接收此消息。

MQ基本概念

队列管理器

队列管理器是MQ系统中最上层的一个概念，由它为我们提供基于队列的消息服务。

MQ基本概念

消息

在MQ中，我们把应用程序交由MQ传输的数据定义为消息，我们可以定义消息的内容并对消息进行广义的理解，比如：用户的各种类型的数据文件，某个应用向其它应用发出的处理请求等都可以作为消息。消息有两部分组成：消息描述符(Message Description或Message Header)，描述消息的特征，如：消息的优先级、生命周期、消息Id等；消息体(Message Body)，即用户数据部分。在MQ中，消息分为两种类型，非永久性(non-persistent)消息和永久性(persistent)消息，非永久性消息是存储在内存中的，它是为了提高性能而设计的，当系统掉电或MQ队列管理器重新启动时，将不可恢复。当用户对消息的可靠性要求不高，而侧重系统的性能表现时，可以采用该种类型的消息，如：当发布股票信息时，由于股票信息是不断更新的，我们可能每若干秒就会发布一次，新的消息会不断覆盖旧的消息。永久性消息是存储在硬盘上，并且纪录数据日志的，它具有高可靠性，在网络和系统发生故障等情况下都能确保消息不丢、不重。此外，在MQ中，还有逻辑消息和物理消息的概念。利用逻辑消息和物理消息，我们可以将大消息进行分段处理，也可以将若干个本身完整的消息在应用逻辑上归为一组进行处理

MQ基本概念

队列

队列是消息的安全存放地，队列存储消息直到它被应用程序处理。

消息队列以下述方式工作：

- a) 程序**A**形成对消息队列系统的调用，此调用告知消息队列系统，消息准备好了投向程序**B**；
- b) 消息队列系统发送此消息到程序**B**驻留处的系统，并将它放到程序**B**的队列中；
- c) 适当时间后，程序**B**从它的队列中读此消息，并处理此信息。

由于采用了先进的程序设计思想以及内部工作机制，MQ能够在各种网络条件下保证消息的可靠传递，可以克服网络线路质量差或不稳定的现状，在传输过程中，如果通信线路出现故障或远端的主机发生故障，本地的应用程序都不会受到影响，可以继续发送数据，而无需等待网络故障恢复或远端主机正常后再重新运行。

MQ基本概念

在MQ中，队列分为很多种类型，其中包括：本地队列、远程队列、模板队列、动态队列、别名队列等。本地队列又分为普通本地队列和传输队列，普通本地队列是应用程序通过API对其进行读写操作的队列；传输队列可以理解为存储-转发队列，比如：我们将某个消息交给MQ系统发送到远程主机，而此时网络发生故障，MQ将把消息放在传输队列中暂存，当网络恢复时，再发往远端目的地。远程队列是目的队列在本地的定义，它类似一个地址指针，指向远程主机上的某个目的队列，它仅仅是个定义，不真正占用磁盘存储空间。模板队列和动态队列是MQ的一个特色，它的一个典型用途是用作系统的可扩展性考虑。我们可以创建一个模板队列，当今后需要新增队列时，每打开一个模板队列，MQ便会自动生成一个动态队列，我们还可以指定该动态队列为临时队列或者是永久队列，若为临时队列我们可以在关闭它的同时将它删除，相反，若为永久队列，我们可以将它永久保留，为我所用。

MQ基本概念

通道

通道是MQ系统中队列管理器之间传递消息的管道，它是建立在物理的网络连接之上的一个逻辑概念，也是MQ产品的精华。在MQ中，主要有三大类通道类型，即消息通道，MQI通道和Cluster通道。消息通道是用于在MQ的服务器和服务器之间传输消息的，需要强调指出的是，该通道是单向的，它又有发送(sender)，接收(receive)，请求者(requestor)，服务者(server)等不同类型，供用户在不同情况下使用。MQI通道是MQ Client和MQ Server之间通讯和传输消息用的，与消息通道不同，它的传输是双向的。群集(Cluster)通道是位于同一个MQ群集内部的队列管理器之间通讯使用的。

MQ工作原理

首先来看本地通讯的情况，应用程序A和应用程序B运行于同一系统A，它们之间可以借助消息队列技术进行彼此的通讯：应用程序A向队列1发送一条信息，而当应用程序B需要时就可以得到该信息。

其次是远程通讯的情况，如果信息传输的目标改为在系统B上的应用程序C，这种变化不会对应用程序A产生影响，应用程序A向队列2发送一条信息，系统A的MQ发现Q2所指向的目的队列实际上位于系统B，它将信息放到本地的一个特殊队列—传输队列(Transmission Queue)。我们建立一条从系统A到系统B的消息通道，消息通道代理将从传输队列中读取消息，并传递这条信息到系统B，然后等待确认。只有MQ接到系统B成功收到信息的确认之后，它才从传输队列中真正将该信息删除。如果通讯线路不通，或系统B不在运行，信息会留在传输队列中，直到被成功地传送到目的地。这是MQ最基本而最重要的技术--确保信息传输，并且是一次且仅一次(once-and-only-once)的传递。

