**Java消息中间件**

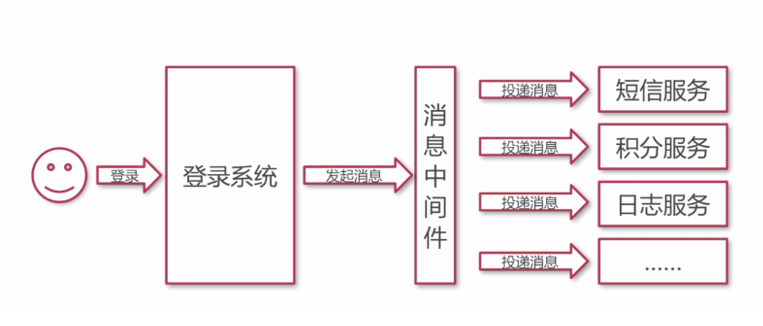
一．引入主题

小朋友小红有个睡前习惯，每次睡觉前，都要爸爸老王为她讲故事才能睡着。这个睡前听故事的习惯一直伴随着小红的成长。后来小红考上了初中，要离家住宿在学校。问题来了，小红住宿在学校就不能听爸爸讲故事了，这可把小红急坏了，她担心晚上没有故事听睡不着，影响自己的学业成绩。以是聪明的老王，利用微信公众号的功能，将故事发布到微信公众号，然后让女儿小红关注微信公众号，这样每晚睡觉前都可以听到爸爸讲的故事了。小红再也不用担心晚上没有故事听睡不着了。后来老王的微信公众号得到了推广，有了更多人来听老王将故事了。



其实这里所说的微信公众号，你可以把它看做是消息中间件，它具有发布订阅消息的功能。

生活中，常见的消息中间件还有很多，比如电视机，广播，收音机等等，我们都可以把它看做是消息中间件。有了消息中间件，可以为我们提供很多服务，比如消息的发布订阅，短信服务、积分服务、日志服务等等。



