



《手写OS操作系统》小班二期招生，全程直播授课，大牛带你掌握硬核技术！

点此查看

工具

式单元测试

模式 - 函数

加载

函数开发、

插件

包设计

生成工具开

码结构

从所有教程的词条中查询...

首页 > 慕课教程 > Go工程师体系课全新版 > 10. mq技术选型



bobby · 更新于 2022-11-16

← 上一节 9. mq (messag... 11. rocketmq安... 下一节 →

常见的mq的区别

特性	ActiveMQ	RabbitMQ	RocketMQ	Kafka
单机吞吐量	万级，吞吐量比RocketMQ和Kafka要低了一个数量级	万级，吞吐量比RocketMQ和Kafka要低了一个数量级	10万级，RocketMQ也是可以支撑高吞吐的一种MQ	10万级别，这是kafka最大的优点，就是吞吐量高。一般配合大数据类的系统来进行实时数据计算、日志采集等场景
topic数量对吞吐量的影响			topic可以达到几百，几千个的级别，吞吐量会有较小幅度的下降这是RocketMQ的一大优势，在同等机器下，可	topic可以达到几百，几千个的级别，吞吐量会有较小幅度的下降这是RocketMQ的一大优势，在同等机器下，可
时效性	ms级	微秒级，这是rabbitmq的一大特点，延迟是最低的	ms级	延迟在ms级以内
可用性	高，基于主从架构实现高可用性	高，基于主从架构实现高可用性	非常高，分布式架构	非常高，kafka是分布式的，一个数据多个副本，少数机器宕机，不会丢失数据，不会导致不可用
消息可靠性	有较低的概率丢失数据		经过参数优化配置，可以做到0丢失	经过参数优化配置，消息可以做到0丢失
功能支持	MQ领域的功能极其完备	基于erlang开发，所以并发能力很强，性能极好，延时很低	MQ功能较为完善，还是分布式的，扩展性好	功能较为简单，主要支持简单的MQ功能，在大数据领域的实时计算以及日志采集被大规模使用，是事实上的标准
优劣势总结	非常成熟，功能强大，在业内大量的公司以及项目中都有应用偶尔会有较低概率丢失消息而且现在社区以及国内应用都越来越少，官方社区现在对ActiveMQ 5.x维护越来越少，几个月才发布一个版本而且确实主要是基于解耦和异步来用的，较少在大规模吞吐的场景中使用	erlang语言开发，性能极好，延时很低；吞吐量到万级，MQ功能比较完备而且开源提供的管理界面非常棒，用起来很好用社区相对比较活的。RabbitMQ吞吐量会低一些，这是因为他做的实现机制比较重。erlang开发很难去看懂源码，你公司对这个东西的掌控很弱，基本职能依赖于开源社区的快速维护和修复bug。	接口简单易用，而且毕竟在阿里大规模应用过，可以做到大规模吞吐，性能也非常好，分布式扩展也很方便，社区维护还可以，可靠性和可用性是ok的，还可以支撑大规模的topic数量。阿里出品都是java系的，我们可以自己阅读源码。	kafka的特点其实很明显，就是仅提供较少的核心功能，但是提供超高的吞吐量，ms级的延迟，极高的可用性以及可靠性。而且分布式可以任意扩展同时kafka最好是支撑较少的topic数量即可，保证其超高吞吐量而且kafka唯一的缺点是有可能消息重复

结论:

(1)中小型软件公司，建议选RabbitMQ.一方面，erlang语言天生具备高并发的特性，而且他的管理界面用起来十分方便。正所谓，成也萧何，败也萧何！他的弊端也在这里，虽然RabbitMQ是开源的，然而国内有几个能定制化开发erlang的程序员呢？所幸，RabbitMQ的社区十分活跃，可以解决开发过程中遇到的bug，这点对于中小型公司来说十分重要。不考虑rocketmq和kafka的原因是，一方面中小型软件公司不如互联网公司，数据量没那么小，选消息中间件，应首选功能比较完备的，所以kafka排除。但是rocketmq已经交给apache管理，所以rocketmq的未来发展趋势看好。

(2)大型软件公司，根据具体使用在rocketMq和kafka之间二选一。一方面，大型软件公司，具备足够的资金搭建分布式环境，也具备足够大的数据量。针对rocketMQ,大型软件公司也可以抽出人手对rocketMQ进行定制化开发，毕竟国内有能力改JAVA源码的人，还是相当多的。至于kafka，根据业务场景选择，如果只有高吞吐功能，肯定是首选kafka了。具体该选哪个，看使用场景。

意见反馈

收藏教程

标记书签

课程中选择rocketmq，基于两点考虑：

1. 延迟消息简单高效

1. 死信队列

2. 完善的事务消息功能

9. mq（message queue）的使用场景 ◀ 上一节 下一节 ▶ 11. rocketmq安装和配置

✎ 我要提出意见反馈

索引目录

