# Activemq的安装

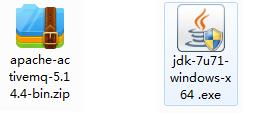
## 简介

Apache ActiveMQ是Apache出品的最流行的，能力强劲的开源消息总线。它支持多种语言和协议编写客户端。

本文档以Active MQ 5.14.4版本为例来进行截图。

Active MQ 5.15.0把jdk的最小版本设置为了JDK8

ActiveMQ需要运行在JRE下，因此安装active mq之前需要先安装jdk或jre。建议选用jdk1.7的相关版本。本次安装截图以jdk7u71的64位版本为例。



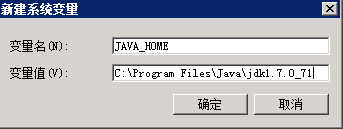
## JDK安装

### 下载jdk

下载jdk时，第一要选择jdk1.7版本；第二要根据安装的机器是64位还是32位来选择。

### 设置环境变量

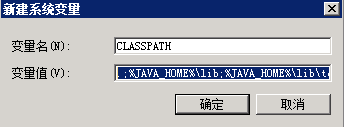
1. 新建系统变量JAVA\_HOME，设置为JDK的安装路径



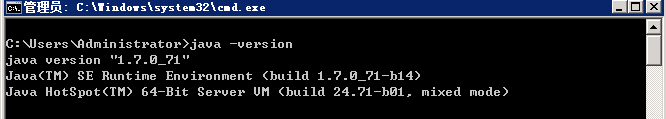
1. 在Path变量后添加;%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin



1. 新建CLASSPATH系统变量，设置为.;%JAVA\_HOME%\lib;%JAVA\_HOME%\lib\tools.jar;%JAVA\_HOME%\lib\dt.jar



1. 在cmd中输入java或者javac进行测试，至此jdk安装完毕



## ActiveMQ安装测试

### 下载ActiveMQ

去官方网站下载：[http://activemq.apache.org/](http://activemq.apache.org/" \t "_blank)

