

你们项目中哪些地方有使用到 MQ

1. 使用 mq 异步发送优惠券；
2. 使用 mq 异步发送短信；
3. 使用 mq 异步扣库存

总之将执行比较耗时的代码操作，交给 mq 异步实现接口。

为什么需要使用 MQ

- 1.异步处理（多线程和 MQ）
- 2.实现解耦
- 3.流量削峰（MQ 可以实现抗高并发）

MQ 与多线程实现异步的区别

- 1.多线程方式实现异步可能会消耗到我们的 cpu 资源，可能会影响到我们业务线程执行会发生 cpu 竞争的问题；
- 2.MQ 方式实现异步是完全解耦，适合于大型互联网项目；
- 3.小的项目可以使用多线程实现异步，大项目建议使用 MQ 实现异步；

MQ 如何避免消息堆积的问题

1.产生背景：

生产者投递消息的速率与我们消费者消费的速率完全不匹配。

2.生产者投递消息的速率>消费者消费的速率

导致我们消息会堆积在我们 mq 服务器端中，没有及时的被消费者消费
所以就会产生消息堆积的问题

3.注意的是：rabbitmq 消费者我们的消息消费如果成功的话 消息会被立即删除。
kafka 或者 rocketmq 消息消费如果成功的话，消息是不会立即被删除。

4.解决办法：

- A.提高消费者消费的速率；（对我们的消费者实现集群）
- B.消费者应该批量形式获取消息 减少网络传输的次数；

MQ 宕机了消息是否会丢失呢

不会，因为我们消息会持久化在我们硬盘中；

MQ 如何保证消息不丢失

1. MQ 服务器端

消息持久化到硬盘

2. 生产者

消息确认机制

必须确认消息成功刷盘到硬盘中，才能够人为消息投递成功。

3. 消费者

必须确认消息消费成功

rabbitmq 中：才会将该消息删除。

rocketmq 或者 kafka 中：才会提交 offset

生产者投递消息，mq 宕机了如何处理

1.生产者投递消息会将 msg 消息内容记录下来，后期如果发生生产者投递消息失败；

2.可以根据该日志记录实现补偿机制；

3.补偿机制（获取到该 msg 日志消息内容实现重试）

MQ 如何保证消息顺序一致性问题

1. 大多数的项目是不需要保证 mq 消息顺序一致性的问题，只有在一些特定的场景可能会需要，比如 MySQL 与 Redis 实现异步同步数据；

2. 所有消息需要投递到同一个 mq 服务器，同一个分区模型中存放，最终被同一个消费者消费，核心原理：设定相同的消息 key，根据相同的消息 key 计算 hash 存放在同一个分区中。

3.如果保证了消息顺序一致性有可能降低我们消费者消费的速率。

MQ 如何保证消息幂等问题

1. 消费者获取消息，如果消费消息失败，mq 服务器则会间隔的形式 实现重试策略；
2. 重试过程中，需要保证业务幂等性问题，保证业务不能够重复执行。
3. 我们可以通过全局的消息 id，提前查询如果该业务逻辑已经执行过，则不会重复执行。
4. 我们也需要在数据库的 db 层面需要保证幂等性问题，唯一主键约束、乐观锁等。

MQ 与 Redis 如何保证数据一致性问题

- 方案 1: 直接删除 Redis 缓存；
方案 2: 基于 MQ 异步同步更新
方案 3: 基于 canal 订阅 binlog 同步