二、直奔主题

1. 什么是中间件？

中间件可以理解为是非底层操作系统软件，非业务应用软件，它不是直接给最终用户使用的，不能直接给客户带来价值的软件，这一类软件统称为中间件。

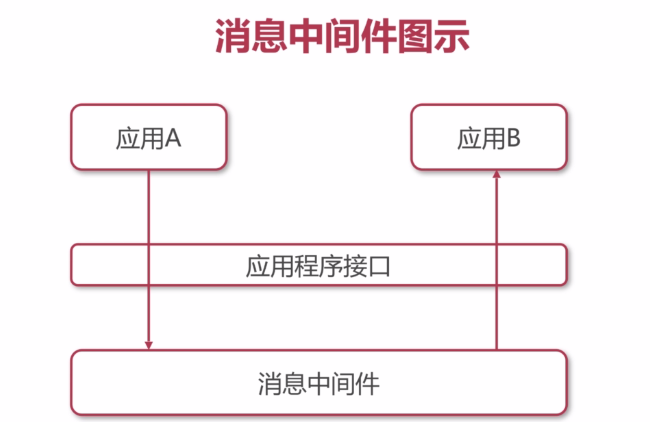
2. 中间件带来的好处

1. 利用中间件实现解耦、异步、横向扩展

2. 中间件具有顺序保证。安全可靠等优点

3. 什么是消息中间件？

消息中间件是一个关注于数据的发送与接收，它利用高效可靠的异步消息传输机制，实现集成分布式系统。



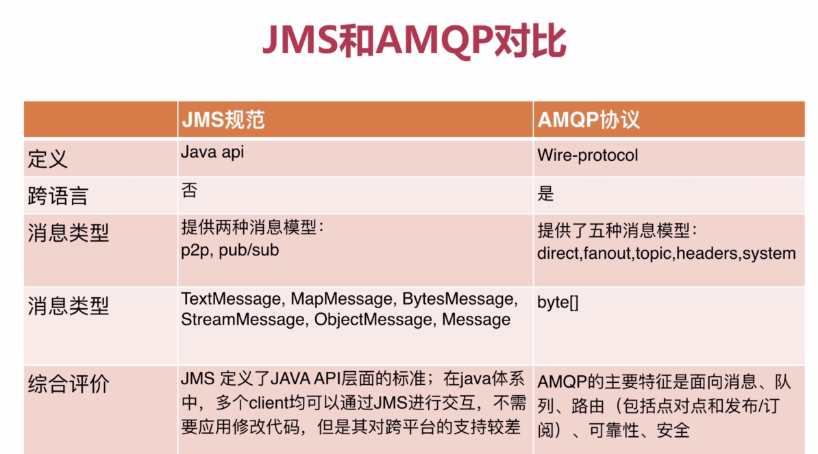
消息中间件——JMS概述

在Java中有一个消息中间件JMS，它是Java消息服务（Java Message Service）中间件，它是一个Java平台中关于面向消息中间件的API，用于在两个应用程序之间或者是分布式系统中发送消息，进行异步通信。

消息中间件——AMQP协议概述

AMQP（advanced message queuing protocol）是一个提供统一消息服务的应用层标准协议，基于此协议的客户端与消息中间件可以传递消息，它不受客户端/中间件不同产品，不同开发语言等条件的限制。

下面是JMS和AMQP的对比：



1. 常见的消息中间件对比

（1）ActiveMQ消息中间件

ActiveMQ是Apache出品，最流行的，能力强劲的开源消息总线。ActiveMQ是一个完全支持JMS1.1和J2EE 1.4规范的JMS Provider实现，尽管JSM规范很早就出了，但是JMS至今在J2EE应用中任然扮演着特殊地位。

ActiveMQ特性：

1. 它支持多种语言和协议编写客户端。支持的语言有：Java、C、C++、Python、PHP、Ruby、C#、Perl等。
2. 它支持的应用协议有：OpenWire、Stomp、REST、WS Notification、XMPP、AMQP等
3. 它完全支持JMS1.1 和 J2EE 1.4规范（持久化，XA消息，事务）
4. 虚拟主题、组合目的、镜像队列

（2）RabbitMQ消息中间件

RabbitMQ是一个开源的AMQP实现，服务器端用Erlang语言编写。用于在分布式系统中存储转发消息，在易用性、扩展性、高可用性等方面表现不俗。

RabbitMQ特性：

1. 它支持多种客户端，比如：Java、Python、JMS、C、PHP、.NET、Ruby、ActionScript等。
2. 它是AMQP的完整实现（vhost、Exchange、Binging、Routing Key等）。
3. 事务支持、发布确认
4. 消息持久化

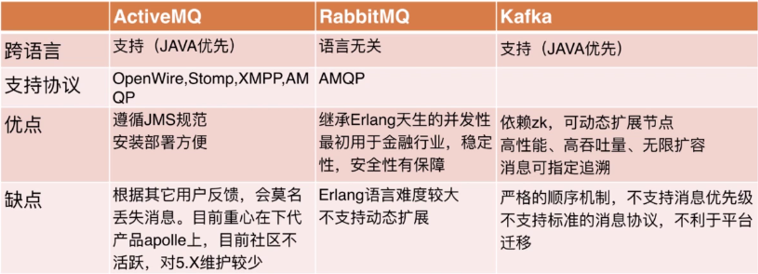
（3）Kafka消息中间件

Kafka是一种高吞吐量的分布式发布订阅消息系统，是一个分布式的、分区的、可靠的分布式日志存储服务。它通过一种独一无二的设计提供了一个消息系统的功能。

Kafka特性：

1. 通过O(1)的磁盘数据结构提供消息的持久化，这种结构对于即使数以TB的消息存储也能够保持长时间的稳定性。
2. 高吞吐量：即使是非常普通的硬件Kafka也可以支持每秒数百万的消息。
3. Partition、Consumer Group

