ActiveMQ基本介绍

1、 ActiveMQ服务器工作模型

通过 ActiveMQ 消息服务交换消息。消息生产者将消息发送至消息服务，消息消费者则从消息服务接收这些消息。这些消息传送操作是使用一组实现 ActiveMQ 应用编程接口 (API) 的对象来执行的。

ActiveMQ 客户端使用 ConnectionFactory 对象创建一个连接，向消息服务发送消息以及从消息服务接收消息均是通过此连接来进行。Connection 是客户端与消息服务的活动连接。创建连接时，将分配通信资源以及验证客户端。这是一个相当重要的对象，大多数客户端均使用一个连接来进行所有的消息传送。 连接用于创建会话。 Session 是一个用于生成和使用消息的单线程上下文。它用于创建发送的生产者和接收消息的消费者，并为所发送的消息定义发送顺序。会话通过大量确认选项或通过事务来支持可靠传送。 客户端使用 MessageProducer 向指定的物理目标（在 API 中表示为目标身份对象）发送消息。生产者可指定一个默认传送模式（持久性消息与非持久性消息）、优先级和有效期值，以控制生产者向物理目标发送的所有消息。

同样，客户端使用 MessageConsumer 对象从指定的物理目标（在 API 中表示为目标对象）接收消息。消费者可使用消息选择器，借助它，消息服务可以只向消费者发送与选择标准匹配的那些消息。 消费者可以支持同步或异步消息接收。异步使用可通过向消费者注册 MessageListener 来实现。当会话线程调用 MessageListener 对象的 onMessage 方法时，客户端将使用消息。

2、 ActiveMQ 消息传送模型

ActiveMQ 支持两种截然不同的消息传送模型： PTP（即点对点模型）和 Pub/Sub（即发布 / 订阅模型），分别称作： PTP Domain 和Pub/Sub Domain。

PTP（使用 Queue 即队列目标） 消息从一个生产者传送至一个消费者。在此传送模型中，目标是一个队列。消息首先被传送至队列目标，然后根据队列传送策略，从该队列将消息传送至向此队列进行注册的某一个消费者，一次只传送一条消息。可以向队列目标发送消息的生产者的数量没有限制，但每条消息只能发送至、并由一个消费者成功使用。如果没有已经向队列目标注册的消费者，队列将保留它收到的消息，并在某个消费者向该队列进行注册时将消息传送给该消费者。

Pub/Sub（使用 Topic 即主题目标） 消息从一个生产者传送至任意数量的消费者。在此传送模型中，目标是一个主题。消息首先被传送至主题目标，然后传送至所有已订阅此主题的活动消费者。可以向主题目标发送消息的生产者的数量没有限制，并且每个消息可以发送至任意数量的订阅消费者。主题目标也支持持久订阅的概念。持久订阅表示消费者已

向主题目标进行注册，但在消息传送时此消费者可以处于非活动状态。当此消费者再次处于活动状态时，它将接收此信息。如果没有已经向主题目标注册的消费者，主题不保留其接收到的消息，除非有非活动消费者注册了持久订阅。

3、ActiveMQ 消息选择器

ActiveMQ 提供了一种机制，使用它，消息服务可根据消息选择器中的标准来执行消息过滤。生产者可在消息中放入应用程序特有的属性，而消费者可使用基于这些属性的选择标准来表明对消息是否感兴趣。这就简化了客户端的工作，并避免了向不需要这些消息的消费者传送消息的开销。然而，它也使得处理选择标准的消息服务增加了一些额外开销。消息选择器是用于 MessageConsumer的过滤器，可以用来过滤传入消息的属性和消息头部分（但不过滤消息体），并确定是否将实际消费该消息。消息选择器是一些字符串，它们基于某种语法，而这种语法是SQL-92的子集。可以将消息选择器作为 MessageConsumer 创建的一部分。

