面试真题

【美团】用RocketMQ主要是用来解决什么问题的？

RocketMQ 是一种分布式消息中间件，主要功能是实现高性能、高吞吐量的消息传递和流处理。

主要应用场景：

1）异步通信：当系统中有大量的请求需要异步处理时，使用 RocketMQ 可以解耦系统，提升性能。例如，用户注册后发送欢迎邮件等操作可以通过消息队列异步处理。

2）服务解耦：通过使用消息队列，各个服务之间可以通过消息进行通信，降低耦合，提高系统的可扩展性和稳定性。

3）削峰填谷：在高并发系统中，短时间内的高并发请求可能导致系统崩溃，通过消息队列可以进行流量削峰填谷，保护系统。

4）日志传输：分布式系统中各个节点产生的日志，可以通过消息队列进行集中采集和传输。

5）分布式事务：RocketMQ 提供了分布式事务的支持，在分布式系统中可以保证事务的一致性。

【美团】RocketMQ什么情况下会出现重复消费的问题？

重复消费问题出现的情况：

1. **没有及时确认**：默认情况下，RocketMQ使用"消费确认"机制，消费者需要手动确认消息是否处理成功。如果消费者在一定时间内未成功消费消息，RocketMQ会认为该消息未成功处理并重新投递。若消息处理成功但消费者没有及时确认，也会导致该消息被重新消费。
2. **没能成功确认**：在消费者与RocketMQ之间发生网络故障的情况下，消息可能未能成功确认，RocketMQ会再次投递该消息，造成重复消费。
3. **消费失败**：如果消费者出现异常导致消费失败，RocketMQ会尝试将消息重新投递。消费者处理失败的消息会被重复消费，直到达到最大重试次数。
4. **消费进度丢失**：在某些情况下，如果消费者的消费进度（偏移量）丢失或错误，RocketMQ无法正确判断已消费的消息，可能导致重复消费。

【美团】RocketMQ如何避免重复消费？

在 RocketMQ 中，通过以下几种方法可以确保消息不会重复消费：

1）幂等性：确保消费者的处理逻辑是幂等的，即对于相同的消息重复处理多次，结果应保持一致。

2）消息状态记录：在消费前记录消息状态，如果记录存在则不重复处理。常见的实现方式是使用数据库记录消费的消息ID。

3）消息重试机制：配置合理的消息重试机制，限制重试次数，避免消息反复被消费。

【美团】Kafka和RocketMQ有什么区别，实现细节上有了解吗

RocketMQ 和 Kafka 都是流行的分布式消息队列系统。

1. 数据模型：
   1. Kafka 主要是日志模型，适用于高吞吐量的数据流处理。
   2. RocketMQ 的消息模型更为多样，包括点对点、发布订阅和顺序消息。
2. 存储机制：
   1. Kafka 使用的是磁盘顺序写，主要依赖文件系统的缓存机制；
   2. RocketMQ 存储有自己的机制，更像数据库，有索引能快速定位消息。
3. 高可用机制：
   1. Kafka 有 ISR（in-sync replicas）机制来保障数据的高可用性
   2. RocketMQ 有强一致性协议及消息重发机制。
4. 消费模式和顺序：
   1. Kafka 基于分区偏移量来进行消费，并且有严格的顺序保证。
   2. RocketMQ 数据消费模式灵活，支持普通消费与顺序消费。
5. 适用场景：
   1. Kafka适用于高吞吐量的日志收集系统
   2. RocketMQ适合低延迟、高可靠的电商支付、金融交易等。

面向简历

Nginx

Docker及其相关组件

容器化技术

Devops