### 配置ActiveMQ

#### 禁用死信队列

可以配置过期消息不发送到DLQ。默认是发送的。

<!-- Set the following policy on all queues using the '>' wildcard -->

<policyEntry queue=">">

<!--

Tell the dead letter strategy not to process expired messages

so that they will just be discarded instead of being sent to

the DLQ

-->

<deadLetterStrategy>

<sharedDeadLetterStrategy processExpired="false" />

</deadLetterStrategy>

</policyEntry>

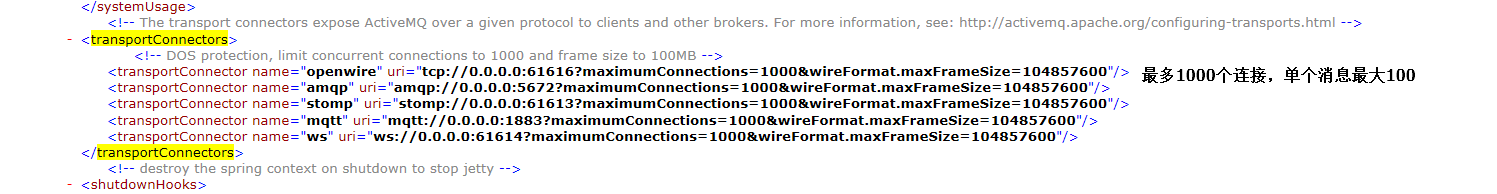


#### 修改端口和单个消息最大值和最大连接数

默认端口61616，默认单个消息最大100M，默认最大同时连接1000

将默认最大连接数1000修改为2048

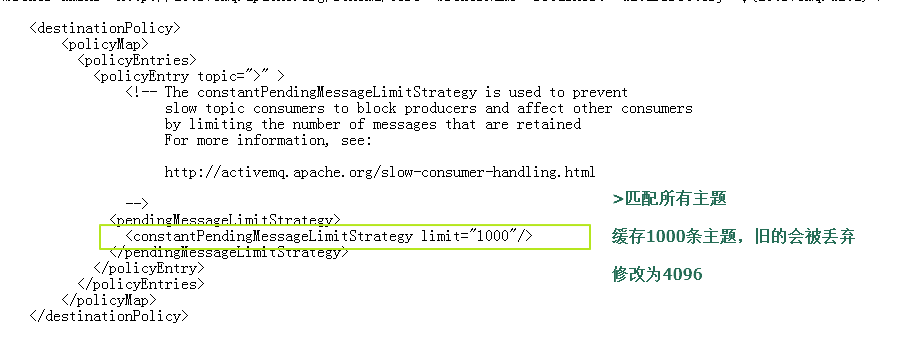
Conf/activemq.xml配置文件



配置项更改后，必须重启才能生效。

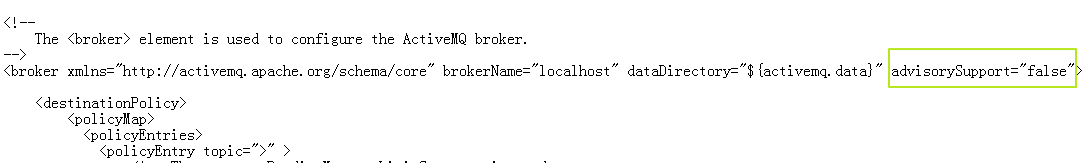
修改openwire支持的最大连接数和单个消息的最大字节数。activeMQ默认使用OpenWire协议交换消息。

