## <https://blog.csdn.net/wender/article/details/86317970>

## 一、消息中间件相关知识

### 1、概述

消息队列已经逐渐成为企业IT系统内部通信的核心手段。它具有低耦合、可靠投递、广播、流量控制、最终一致性等一系列功能，成为异步RPC的主要手段之一。当今市面上有很多主流的消息中间件，如老牌的ActiveMQ、RabbitMQ，炙手可热的Kafka，阿里巴巴自主开发RocketMQ等。

2 消息中间件的组成

2.1 Broker

消息服务器，作为server提供消息核心服务

2.2 Producer

消息生产者，业务的发起方，负责生产消息传输给broker，

2.3 Consumer

消息消费者，业务的处理方，负责从broker获取消息并进行业务逻辑处理

2.4 Topic

主题，发布订阅模式下的消息统一汇集地，不同生产者向topic发送消息，由MQ服务器分发到不同的订阅者，实现消息的 广播

2.5 Queue

队列，PTP模式下，特定生产者向特定queue发送消息，消费者订阅特定的queue完成指定消息的接收

2.6 Message

消息体，根据不同通信协议定义的固定格式进行编码的数据包，来封装业务数据，实现消息的传输

3 消息中间件模式分类

3.1 点对点 PTP

3.2 发布/订阅 Pub/Sub发布订阅（广播）：使用topic作为通信载体

4 消息中间件的优势

4.1 系统解耦

交互系统之间没有直接的调用关系，只是通过消息传输，故系统侵入性不强，耦合度低。

4.2 提高系统响应时间

例如原来的一套逻辑，完成支付可能涉及先修改订单状态、计算会员积分、通知物流配送几个逻辑才能完成；通过MQ架构设计，就可将紧急重要（需要立刻响应）的业务放到该调用方法中，响应要求不高的使用消息队列，放到MQ队列中，供消费者处理。

4.3 为大数据处理架构提供服务

通过消息作为整合，大数据的背景下，消息队列还与实时处理架构整合，为数据处理提供性能支持。

4.4 Java消息服务——JMS

Java消息服务（Java Message Service，JMS）应用程序接口是一个Java平台中关于面向消息中间件（MOM）的API，用于在两个应用程序之间，或分布式系统中发送消息，进行异步通信。

JMS中的P2P和Pub/Sub消息模式：点对点（point to point， queue）与发布订阅（publish/subscribe，topic）最初是由JMS定义的。这两种模式主要区别或解决的问题就是发送到队列的消息能否重复消费(多订阅)。

5 消息中间件应用场景

5.1 异步通信

5.2 解耦

5.3 冗余

5.4 扩展性

5.5 过载保护

5.6 可恢复性

5.7 顺序保证

5.8 缓冲

5.9 数据流处理

6 消息中间件常用协议

6.1 AMQP协议

6.2 MQTT协议

6.3 STOMP协议

6.4 XMPP协议

6.5 其他基于TCP/IP自定义的协议

7 常见消息中间件MQ介绍

7.1 RocketMQ

具有以下特点：

能够保证严格的消息顺序

提供针对消息的过滤功能

提供丰富的消息拉取模式

高效的订阅者水平扩展能力

实时的消息订阅机制

亿级消息堆积能力

7.2 RabbitMQ

7.3 ActiveMQ

7.4 Redis

7.5 Kafka

7.6 ZeroMQ

特性：

无锁的队列模型：对于跨线程间的交互（用户端和session）之间的数据交换通道pipe，采用无锁的队列算法CAS；在pipe的两端注册有异步事件，在读或者写消息到pipe的时，会自动触发读写事件。

批量处理的算法：对于批量的消息，进行了适应性的优化，可以批量的接收和发送消息。

多核下的线程绑定，无须CPU切换：区别于传统的多线程并发模式，信号量或者临界区，zeroMQ充分利用多核的优势，每个核绑定运行一个工作者线程，避免多线程之间的CPU切换开销。

**二、主要消息中间件的比较**

