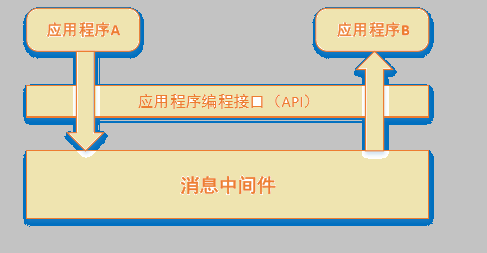
# 一 消息中间件

消息中间件是在分布式系统中完成消息的发送和接收的基础软件。

# 二 消息中间件的作用

消息中间件可利用高效可靠的消息传递机制进行平台无关的数据交流， 并基于数据通信来进行分布式系统的集成。通过提供消息传递和消息排队模型，可以在分布式环境下扩展进程间的通信。

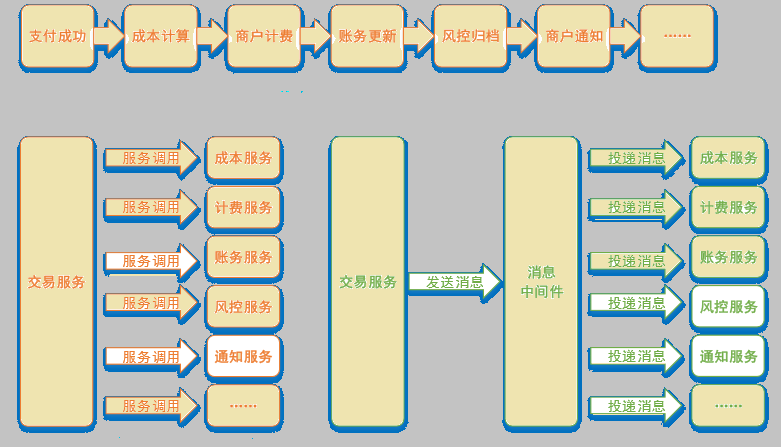
通过消息中间件，应用程序或组件之间可以进行可靠的异步通讯，从而 降低系统之间的耦合度，提高系统的可扩展性和可用性。



如上图所示，通过消息中间件来实现同构或者异构系统之间的异步消息通信。

三 使用场景

通过消息中间件，对服务间的调用进行解耦.



## 3.1传统的服务调用

都是同步的调用依次调用完成以后才给用户返回结果。

1 服务之间耦合度太高

2 服务调用过程中，本可以并发执行的服务存在等待

3 影响用户体验

4 数据一致性与操作的复制性得不到保障

## 3.2消息中间件的引入

1 消除高并发的请求,通过排队的方式来减少服务器的压力（如12306的抢票）

2 通过消息中间件让本可以并发的计算或者服务独立并发执行(比如邮件通知服务)

