

MQ

优点

- 异步
- 解耦
- 削峰

缺点

- 系统可用性降低
 - 保证高可用
 - RabbitMq
 - 普通集群模式（无高可用）
 - 优点：提高吞吐量
 - 缺点：可用性无保障，queue所在节点宕机数据就丢失；集群内部可能产生大量的数据传输，有数据拉取开销
 - 镜像集群模式
 - 优点：完整数据存在多个实例中
 - 缺点：性能开销较大，消息需要同步到所有机器；无法线性扩展
 - Kafka
 - replica副本机制
 - 如何处理重复消费
 - 若是写库则可通过判断主键是否存在，不存在则插入，存在则更新。也可以增加唯一键约束
 - 生产者发送消息前记录消息id到redis，消费者消费时去redis查看是否有消费过，没有消费则等消费者消费完成后再从Redis中移除该ID
 - 系统复杂度提高
 - 如何处理消息丢失
 - RabbitMq
 - 1. 生产者弄丢了数据
 - 1. 使用事务：会降低吞吐量，因为太耗性能
 - 2. 使用Confirm机制
 - 2. RabbitMq弄丢了数据：开启持久化
 - 3. 消费端弄丢了数据：关闭自动ACK
 - kafka
 - 如何保证消息传递的顺序性
 - RabbitMq
 - 拆分多个 queue，每个 queue 一个 consumer，就是多一些 queue 而已，确实是麻烦点；或者就一个 queue 但是对应一个 consumer，然后这个 consumer 内部用内存队列做排队，然后分发给底层不同的 worker 来处理。
 - Kafka
 - 一个 topic，一个 partition，一个 consumer，内部单线程消费，单线程吞吐量太低，一般不会用这个
 - 写 N 个内存 queue，具有相同业务规则的数据都到同一个内存 queue；然后对于 N 个线程，每个线程分别消费一个内存 queue 即可，这样就能保证顺序性
 - 一致性问题