# ActiveMQ技术预研报告

## 1.文档介绍

### 1.1 文档目的

对前期进行的ActiveMQ研究进行总结和分享ActiveMQ方面的心得。

### 1.2 文档范围

适合技术中心技术人员，可以作为技术参考。

### 1.3 参考文档

ActiveMQ官方网站 http://activemq.apache.org/

开源中国社区 http://www.oschina.net/p/activemq

百度百科    http://baike.baidu.com/view/157103.htm?fr=ala0\_1\_1

### 1.4 术语与缩写解释

MQ：消息队列，很多时候指消息中间件服务器

ActiveMQ：ActiveMQ是一个开放源码基于Apache 2.0 licenced 发布并实现了JMS 1.1的消息中间件

JMS：jms即Java消息服务（Java Message Service）,一种协议标准

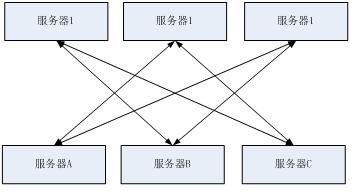
URI：资源标志符（Uniform Resource Identifier, 简称”URI”）

## 2. 研究背景介绍

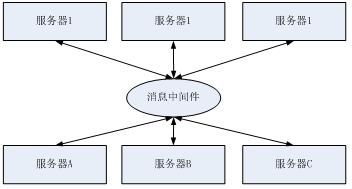
为了快速有效的丰富汇讯的功能，减少开发时间和开发成本，我们将web的技术融合起来，技术融合中出现一个问题就是消息实时通知汇讯客户端的处理消息的流转。简单的方法可以通过web服务器和IM服务器进行直连来完成通信，但这样的架构方案让web和IM过于依赖，为了提供更好的扩展性，降低耦合，考虑引入消息中间件来作为两者的中介，减少依赖，由此我们需要对消息中间件进行一些分析和研究。

## 3. ActiveMQ技术概述

         ActiveMQ是一个开放源码基于Apache 2.0 licenced 发布并实现了JMS 1.1的消息中间件，应用中引入中间件的好处是减少服务器之间的依赖关系，提高扩展性，在没有引入消息中间件的情况可能出现如下图所示：



出现服务器多依赖的情况，不方面扩展，而引入消息中间件后如



         从图中可以看出引入消息中间件后，每个服务器只依赖于消息中间件，而且在应用中这种依赖关系式一种弱依赖关系，为什么这么讲呢？请看下节对消息中间件的分享报告。

## 4.ActiveMQ的技术分析

         ActiveMQ实现了sub/pub（订阅/发布）的机制，实现jms协议，sub/pub属于设计模式中典型的观察者模式，每台服务器只需要订阅自己希望得到的消息，而不必要轮询服务器是否有自己需要的消息，也不需要知道消息发布者是谁，而发布消息的一端也不需知道谁是消息的接收端，有多少给接受者，这些都不重要，只要将消息发布到消息中间件就可以了。

ActiveMQ消息发布和订阅的类型分为BytesMessage(二进制消息流的方式)、MapMessage（一种键值对方式的消息）、 ObjectMessage（一种序列化对象的消息形式）、TextMessage（字符流的消息类型），可以根据需要进行消息类型选择，当然对于同一个消息发布/订阅双方需要采用一致的消息类型，同一个服务器可以采用多种消息格式，不过不同的消息格式需要奖励不同的发布/和订阅者。这样一来应用就相当灵活，可以根据需要进行类型选择，不过选择ObjectMessage类型的时候应该注意，这个消息类型主要是序列化的Java对象，所以不支持不同的语言进行类型数据交换。

### 4.1ConnectionFactory 接口（连接工厂）

用户用来创建到JMS提供者的连接的被管对象。JMS客户通过可移植的接口访问连接，这样当下层的实现改变时，代码不需要进行修改。 管理员在JNDI名字空间中配置连接工厂，这样，JMS客户才能够查找到它们。根据消息类型的不同，用户将使用队列连接工厂，或者主题连接工厂。

1. Connection 接口（连接）

连接代表了应用程序和消息服务器之间的通信链路。在获得了连接工厂后，就可以创建一个与JMS提供者的连接。根据不同的连接类型，连接允许用户创建会话，以发送和接收队列和主题到目标。

1. Destination 接口（目标）

目标是一个包装了消息目标标识符的被管对象，消息目标是指消息发布和接收的地点，或者是队列，或者是主题。JMS管理员创建这些对象，然后用户通过JNDI发现它们。和连接工厂一样，管理员可以创建两种类型的目标，点对点模型的队列，以及发布者／订阅者模型的主题。