下面是以上三类消息中间件的对比：

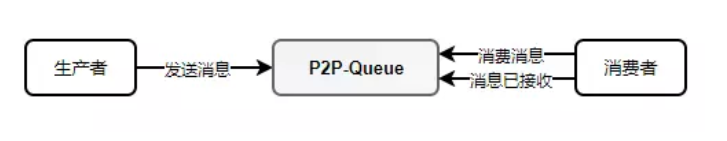


应用场景：

1. ActiveMQ消息中间件：适合中小企业级消息应用场景，它不适合上千个消息队列应用场景。
2. RabbitMQ消息中间件：适合对稳定性要求比较高的企业级应用。
3. Kafka消息中间件：它一般用在大数据日志处理货对实时性（少量延迟），可靠性（少量丢数据）要求稍低的场景使用。
4. 消息中间件的传输模式

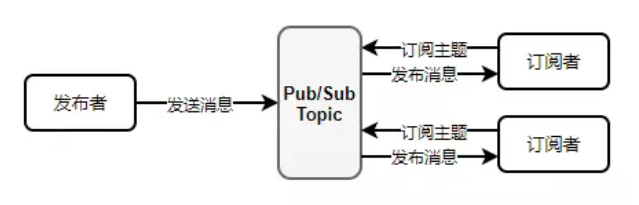
（1）点对点（p2p）模式：

点对点(p2p)模式有三个角色：消息队列（Queue），发送者(Sender)，接收者(Receiver)。发送者将消息发送到一个特定的队列中，等待接收者从队列中获取消息消耗。  
P2P的三个特点：  
一、每个消息只能被一个接收者消费，且消息被消费后默认从队列中删掉（也可以通过其他签收机制重复消费）。  
二、发送者和接收者之间没有依赖性，生产者发送消息和消费者接收消息并不要求同时运行。  
三、接收者在成功接收消息之后需向队列发送接收成功的确认消息。



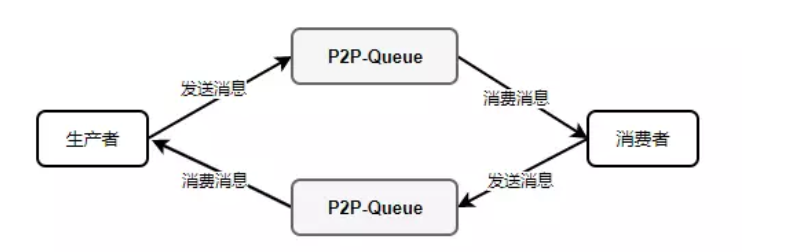
（2）消息订阅发布（Pub/Sub）模式

发布订阅(Pub/Sub)模式也有三个角色：主题（Topic），发布者（Publisher），订阅者（Subscriber）。发布者将消息发送到主题队列中，系统再将这些消息传递给订阅者。  
Pub/Sub的特点：  
一、每个消息可以被多个订阅者消费。  
二、发布者和订阅者之间存在依赖性。订阅者必须先订阅主题后才能接收到信息，在订阅前发布的消息，订阅者是接收不到的。  
三、非持久化订阅：如果订阅者不在线，此时发布的消息订阅者是也接收不到，即便订阅者重新上线也接收不到。  
四、持久化订阅：订阅者订阅主题后，即便订阅者不在线，此时发布的消息可以在订阅者重新上线后接收到的。



（3）双向应答模式

双向应答模式并不是消息中间件提供的一种通信模式，它是由于实际生成环境的需要，在原有的基础上做了改良。即消息的发送者也是消息的接收者。消息的接收者也是消息的发送者。



