

# 消息队列（泛指RocketMQ）考察点

## 入门 RocketMQ

### 1. 文章学习

- [拿个offer：从零到一学习RocketMQ](#)
- [RocketMQ官网：从零到一学习RocketMQ 4.x](#)
- [公众号@君哥聊技术：【修正版】7张图带你轻松入门RocketMQ](#)
- [博客园：高并发异步解耦利器：RocketMQ究竟强在哪里？](#)

### 2. 视频学习

TODO

## 面试高频考点

### 1. RocketMQ为什么这么快？

问题回答核心：

- 可横向扩展的分布式架构（Broker）。
- Netty 网络通信模型。
- 支持异步刷盘。
- 页缓存 PageCache。
- 零拷贝。
- 顺序写入 CommitLog 磁盘文件。

### 2. 为什么使用RocketMQ而不用Kafka或者RabbitMQ？

问题回答核心：

RabbitMQ 从性能上不及 RocketMQ、Kafka、功能上不及 RocketMQ，国内目前应用场景没那么大众，所以不参与比对。

- 功能方面：
  - Kafka 不支持广播消息。
  - Kafka 不支持延时消息。
  - Kafka 不支持事务消息。

- Kafka 不支持消息轨迹。
- Kafka 在处理顺序消息时，如果一台 Broker 宕机后，会产生消息乱序。而 RocketMQ 如果消息消费失败，会暂停消费。
- Kafka 不支持基于 Broker 的 Tag 消息过滤（这个我存疑，说不定后面版本会出现），RocketMQ 支持。
- 存疑点：网上说当 Kafka 分区多的时候，性能会下降，RocketMQ 不会。
- 语言方法：
  - Kafka 使用 Java 和 Scala，针对一些框架原理查看以及二次开发不便利。
- 使用场景：
  - Kafka 侧重于通过流处理引擎实现实时数据流处理，在大数据流处理和实时数据分析方面使用较多。
  - RocketMQ 的设计更注重实现高可用和多功能的消息服务，在国内较多公司应用较为广泛。

综合比对，Kafka 的优势在于性能高，而 RocketMQ 的功能和业务场景更贴合国内公司，所以使用也较多。

推荐学习文章：

- [51CTO@IT利刃出鞘：RabbitMQ，RocketMQ，Kafka--区别/对比/选型](#)

### 3. RocketMQ消息如何防止丢失？（高可用）

问题回答核心：

- Producer：同步发送和异步发送的区别，捎带脚说说生产者发送的重试机制。
- Broker：刷盘策略和多副本（主从机制）。
- Consumer：ACK 消息确认以及重试机制。
- 极端情况：如果在发送阶段 Broker 集群挂掉了，那么我们可以提供两种思路。
  - 中间件降级：比如 RocketMQ 发送失败，那我们可以通过 Redis Stream 充当消息队列，进行发送。RocketMQ 好过来后，再切换回来。
  - 消息表：发送失败后将消息保存到 MySQL 消息发送失败表中，等 RocketMQ 好过来后再进行重发消费。

推荐学习文章：

- [公众号@君哥聊技术：4 张图，9 个维度告诉你怎么做能确保 RocketMQ 不丢失消息](#)

### 4. 如何解决消息堆积？

问题回答核心：

消息堆积无外乎因为两点原因：消息发的太快&消息消费太慢，大多数情况下根本原因是因为后者。

1) 消息消费慢的话，原因有两点：

- 应用节点消费太少，也就是消费力不足，或者消费的流程长。
- 涉及到三方中间件变慢：比如说 MySQL 正在被高并发访问或者说执行大量 Count 操作，导致性能变慢，这种应该属于通用问题，发现后需要权衡扩容数据库配置或者优化组件变慢的原因。比如慢 SQL 还是其他。

针对节点太少这种的话，就应该适当进行节点扩容，或者调整拉消息底层线程池的线程大小。节点扩容有上限，RocketMQ 4.x 的版本节点数最多扩充到 Topic 下队列最大数量，因为一个队列仅允许单个消费者订阅。但是要考虑到后续如果业务中有 DB 操作，DB 是否支持这么高的读写。操作流程长容易理解，就是一个消费者要调用 N 个请求或者流程，自然快不起来，解决思路同上。

另外，如果说应用节点数大于 Topic 下队列数量，增加节点就没有用了。我们可以创建一个临时 Topic，然后修改项目里原来消费者的代码，将原来的逻辑修改为向临时 Topic 投递，在新临时 Topic 的消费者里执行真实的消费逻辑，相当于中转了下。

或者改造生产者，将生产者进行判断，均匀发送到两个队列中。

2) 消息发的太快：可以通过限流方式进行流控，不过这种会影响用户正常使用，一般我们都是消费者做调整。

推荐学习文章：

- [公众号@勇哥Java实战：如何应对RocketMQ消息堆积？](#)
- [公众号@君哥聊技术：RocketMQ消息积压了，增加消费者有用吗？](#)

## 5. 如何防止重复消费？

问题回答核心：

1) 首先说下什么情况下会重复消费，参考下边学习文章。

2) 其次说下如何保障消息队列不被重复消费，比如通过 Redis 和 MySQL 做消息表。另外，这个点会有很多细节，刚开始不要把细节说太多，面试官容易绕进去。保障一个大体逻辑，然后很多点当做问题一嘴带过即可。然后面试官感兴趣，自然会问你相关的细节。

推荐学习文章：

- [博客园@轻风博客：RocketMQ会重复消费吗？](#)
- [图4. 如何保证消息队列不被重复消费](#)

## 6. RocketMQ消费消息是拉还是推？

问题回答核心：

推和拉都有，不过根本上，消费者去 Broker 上获取消息，底层就是拉模型，Broker 不会向你推送消息。拉消息会涉及到一个长轮询机制，是个不错的亮点，具体解答查看下述文章。

推荐学习文章：

- [CSDN@一颗向上的草莓：RocketMQ消息消费方式-推拉模式](#)
- [公众号@勇哥Java实战：万字长文讲透 RocketMQ 的消费逻辑](#)

## 7. RocketMQ如何实现延时消息？

问题回答核心：

4.x 和 5.x 是两个完全不同的实现，4.x 仅支持固定的延迟级别，5.x 支持了任意时间延迟，都可以结合说说。

推荐学习文章：

- [公众号@君哥聊技术：5张图带你理解RocketMQ延时消息机制](#)
- [公众号@君哥聊技术：RocketMQ基于时间轮算法实现了定时消息](#)