#### 修改主题缓存条数



#### 禁用Advisory Message

<broker advisorySupport="false">



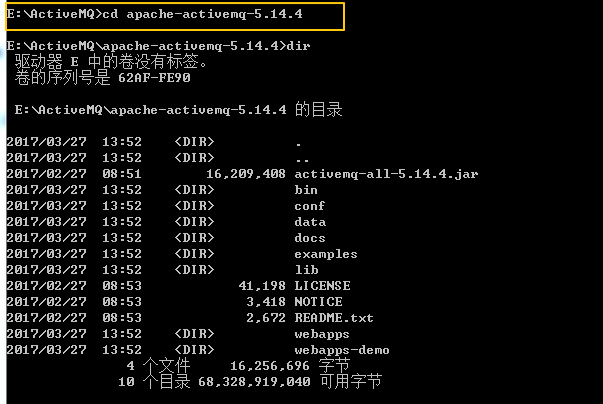
Advisory Message是非持久的，不过期的

### 运行ActiveMQ

解压缩apache-activemq-5.14.4-bin.zip到文件夹apache-activemq-5.14.4-bin下

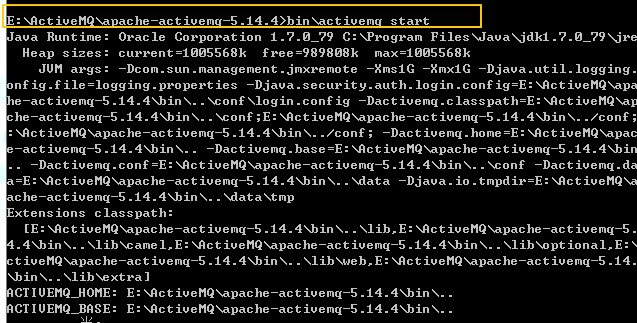
#### 启动ActiveMQ

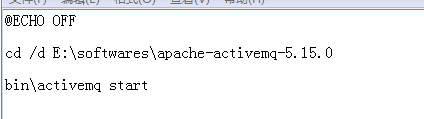
1. 启动cmd命令，进入activeMQ安装路径



1. 执行如下命令，启动ActiveMQ 或者编写一个bat文件来启动

bin\activemq start



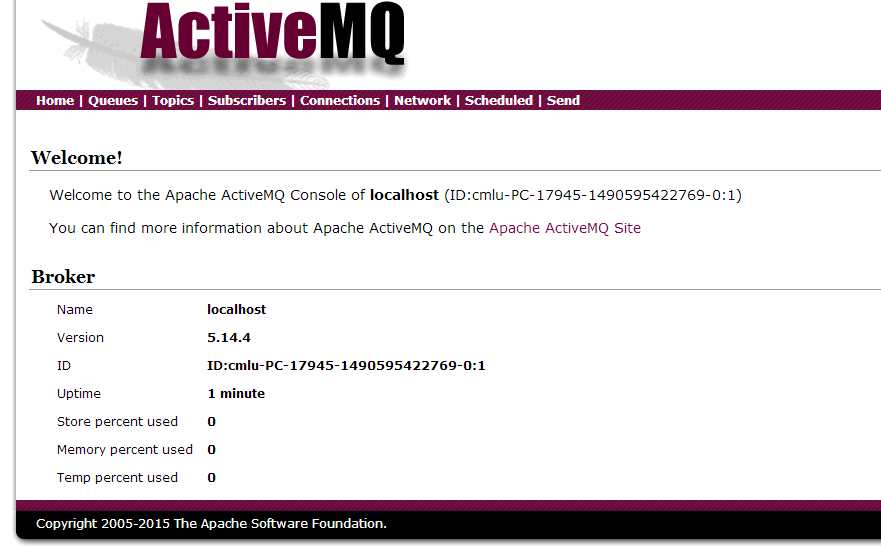


cd命令是不会修改当前的drive的。必须加/d参数

3、测试安装是否成功

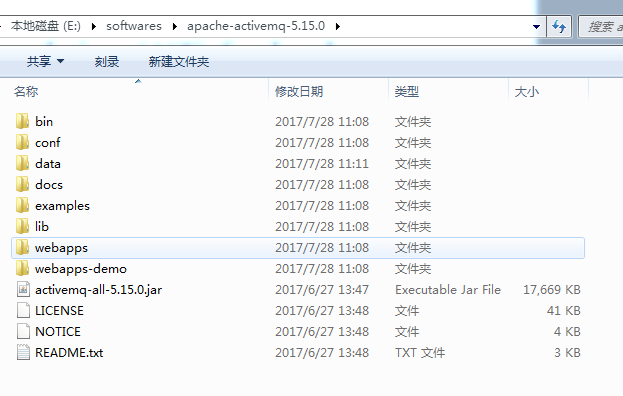
启动ActiveMQ程序后，登陆： <http://localhost:8161/admin/>，登陆时，可能需要用户名和密码。默认的用户名和密码均为admin

登陆后界面如下即为成功：



4、按CTRL+C可关闭ActiveMQ

### ActiveMQ目录结构



Bin – 存放启动脚本，执行文件

Conf -- 存放配置文件

Data – 存放日志文件和持久消息

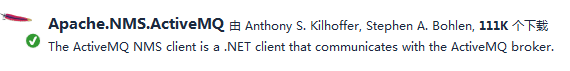
Lib – 存放activeMQ使用到的库

Webapps – 存放activeMQ web console

### 代码测试和使用样例

本项目是C#的项目，因此使用C#编写代码样例。Java请自行查找资料。

通过Nuget安装 Apache.NMS.ActiveMQ



### JMS消息

每个消息有三部分组成：Header、Properties和Body

Header：所有的消息支持the same set of header fields。Header fields contain values used by both clients and providers to identify and route messages.

Properties：用户自定义的属性信息。

Body:消息体

#### 消息头

JMS部分消息头如下：

JMSDestination:消息发送的目的地

JMSDeliveryMode:消息的模式：Persistent或者NonPersistent

JMSExpiration:消息过期时间

JMSMessageID:消息ID，必须以ID:开头

JMSPriority:消息优先级

JMSTimestamp:消息发送的时间

JMSCorrelationID:消息ID用来与之前消息来年系起来。provider-specific message ID will begin with the ID: prefix

JMSReplyTo：标记回复发送的目的地

//除此之外还可以自定义消息头

#### 消息类型

MapMessage:消息体使用键值对

TextMessage:消息体使用字符串

BytesMessage:消息体使用字节数组

[BytesMessage](https://docs.oracle.com/javaee/7/api/javax/jms/BytesMessage.html), [MapMessage](https://docs.oracle.com/javaee/7/api/javax/jms/MapMessage.html), [ObjectMessage](https://docs.oracle.com/javaee/7/api/javax/jms/ObjectMessage.html), [StreamMessage](https://docs.oracle.com/javaee/7/api/javax/jms/StreamMessage.html), [TextMessage](https://docs.oracle.com/javaee/7/api/javax/jms/TextMessage.html)

* Stream - A StreamMessage object's message body contains a stream of primitive values in the Java programming language ("Java primitives"). It is filled and read sequentially.
* Map - A MapMessage object's message body contains a set of name-value pairs, where names are String objects, and values are Java primitives. The entries can be accessed sequentially or randomly by name. The order of the entries is undefined.
* Text - A TextMessage object's message body contains a java.lang.String object. This message type can be used to transport plain-text messages, and XML messages.
* Object - An ObjectMessage object's message body contains a Serializable Java object.
* Bytes - A BytesMessage object's message body contains a stream of uninterpreted bytes. This message type is for literally encoding a body to match an existing message format.