3 为数据失误留痕 保证可以二次执行,因为数据在消息队列中还有副本

4 让本不需要立即执行的复杂计算或者服务通知，数据处理，尽量放在后面做，保证用户体验。

5 如果服务端故障，可以保证消息不丢失，等服务端成功脱离故障以后，再从消息队列中获取消息。

# 三 JMS

JMS （Java Message Service） JMS是JavaEE中的一个关于消息的规范，是一套与具体平台无关的API

JMS提供者 ---- 连接面向消息中间件的，JMS接口的一个实现。

JMS客户 ------ 生产或消费消息的基于Java的应用程序或对象。

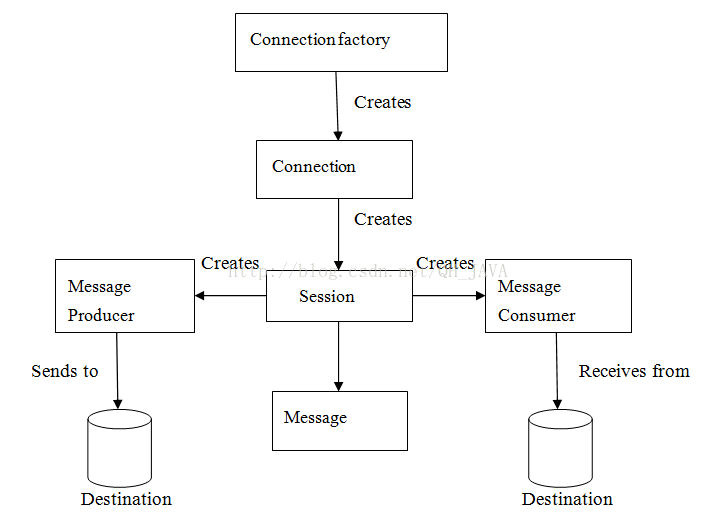
JMS生产者 ---- 创建并发送消息的JMS客户。

JMS消费者 ---- 接收消息的JMS客户。

JMS消息 ------ 可以在JMS客户之间传递的数据的对象

JMS队列 ------ 一个容纳那些被发送的等待阅读的消息的区域。

JMS主题 ------ 一种支持发送消息给多个订阅者的机制。



## 3.1、JMS管理对象

管理对象（Administered objects）是预先配置的JMS对象，由系统管理员为使用JMS的客户端创建，主要有两个被管理的对象：  
连接工厂（ConnectionFactory）,目的地（Destination）  
这两个管理对象由JMS系统管理员通过使用Application Server管理控制台创建，存储在应用程序服务器的JNDI名字空间或JNDI注册表。

## 3.2、连接工厂（ConnectionFactory）

客户端使用一个连接工厂对象连接到JMS服务提供者，它创建了JMS服务提供者和客户端之间的连接。JMS客户端（如发送者或接受者）会在JNDI名字空间中搜索并获取该连接。使用该连接，客户端能够与目的地通讯，往队列或话题发送/接收消息。让我们用一个例子来理解如何发送消息：

## 3.3、目的地（Destination）

目的地指明消息被发送的目的地以及客户端接收消息的来源。JMS使用两种目的地，队列和话题。

## 3.4、JMS连接

连接对象封装了与JMS提供者之间的虚拟连接，如果我们有一个ConnectionFactory对象，可以使用它来创建一个连接。

## 3.5、JMS 会话（Session）

Session是一个单线程上下文，用于生产和消费消息，可以创建出消息生产者和消息消费者。  
Session对象实现了Session接口，在创建完连接后，我们可以使用它创建Session。

## 3.6、JMS消息生产者

消息生产者由Session创建，用于往目的地发送消息。生产者实现MessageProducer接口，我们可以为目的地、队列或话题创建生产者；

## 3.7、JMS消息消费者

消息消费者由Session创建，用于接收目的地发送的消息。消费者实现MessageConsumer接口，我们可以为目的地、队列或话题创建消费者；

## 3.8、JMS消息监听器

JMS消息监听器是消息的默认事件处理者，他实现了MessageListener接口，该接口包含一个onMessage方法，在该方法中需要定义消息达到后的具体动作。通过调用setMessageListener方法我们给指定消费者定义了消息监听器.

# 四 JMS模型

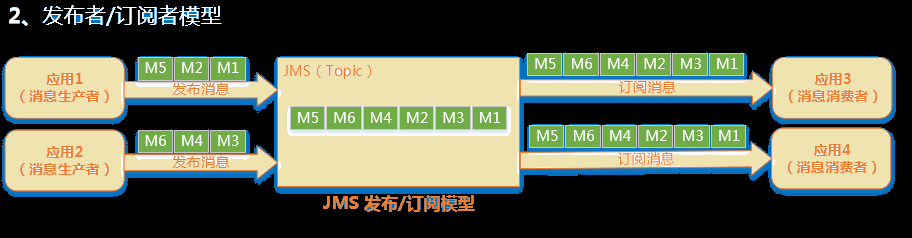


## 4.1点对点的消息队列模型

1 消息生产者生产消息发送到queue中，消息消费者从queue中取出消息并消费。

2 消息消费以后，queue中就不再存储,所以消息消费者不能够再消费已经消费了的消息。

3 queue支持存在多个消费者，但是一个消息只能被一个消费者消费。



## 4.2发布订阅模型

生产者把消息发布到Topic，同时有多个消费者消费该消息，发布到topic的消息，会被所有的消费者，订阅消费。

# 五 安装消息中间件

1. 安装 JDK 并配置环境变量（略）

JAVA\_HOME=/usr/local/java/jdk1.7.0\_72

1. 下载 Linux 版的 ActiveMQ（当前最新版 apache-activemq-5.11.1-bin.tar.gz）

$ wget http://apache.fayea.com/activemq/5.11.1/apache-activemq-5.11.1-bin.tar.gz

1. 解压安装

$ tar -zxvf apache-activemq-5.11.1-bin.tar.gz

$ mv apache-activemq-5.11.1 activemq-01

如果启动脚本 activemq 没有可执行权限，此时则需要授权（此步可选）

$ cd /usr/local/activeMQ/activeMq/bin/

$ chmod 755 ./activemq

4、 防火墙中打开对应的端口 ActiveMQ 需要用到两个端口

一个是消息通讯的端口（默认为 61616）

一个是管理控制台端口（默认为 8161）可在 conf/jetty.xml 中修改，如下：

<bean id="jettyPort" class="org.apache.activemq.web.WebConsolePort" init-method="start">