三、进入高潮

1、JMS相关概念

提供者：实现JMS规范的消息中间件服务器

客户端：发送或接收消息的应用程序

生产者/发布者：创建并发送消息的客户端

消费者/订阅者：接收并处理消息的客户端

消息：应用程序之间传递的数据内容

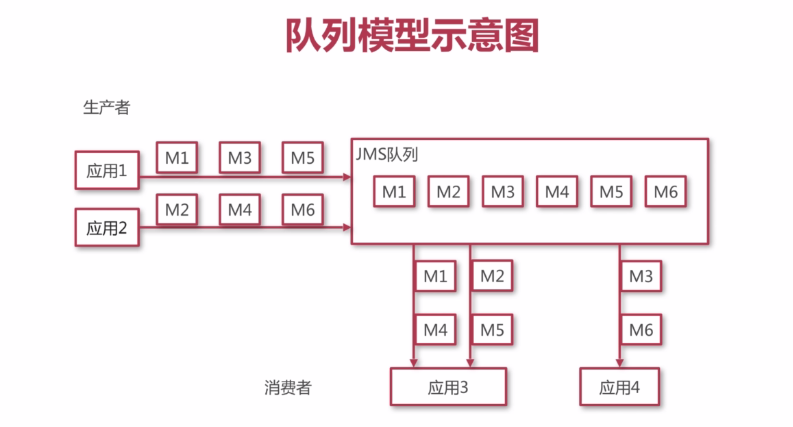
消息模式：在客户端之间传递消息的方式，JMS中定义了主题模式和队列模式。

2. JMS消息模式——队列模式

（1）队列模式的客户端包括生产者和消费者

（2）队列中的消息只能被一个消费者消费

（3）消费者可以随时消费队列中的消息

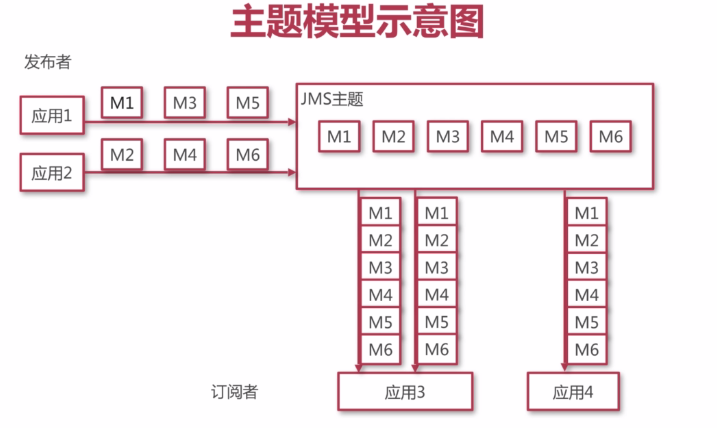


3.JMS消息模式——主题模式

（1）主题模式客户端包括发布者和订阅者

（2）主题中的消息被所有订阅者消费

（3）消费者不能消费订阅之前就发送到主题中的消息



4. JMS规范——编码接口说明

（1）ConnectionFactory：用于创建连接到消息中间件的连接工厂

（2）Connection：代表了应用程序和消息服务器之间的通信链路

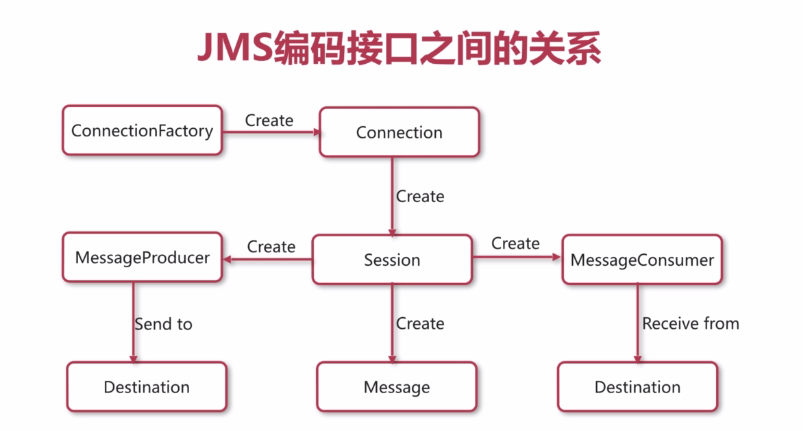
（3）Destination：指消息发布和接收的地点，包括队列或主题

（4）Session：表示一个单线程的上下文，用于发送和接收消息

（5）MessageConsumer：由会话创建，用于接收发送到的目标消息

（6）MessageProducer：由会话创建，用于发送消息到目标

（7）Message：是在消费者和生产者之间传送的对象，消息头，一组消息属性，一个消息体。