#### 队列与主题

队列 queue:消息只发送到一个消费者。队列存储所有的消息，直到他们被发送或者过期。多个消费者可以监听同一个队列。多个生产者可以向同一个队列发送消息。

主题 topic:主题中消息会发送到所有的订阅者。Topic不会保存消息。如果订阅者掉线将会丢失消息，即使消息持久且没有过期。 Durable subscription持久订阅，在断线后可以保存消息，上线后重新收到未过期消息。多个生产者可以向同一个主题发送消息

#### 请求/响应

请求、响应是通过JMSReplyTo和JMSCorrelationID实现，JMSReplyTo指定响应发送的目的站，JMSMessageID指定请求消息的ID. 客户端设置JMSReplyTo和JMSCorrelationID，服务器设置相同的JMSCorrelationID，并发送ReplyTo.

不建议使用该特性。

#### Destination

Destination:生命周期matches the connection from which the session was created.

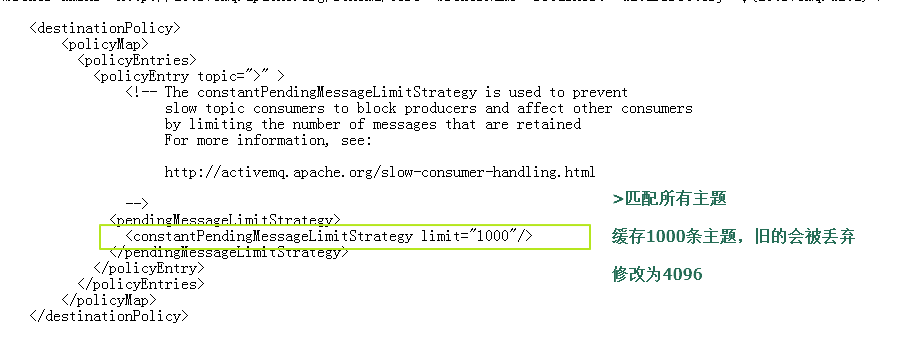
Temporary destination(临时目的站)生命周期as long as the connection that created them and only connection that created them can create consumers去消费。

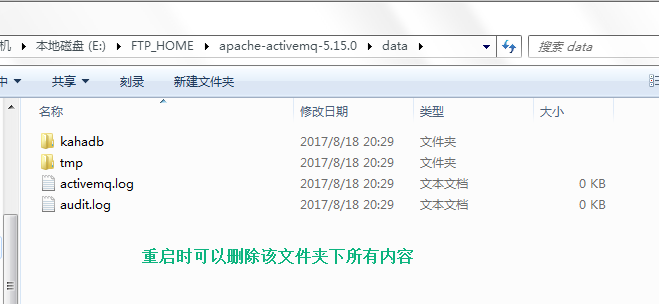
当持久消息被接收，它将从持久存储上删除。持久会有性能影响。

#### 持久和非持久

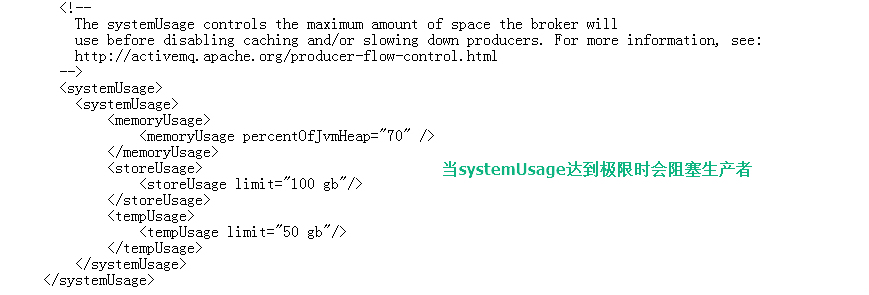
持久消息放到硬盘上，非持久消息放到内存，当内存告急，非持久消息会写入临时文件。重启后持久消息会被恢复，非持久的临时文件直接删除。当内存或者文件超过指定限制会阻塞生产者。尤其是非持久主题，会发给每个消费者，才删除消息。这样慢的消费者会拖累快的消费者。Broker增加了缓存的配置，缓存一定的数量，超过这个旧的会被丢弃。