4、ActiveMQ 消息签收

在不带事务的 Session 中，一条消息何时和如何被签收取决于 Session 的设置。

1. Session.AUTO_ACKNOWLEDGE

当客户端从 receive 或 onMessage成功返回时， Session 自动签收客户端的这条消息的收条。在 AUTO_ACKNOWLEDGE 的 Session 中，同步接收 receive 是上述三个阶段的一个例外，在这种情况下，收条和签收紧随在处理消息之后发生。

2. Session.CLIENT_ACKNOWLEDGE

客户端通过调用消息的 acknowledge 方法签收消息。在这种情况下，签收发生在 Session 层面：签收一个已消费的消息会自动地签收这个 Session 所有已消费消息的收条。

3. Session.DUPS_OK_ACKNOWLEDGE

此选项指示 Session 不必确保对传送消息的签收。它可能引起消息的重复，但是降低了 Session 的开销，所以只有客户端能容忍重复的消息，才可使用（如果 ActiveMQ 再次传送同一消息，那么消息头中的 JMSRedelivered 将被设置为 true ）。客户端成功接收一条消息的标志是这条消息被签收。成功接收一条消息一般包括如下三个阶段：

1. 客户端接收消息；
2. 客户端处理消息；
3. 消息被签收。签收可以由 ActiveMQ 发起，也可以由客户端发起，取决于 Session 签收模式的设置。 在带事务的 Session 中，签收自动发生在事务提交时。如果事务回滚，所有已经接收的消息将会被再次传送。

5、ActiveMQ 消息传送模式

ActiveMQ 支持两种消息传送模式： PERSISTENT和 NON_PERSISTENT

两种。
1. PERSISTENT(持久性消息)
这是 ActiveMQ 的默认传送模式，此模式保证这些消息只被传送一次和成功使用一次。对于这些消息，可靠性是优先考虑的因素。可靠性的另一个重要方面是确保持久性消息传送至目标后，消息服务在向消费者传送它们之前不会丢失这些消息。这意味着在持久性消息传送至目标时，消息服务将其放入持久性数据存储。如果消息服务由于某种原因导致失败，它可以恢复此消息并将此消息传送至相应的消费者。虽然这样增加了消息传送的开销，但却增加了可靠性。
2. NON_PERSISTENT(非持久性消息)
保证这些消息最多被传送一次。对于这些消息，可靠性并非主要的考虑因素。此模式并不要求持久性的数据存储，也不保证消息服务由于某种原因导致失败后消息不会丢失。
有两种方法指定传送模式：
1. 使用 setDeliveryMode 方法，这样所有的消息都采用此传送模式；
2. 使用 send 方法为每一条消息设置传送模式。

6、ActiveMQ 优先级设置

通常，可以确保将单个会话向目标发送的所有消息按其发送顺序传送至消费者。然而，如果为这些消息分配了不同的优先级，消息传送系统将首先尝试传送优先级较高的消息。
有两种方法设置消息的优先级：
1. 使用 setDeliveryMode 方法，这样所有的消息都采用此传送模式；
2. 使用 send 方法为每一条消息设置传送模式；
消息优先级从 0-9 十个级别， 0-4 是普通消息， 5-9 是加急消息。如果不指定优先级，则默认为 4。JMS 不要求严格按照这十个优先级发送消息，但必须保证加急消息要先于普通消息到达。

7、ActiveMQ 消息过期设置

允许消息过期 。默认情况下，消息永不会过期。如果消息在特定周期内失去意义，那么可以设置过期时间。
有两种方法设置消息的过期时间，时间单位为毫秒：
1. 使用 setTimeToLive 方法为所有的消息设置过期时间；
2. 使用 send 方法为每一条消息设置过期时间。
消息过期时间， send 方法中的 timeToLive 值加上发送时刻的 GMT 时间值。如果 timeToLive 值等于零，则 JMSExpiration 被设为零， 表示该消息永不过期。如果发送后，在消息过期时间之后消息还没有被发送到目的地，则该消息被清除。