1. MessageConsumer 接口（消息消费者）

由会话创建的对象，用于接收发送到目标的消息。消费者可以同步地（阻塞模式），或异步（非阻塞）接收队列和主题类型的消息。

1. MessageProducer 接口（消息生产者）

由会话创建的对象，用于发送消息到目标。用户可以创建某个目标的发送者，也可以创建一个通用的发送者，在发送消息时指定目标。

### 4.2Message 接口（消息）

是在消费者和生产者之间传送的对象，也就是说从一个应用程序创送到另一个应用程序。一个消息有三个主要部分：

1. 消息头（必须）：包含用于识别和为消息寻找路由的操作设置。
2. 一组消息属性（可选）：包含额外的属性，支持其他提供者和用户的兼容。可以创建定制的字段和过滤器（消息选择器）。
3. 一个消息体（可选）：允许用户创建五种类型的消息（文本消息，映射消息，字节消息，流消息和对象消息）。
4. 消息接口非常灵活，并提供了许多方式来定制消息的内容。

### 4.3Session 接口（会话）

表示一个单线程的上下文，用于发送和接收消息。由于会话是单线程的，所以消息是连续的，就是说消息是按照发送的顺序一个一个接收的。会话的好处是它支持事务。如果用户选择了事务支持，会话上下文将保存一组消息，直到事务被提交才发送这些消息。在提交事务之前，用户可以使用回滚操作取消这些消息。一个会话允许用户创建消息生产者来发送消息，创建消息消费者来接收消息。

### 4.4MessageListener接口（消息监听者）

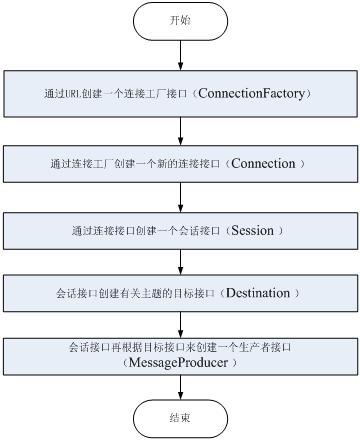
这是为一个消息消费者的消息监听接口，生产者必选设置消息监听者，否则消息将不处理，当客户端接收到消息后，会通过调用消息监听者的接口来进行相应的消息处理，一般在开发过程中通过重载的方式重新定义监听着的onMessage虚接口，来完成消息的监听和处理。

## 5．ActiveMQ的C++客户端的实现

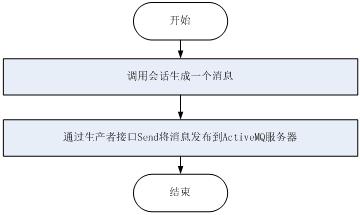
         ActiveMQ提供了c++的client开发库支持，这样我们实现起来就比较简单了，下面我们来看看发布和订阅的简单例子。

### 5.1 pub（发布）端的简单实现过程

首先根据传入的URI创建一个发布接口，创建过程如下：

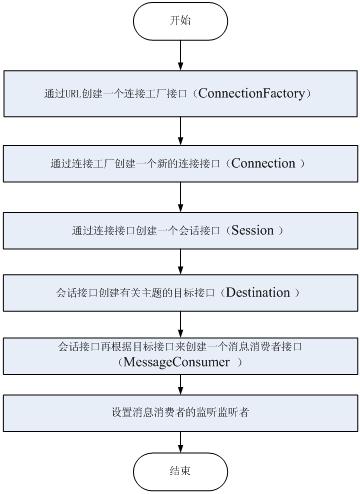


然后调用发布接口发布消息，发布过程如下：



### 5.2 sub（订阅）端的简单实现过程

订阅者的创建过程和发布者的创建过程基本一样，不过最后创建的不是生产者接口，而是消息消费者接口（MessageConsumer）,创建流程如下：



         在实现时，我们重载消息消费者的监听者，并设置消息消费者的监听接口为我们实现的监听接口，再重载监听者onMessage接口来进行消息处理。这样，只要客户端接收到消息，就会调用我们的监听者的onMessage接口，我们就可以在这个接口进行相应的处理，完成消息接收处理过程。

### 5.3 消息处理过程

         a) 消息订阅

需要订阅某主题的客户端实现订阅过程，产生消息消费者的监听者实例，并实现消息处理过程。

         b) 消息发布

         需要发布消息的客户端实现消息发布的过程，等到一个生产者实力，并通过生产者接口向消息中间件发布消息。

         c) 服务器消息转发

消息中间件服务器收到生产者发送过来的消息后，查找是否有该类型主题消息的订阅者，有则分别发送消息。

         d) 订阅者消息处理

订阅者客户端收到消息中间件服务器发送过来的的消息后，调用监听者onMessage接口完成消息处理

## 6．总结

         对ActiveMQ的研究尚浅，也许有些理解不当之处，欢迎大家指出，一起学习，消息中间件在跨语言和跨平台，服务器间解耦都起到了比较到的作用。值得我们去学习，在此感谢所有提供免费网络资源的网站，感谢开源中国社区。