默认持久保存到data文件夹，整个文件夹重启时可以删除。重启时自动重建。





当systemUsage达到极限时会阻塞生产者。



#### Advisory Message

Advisory Message(咨询消息或者监控消息)用来监控系统。 现有的Advisory Message有：

Consumers,producers and connections starting and stopping

Temporary destinations being created and destroyed

Message expiring on topics and queues（消息过期）

Brokers sending messages to destinations with no consumers

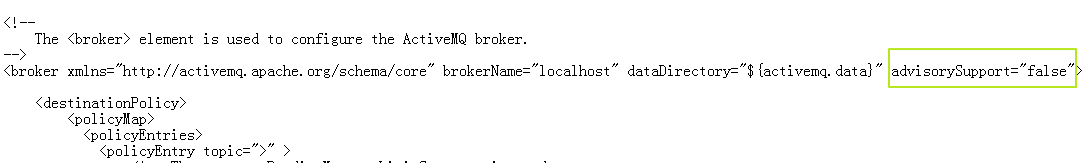
监控消息都是主题，以ActiveMQ.Advisory开头

监控消息列表：

|  |  |
| --- | --- |
| Advisory Topics | Description |
| ActiveMQ.Advisory.Connection | Connection start & stop messages. |
| ActiveMQ.Advisory.Producer.Queue | Producer start & stop messages on a Queue. |
| ActiveMQ.Advisory.Producer.Topic | Producer start & stop messages on a Topic. |
| ActiveMQ.Advisory.Consumer.Queue | Consumer start & stop messages on a Queue. |
| ActiveMQ.Advisory.Consumer.Topic | Consumer start & stop messages on a Topic. |
| ctiveMQ.Advisory.Queue | Queue create & destroy. |
| ActiveMQ.Advisory.Topic | Topic create & destroy. |
| ActiveMQ.Advisory.TempQueue | Temporary Queue create & destroy. |
| ActiveMQ.Advisory.TempTopic | Temporary Topic create & destroy. |
| ActiveMQ.Advisory.Expired.Queue | Expired messages on a Queue. |
| ActiveMQ.Advisory.Expired.Topic | Expired messages on a Topic. |
| ActiveMQ.Advisory.NoConsumer.Queue | No consumer is available to process messages being sent on a Queue. |
| ActiveMQ.Advisory.NoConsumer.Topic | No consumer is available to process messages being sent on a Topic. |

监控列表默认是启动的。禁用配置如下：

<broker advisorySupport="false">

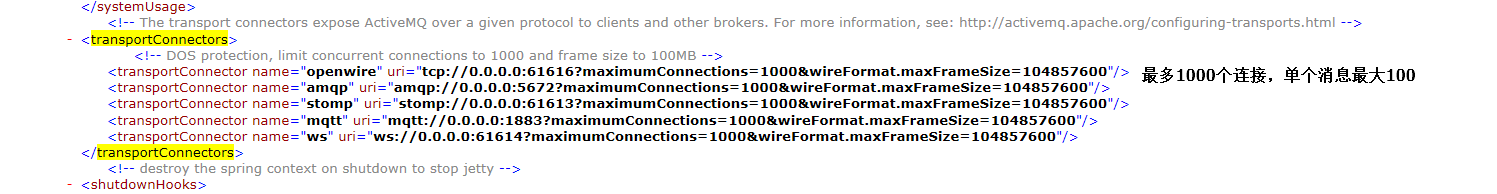


Advisory Message是非持久的

Advisory Message是非持久的，不过期的

### ActiveMQ配置

Conf/activemq.xml配置文件



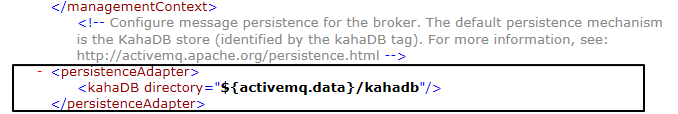
配置项更改后，必须重启才能生效。

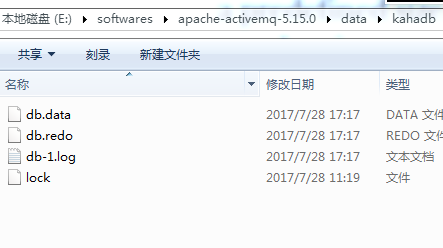
修改openwire支持的最大连接数和单个消息的最大字节数。activeMQ默认使用OpenWire协议交换消息。