<!-- the default port number for the web console -->

<property name="host" value="0.0.0.0"/>

<property name="port" value="8161"/>

</bean>

# vi /etc/sysconfig/iptables

添加：

-A INPUT -m state --state NEW -m tcp -p tcp --dport 61616 -j ACCEPT

-A INPUT -m state --state NEW -m tcp -p tcp --dport 8161 -j ACCEPT

重启防火墙：

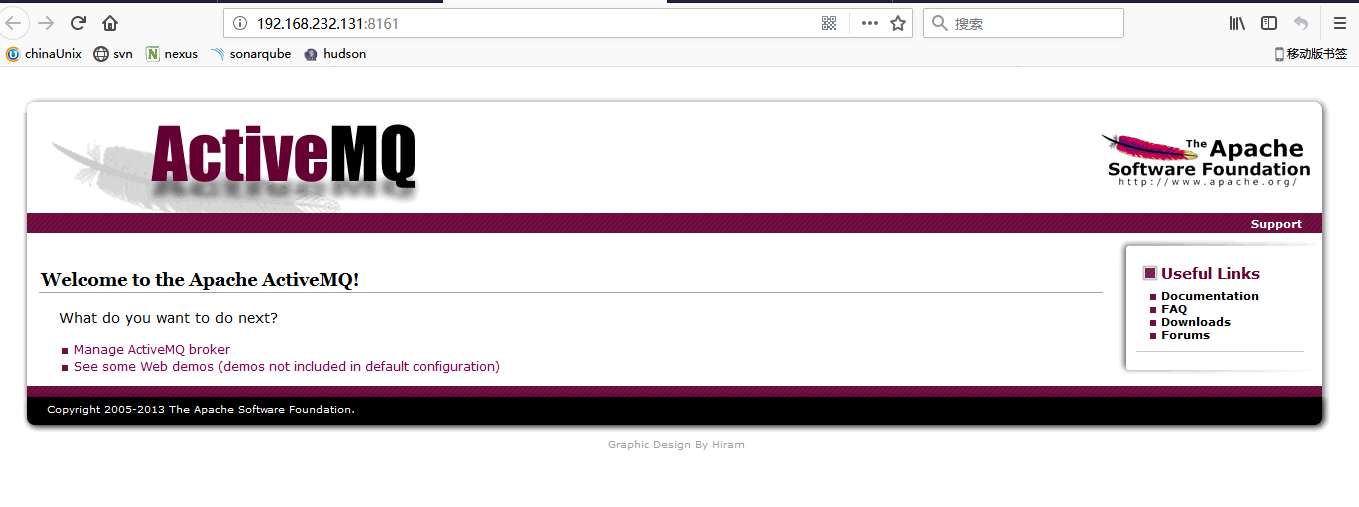
# service iptables restart

5启动

$ cd /usr/local/activeMQ/activeMq/bin

$ ./activemq start

1. 打开管理界面：http://192.168.232.131:8161/



如上图所示是ACtiveMQ的控制台程序



用户名密码是admin/admin

7、 安全配置（消息安全） ActiveMQ 如果不加入安全机制的话，任何人只要知道消息服务的具体地址(包括 ip，端口，消息地址

[队列或者主题地址]，)，都可以肆无忌惮的发送、接收消息。

在 conf/activemq.xml 文件中在 broker 标签最后加入以下内容即可

$ vim /usr/local/activeMQ/activeMq/conf/activemq.xml

<plugins>

<simpleAuthenticationPlugin>

<users>

<authenticationUser username="worn" password="worn" groups="users,admins"/>

</users>

</simpleAuthenticationPlugin>

</plugins>

定义了一个 worn 用户，密码为 worn，角色为 users,admins

设置 admin 的用户名和密码:

$ vi /usr/local/activeMQ/activeMq/conf/jetty.xml

<bean id="securityConstraint" class="org.eclipse.jetty.util.security.Constraint">

<property name="name" value="BASIC" />

<property name="roles" value="admin" />

<property name="authenticate" value="true" />

</bean>

确保 authenticate 的值为 true（默认）

控制台的登录用户名密码保存在 conf/jetty-realm.properties 文件中,内容如下:

$ vi /home/wusc/activemq-01/conf/jetty-realm.properties

# Defines users that can access the web (console, demo, etc.)

# username: password [,rolename ...]

注意:用户名和密码的格式，以上内容可以修改，也可以不改使用默认的。

用户名 : 密码 ,角色名

[root@localhost conf]# sh /usr/local/activeMQ/activeMq/bin/activemq restart

设置开机启动：

# vi /etc/rc.local

加入以下内容

## ActiveMQ

su - root-c '/usr/local/activeMQ/activeMq/bin/activemq start'