8、ActiveMQ 持久订阅设置

通过为发布者设置 PERSISTENT 传送模式，为订阅者时使用持久订阅，这样可以保证 Pub/Sub 程序接收所有发布的消息。
消息订阅分为非持久订阅 (non-durable subscription) 和持久订阅 (durable

subscription)　，非持久订阅只有当客户端处于激活状态，也就是和 ActiveMQ　保持连接状态才能收到发送到某个主题的消息，而当客户端处于离线状态，这个时间段发到主题的消息将会丢失，永远不会收到。
持久订阅时，客户端向　ActiveMQ 注册一个识别自己身份的　ID，当这个客户端处于离线时，　ActiveMQ 会为这个　ID　保存所有发送到主题的消息，当客户端再次连接到　ActiveMQ 时，　会根据自己的　ID 得到所有当自己处于离线时发送到主题的消息。持久订阅会增加开销，同一时间在持久订阅中只有一个激活的用户。　　建立持久订阅的步骤：

1. 为连接设置一个客户　ID ；
2. 为订阅的主题指定一个订阅名称；

上述组合必须唯一。

9、ActiveMQ 异步发送消息

ActiveMQ 支持生产者以同步或异步模式发送消息。使用不同的模式对 send方法的　反应时间有巨大的影响，反映时间是衡量　　ActiveMQ 吞吐量的重要因素，使用异步发送
可以提高系统的性能。　　在默认大多数情况下，　AcitveMQ 是以异步模式发送消息。例外的情况：在没有使用事务的情况下，生产者以 PERSISTEN传送模式发送消息。在这种情况下，　send方法都是同步的，并且一直阻塞直到　　ActiveMQ 发回确认消息：消息已经存储在持久性数据存储中。这种确认机制保证消息不会丢失，但会造成生产者阻塞从而影响反应时间。
高性能的程序一般都能容忍在故障情况下丢失少量数据。如果编写这样的程序，可以通过使用异步发送来提高吞吐量（甚至在使用　　PERSISTENT 传送模式的情况下）。

10、ActiveMQ 消费者特性

（1）消费者异步分派
在　ActiveMQ4　中，支持　ActiveMQ　以同步或异步模式向消费者分派消息。这样的意义：可以以异步模式向处理消息慢的消费者分配消息；以同步模式向处理消息快的消费者分配消息。　　ActiveMQ 默认以同步模式分派消息（setDispatchAsync(false)　　或TEST.QUEUE?consumer.dispatchAsync=false　　），这样的设置可以提高性能。但是对于处理消息慢的消费者，需要以异步模式分派。
（2）消费者优先级
在　ActveMQ　分布式环境中，在有消费者存在的情况下，如果更希望 ActveMQ 发送消给消费者而不是其他的　　ActveMQ 到ActveMQ 的传送，可以如下设置：　TEST.QUEUE?consumer.prority=10
（3）独占消费者
ActiveMQ 维护队列消息的顺序并顺序把消息分派给消费者。但是如果建立了多个 Session 和 MessageConsumer，那么同一时刻多个线程同时

从一个队列中接收消息时就并
不能保证处理时有序。　有时候有序处理消息是非常重要的。　ActiveMQ4 支持独占的消费。　ActiveMQ　挑选一个　MessageConsumer，　并把一个队列中所有消息按顺序分派给它。　如果消费者发生故障，那么 ActiveMQ　将自动故障转移并选择另一个消费者。可以如下设置：
TEST.QUEUE?consumer.exclusive=true

11、ActiveMQ 消息预取机制

ActiveMQ 的目标之一就是高性能的数据传送，所以　ActiveMQ　使用“ 预取限制 ”来 控制有多少消息能及时的传送给任何地方的消费者。　一旦预取数量达到限制，那么就不会有消息被分