failover:(uri1,...,uriN)?key=value&Key=Value

默认重连随机选择一个uri,初始时间间隔10ms，之后加倍，直到最大30000ms。无限重连。

KahaDB消息存储(推荐)





Log日志文件，保存消息，当一个满了，创建下一个，When there are no more references to any of the messages in the data log file, it’ll be deleted or archived.默认是删除。

Db.data是Btree，用来索引log中的消息

*db.redo* —This is the redo file, used for recovering the BTree indexes if the KahaDB message store starts after a hard stop.

Lock：锁文件，用来做Cluster

### Cluster

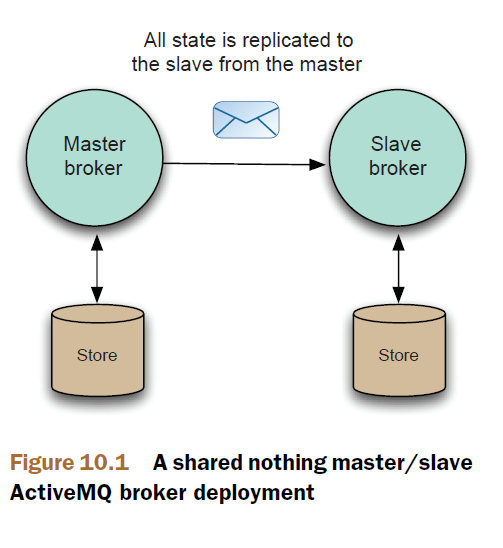
#### 主从模式

分为不共享存储模式和共享消息存储模式。

*shared storage*, where multiple ActiveMQ brokers can connect to the shared message store (a relational database or a shared file system) but only one broker will be active at a time.

##### shared nothing master/slave

主从均有自己独立的消息存储。



Slave配成连接Master。所有的命令（包括消息、通知和订阅等）从主复制到从上。Slave需要在启动时连接Master，因此先启动Master，否则有可能主从不一致。a master is allowed to have only one slave, and that a slave itself can’t have another slave.

当生产者发送持久消息到Master，生产者需要等待Master的确认。Master将消息复制到Slave上。复制后，Master本地处理消息之后给生产者发送确认。

当Master失败后，Slave变成Master.客户端配置成：

failover://(tcp://masterhost:61616,tcp://slavehost:61616)?randomize=false

配置Slave，Master不需要配置。

<services>

<masterConnector remoteURI="tcp://remotehost:62001" userName="Rob" password="Davies"/>

