

RocketMQ

淘宝内部的交易系统使用了淘宝自主研发的 Notify 消息中间件，使用 Mysql 作为消息存储媒介，可完全水平扩容，为了进一步降低成本，我们认为存储部分可以进一步优化，2011 年初，Linkin 开源了 Kafka 这个优秀的消息中间件，淘宝中间件团队在对 Kafka 做过充分 Review 之后，Kafka 无限消息堆积，高效的持久化速度吸引了我们，但是同时发现这个消息系统主要定位于日志传输，对于使用在淘宝交易、订单、充值等场景下还有诸多特性不满足，为此我们重新用 Java 语言编写了 RocketMQ，定位于非日志的可靠消息传输（日志场景也 OK），目前 RocketMQ 在阿里集团被广泛应用在订单，交易，充值，流计算，消息推送，日志流式处理，binglog 分发等场景。

Kafka

Kafka 是 LinkedIn 开源的分布式发布-订阅消息系统，目前归属于 Apache 定级项目。Kafka 主要特点是基于 Pull 的模式来处理消息消费，追求高吞吐量，一开始的目的就是用于日志收集和传输。0.8 版本开始支持复制，不支持事务，对消息的重复、丢失、错误没有严格要求，适合产生大量数据的互联网服务的数据收集业务。

RabbitMQ

RabbitMQ 是使用 Erlang 语言开发的开源消息队列系统，基于 AMQP 协议来实现。AMQP 的主要特征是面向消息、队列、路由（包括点对点和发布/订阅）、可靠性、安全。AMQP 协议更多用在企业系统内，对数据一致性、稳定性和可靠性要求很高的场景，对性能和吞吐量的要求还在其次。

有关测试结论

Kafka 的吞吐量高达 17.3w/s，不愧是高吞吐量消息中间件的行业老大。这主要取决于它的队列模式保证了写磁盘的过程是线性 IO。此时 broker 磁盘 IO 已达瓶颈。

RocketMQ 也表现不俗，吞吐量在 11.6w/s，磁盘 IO %util 已接近 100%。RocketMQ 的消息写入内存后即返回 ack，由单独的线程专门做刷盘的操作，所有的消息均是顺序写文件。

RabbitMQ 的吞吐量 5.95w/s，CPU 资源消耗较高。它支持 AMQP 协议，实现非常重量级，为了保证消息的可靠性在吞吐量上做了取舍。我们还做了 RabbitMQ 在消息持久化场景下的性能测试，吞吐量在 2.6w/s 左右。

在服务端处理同步发送的性能上，Kafka>RocketMQ>RabbitMQ。

对比了最简单的小消息发送场景,Kafka 暂时胜出。但是,作为经受过历次双十一洗礼的 RocketMQ,在互联网应用场景中更有它优越的一面。

功能	消息队列 RocketMQ	Apache RocketMQ(开源)	Apache Kafka	RabbitMQ
安全防护	支持	不支持	不支持	支持
主子账号支持	支持	不支持	不支持	不支持
可靠性	<ul style="list-style-type: none">- 同步刷盘- 同步双写- 超 3 份数据副本- 99.9999999%	<ul style="list-style-type: none">- 同步刷盘- 异步刷盘	异步刷盘,丢数据概率高	同步刷盘
可用性	<ul style="list-style-type: none">- 非常好, 99.95%- Always Writable	好	好	好
横向扩展能力	<ul style="list-style-type: none">- 支持平滑扩展- 支持百万级 QPS	支持	支持	<ul style="list-style-type: none">- 集群扩容依赖前端- LVS 负载均衡调度
Low Latency	支持	不支持	不支持	不支持
消费模型	Push / Pull	Push / Pull	Pull	Push / Pull
定时消息	支持(可精确到秒级)	支持(只支持 18 个固定 Level)	不支持	支持
事务消息	支持	不支持	不支持	不支持
顺序消息	支持	支持	支持	不支持
全链路消息轨迹	支持	不支持	不支持	不支持
消息堆积能力	百亿级别 不影响性能	百亿级别 影响性能	影响性能	影响性能
消息堆积查询	支持	支持	不支持	不支持
消息回溯	支持	支持	不支持	不支持
消息重试	支持	支持	不支持	支持
死信队列	支持	支持	不支持	支持
性能(常规)	非常好 百万级 QPS	非常好 十万级 QPS	非常好 百万级 QPS	一般 万级 QPS
性能(万级 Topic 场景)	非常好 百万级 QPS	非常好 十万级 QPS	低	低

性能（海量消息堆积场景）	非常好 百万级 QPS	非常好 十万级 QPS	低	低
--------------	----------------	----------------	---	---

对比

	Kafka	RocketMq	RabbitMQ
关注度	高	中	高
成熟度	成熟	比较成熟	成熟
所属社区/公司	Apache	Alibaba Apache	Mozilla Public License
社区活跃度	高	中	高
文档	多	中	多
特点	吞吐量与消息积累都很强大 Topic 太多会影响性能。	各个环节分布式扩展设计，主从 HA；支持上万个队列；多种消费模式；性能很好	由于 Erlang 语言的并发能力，性能很好
授权方式	开源	开源	开源
开发语言	scala	Java	Erlang
支持的协议	一套自行设计的基于 TCP 的二进制协议	自己定义的一套（社区提供 JMS--不成熟）	AMQP
客户端支持语言	C/C++, Python, Go, Erlang, Java 等	Java C++（不成熟）	Java、C、C++、Python、PHP、Perl、.net 等
持久化	磁盘文件	磁盘文件	内存、文件
事务	不支持	支持	不支持

集群	Zookeeper	Nameserver	
单机支持的队列	单机超过 64 个队列，性能会明显下降	单机最高支持 5W 个队列，性能没有明显变化	依赖于内存
定时消息	不支持	开源版仅支持定时 Level	不支持
顺序消费	支持顺序消费，但是一台 Broker 宕机后，顺序会乱	支持顺序消费，在顺序消费场景下，消费失败时消费队列将会暂停	支持顺序消费，但是一台 Broker 宕机后，顺序会乱
负载均衡	支持	支持	支持
管理界面	无	无社区有 web console 实现	好
部署依赖	zookeeper	Nameserver	Erlang 环境
消费方式		保证严格的消费顺序	
总结：优点	1、高吞吐、低延迟、高性能 2、提供多种客户端语言 3、生态完善，大数据处理方面的必备工具	模型简单，接口易用。在阿里大规模应用。目前支付宝中的余额宝等新兴产品均使用 rocketmq。集群规模大概在 50 台左右，单日处理消息上百亿；性能非常好，可以大量堆积消息在 broker 中；支持多种消费，包括集群消费、广播消费等。开发度较活跃，版本更新很快。	由于 erlang 语言的特性，mq 性能较好；管理界面较丰富，在互联网公司也有较大规模的应用；支持 amqp 协议，有多种语言且支持 amqp 的客户端可用
总结：缺点	消费者集群数受到分区数的限制 单机 Topic 过多，性能会明显下降 不支持事务。 容易丢数据。	使用者较少，生态不够完善，消息堆积与吞吐量上与 kafka 还是有差距。 客户端支持 java	Erlang 语言难度较大，集群不支持动态扩展。 不支持事务，消息吞吐能力有限 消息堆积时，性能会明显降低。