5. JMS安装

（1）在Windows平台上安装ActiveMQ

下载地址：<http://activemq.apache.org/download.html>

下载之后直接启动服务就可以用了。

（2）在Linux平台安装ActiveMQ

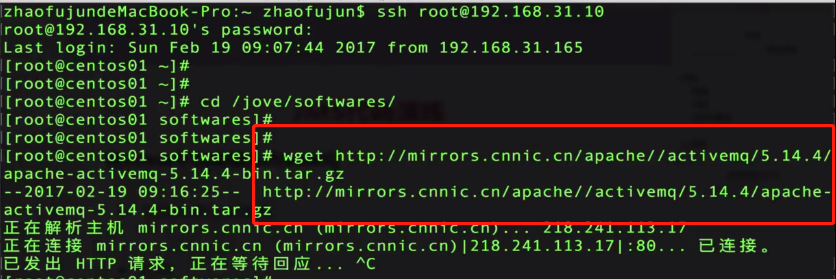
下载命令：

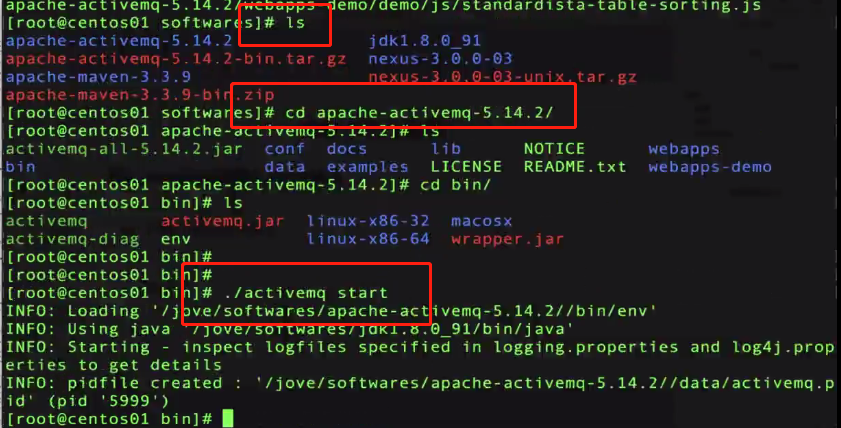
wget http://mirrors.cnnic.cn/apache//activemq/5.15.4/apache-activemq-5.15.4-bin.tar.gz

解压缩：tar –zxvf apache-activemq-5.15.4-bin.tar.gz

进入bin目录下：./activemq start 启动服务

浏览器中访问：<http://127.0.0.1:8161> 登录账号：admin/admin





1. 投入战斗
2. 使用JMS接口规范连接ActiveMQ
3. 创建生产者/消费者（队列模式）
4. 创建发布者/订阅者（主题模式）

连接工厂 提供连接池， 因为，java消息服务 每次发送消息 都会重新创建连接，会话和生产者。消耗性能。 所以 提供了连接池。 提供了 单个连接工厂 和 caching缓存连接工厂













连接工厂 提供连接池， 因为，java消息服务 每次发送消息 都会重新创建连接，会话和生产者。消耗性能。 所以 提供了连接池。 提供了 单个连接工厂 和 caching缓存连接工厂