</services>

RemoterURI：指定Master的地址 username password指定master用户名密码

该模式缺点是主挂了，需要人工参与。一个主只能一个从。

##### Shared storage master/slave

多个实例共享存储，只有一个运行。不需要人参与。可以共享数据库或共享文件

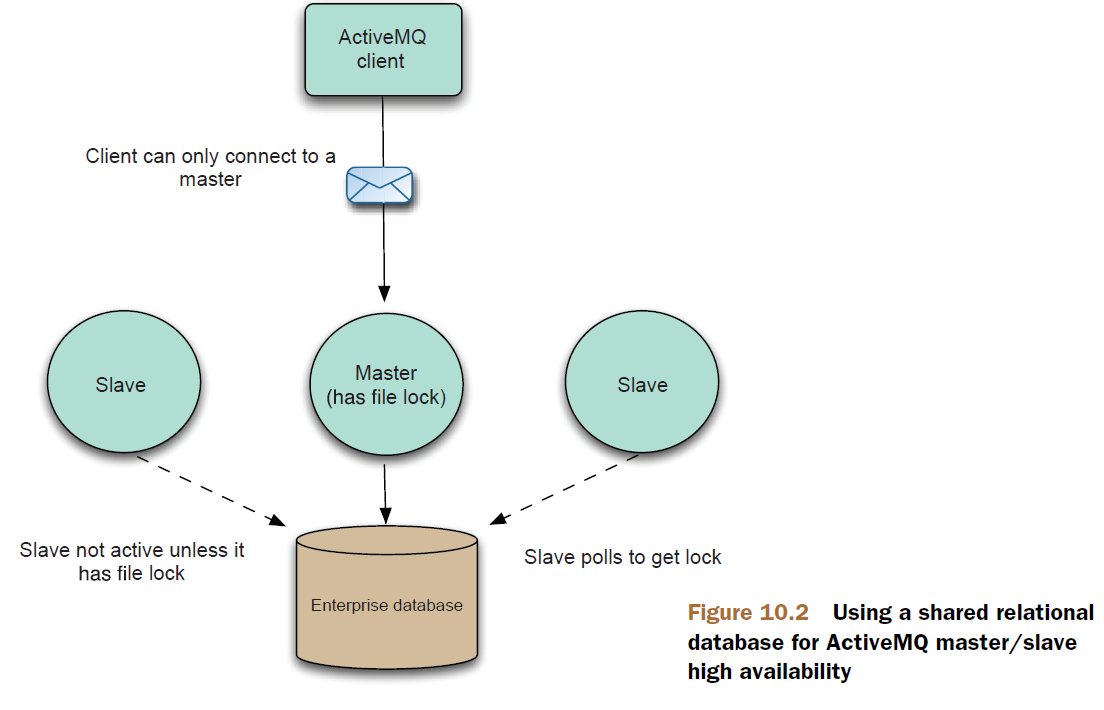
共享数据库。所有的实例都使用相同的配置。

When an ActiveMQ message broker uses a relational database, it grabs an exclusive lock on a table to ensure that no other ActiveMQ broker can access the database at the same time. If you’re running more than one broker that’s trying to access the same database, only the first broker to connect will grab the lock. Any subsequent brokers will poll until they can get access to the lock. While in this polling state, the ActiveMQ broker assumes that it’s a slave, so it won’t start any transport connections or network connections. If a master broker fails, a slave broker will be able to grab the

lock on the database and will then take over as the new master broker. Since all the

ActiveMQ brokers are using the same shared database, no additional manual intervention

is required to introduce new brokers or remove existing ones.



SHARED FILE SYSTEM MASTER/SLAVE

使用kahaDB做消息存储，使用共享文件夹。

可以启动任意多个实例，第一个获取文件锁的是Master，其它是slave。

### 高级特性

ActiveMQ的Destination支持层级，层级使用.分割。\*用来匹配一个元素，>Matches one or all trailing elements。以上是用来接收消息的

用来发送消息使用Composite desitination。组合地址使用,分割。

### 3.9 重发和死信队列

消息在一下情况下重发：

A transacted session is used and **rollback()** is called.

A transacted session is closed before **commit()** is called.

A session is using **CLIENT\_ACKNOWLEDGE** and **Session.recover()** is called.

A client connection times out (perhaps the code being executed takes longer than the configured time-out period).

当消息重发超过指定次数（如listener处理异常），就会被放到DLQ，死信队列。死信队列名字为： **ActiveMQ.DLQ。**所有未发送的消息都放到这个队列。

可以配置过期消息不发送到DLQ。默认是发送的。

<!-- Set the following policy on all queues using the '>' wildcard -->

<policyEntry queue=">">

<!--

Tell the dead letter strategy not to process expired messages

so that they will just be discarded instead of being sent to

the DLQ

-->

<deadLetterStrategy>

<sharedDeadLetterStrategy processExpired="false" />

</deadLetterStrategy>

</policyEntry>



默认非持久消息不会放到死信队列。默认死信队列的消息是不过期的。

主题过期会丢弃，不会放到死信队列。建议主题不持久。

### 3.10 开发建议

禁用死信队列。

将默认最大连接数1000修改为2048

主题非持久，过期时间10分钟。

队列持久，过期时间30分钟。

消费者长连接，单独线程处理取消息和处理消息。

发送者可以短连接。

最大单个消息默认是100M.