MQ提供了用于应用集成的松耦合的连接方法，因为共享信息的应用不需要知道彼此物理位置（网络地址）；不需要知道彼此间怎样建立通信；不需要同时处于运行状态；不需要在同样的操作系统或网络环境下运行。

MQ基本操作

创建队列管理器

`crtmqm -q QMgrName`

-q是指创建缺省的队列管理器

删除队列管理器

`dltmqm QmgrName`

启动队列管理器

`strmqm QmgrName`

如果是启动默认的队列管理器，可以不带其名字

停止队列管理器

`endmqm QmgrName` 受控停止

`endmqm -i QmgrName` 立即停止

`endmqm -p QmgrName` 强制停止

显示队列管理器

`dspmq -m QmgrName`

MQ基本操作

运行MQSeries命令

runmqsc QmgrName

如果是默认队列管理器，可以不带其名字
往队列中放消息

amqsput QName QmgrName

如果队列是默认队列管理器中的队列，可以不带其队列管理器的名字
从队列中取出消息

amqsget QName QmgrName

如果队列是默认队列管理器中的队列，可以不带其队列管理器的名字
启动通道

runmqchl -c ChlName -m QmgrName

启动侦听

runmqslsr -t TYPE -p PORT -m QMgrName

停止侦听

endmqslsr -m QmgrName

MQ基本操作

定义死信队列

```
DEFINE QLOCAL (QNAME) DEFPSIST (YES) REPLACE
```

设定队列管理器的死信队列

```
ALTER QMGR DEADQ (QNAME)
```

定义本地队列

```
DEFINE QL (QNAME) REPLACE
```

定义别名队列

```
DEFINE QALIAS(QALIASNAME) TARGQ(QNAME)
```

远程队列定义

```
DEFINE QREMOTE (QRNAME) +  
RNAME (AAA) RQMNAME (QMGRNAME) +  
XMITQ (QTNAME)
```

定义模型队列

```
DEFINE QMODEL (QNAME) DEFTYPE (TEMPDYN)
```

定义本地传输队列

```
DEFINE QLOCAL(QTNAME) USAGE(XMITQ) DEFPSIST(YES) +  
INITQ (SYSTEM.CHANNEL.INITQ) +  
PROCESS(PROCESSNAME) REPLACE
```

MQ基本操作

创建发送方通道

```
DEFINE CHANNEL (SDRNAME) CHLTYPE (SDR) +  
CONNAME ( '100.100.100.215(1418)' ) XMITQ (QTNAME) REPLACE
```

其中CHLTYPE可以是：SDR、SVR、RCVR、RQSTR、CLNTCONN、SVRCONN、CLUSSDR和CLUSRCVR。

创建接收方通道

```
DEFINE CHANNEL (SDR_TEST) CHLTYPE (RCVR) REPLACE
```

创建服务器连接通道

```
DEFINE CHANNEL (SVRCONNNAME) CHLTYPE (SVRCONN) REPLACE
```

更改属性

```
ALTER QMGR DESCR ( 'NEW DESCRIPTION' )
```

```
ALTER QLOCAL (QNAME) PUT (DISABLED)
```

```
ALTER QALIAS (QNAME) TARGQ (TARGQNAME)
```

删除队列

```
DELETE QLOCAL (QNAME)
```

```
DELETE QREMOTE (QRNAME)
```

MQ编程

JAVA编写代码访问MQ，传递数据

C编写代码访问MQ，传递数据

MQ日常运维

熟悉MQ基本操作命令

查看MQ队列

维护通道

查看监听

实际登录操作

```
welcome to aliyun Elastic Compute Service!  
  
[root@SCM_MQ_TST_01 ~]# su - mqm  
-bash-4.1$ id  
uid=498(mqm) gid=500(mqm) groups=500(mqm)  
-bash-4.1$ dspmqver  
Name:          WebSphere MQ  
Version:       7.0.1.3  
CMVC level:    p701-103-100813  
BuildType:     IKAP - (Production)  
-bash-4.1$ dspmq  
QMNAME(SCM_QMGR)                                STATUS(Running)  
-bash-4.1$
```

- **SCM_MQ_PRD_01**

ssh: 120.76.198.46:38023 10.3.11.3 root/P2nf@ai3rd#

- **SCM_MQ_TST_01**

ssh: 120.76.198.46:38024 10.3.11.4 root/P2nf@ai3rd#

MQ图形化操作

不需记繁复的MQ命令
图形化操作。



Thank You !