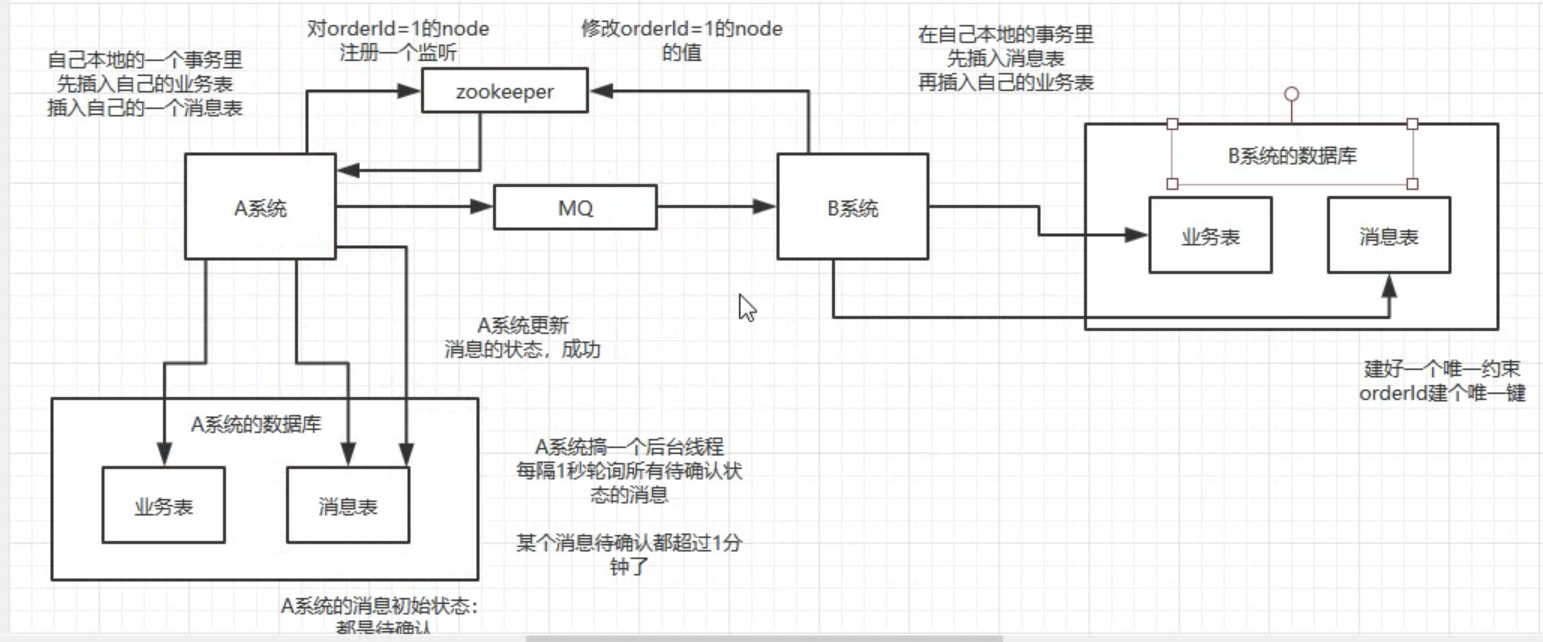
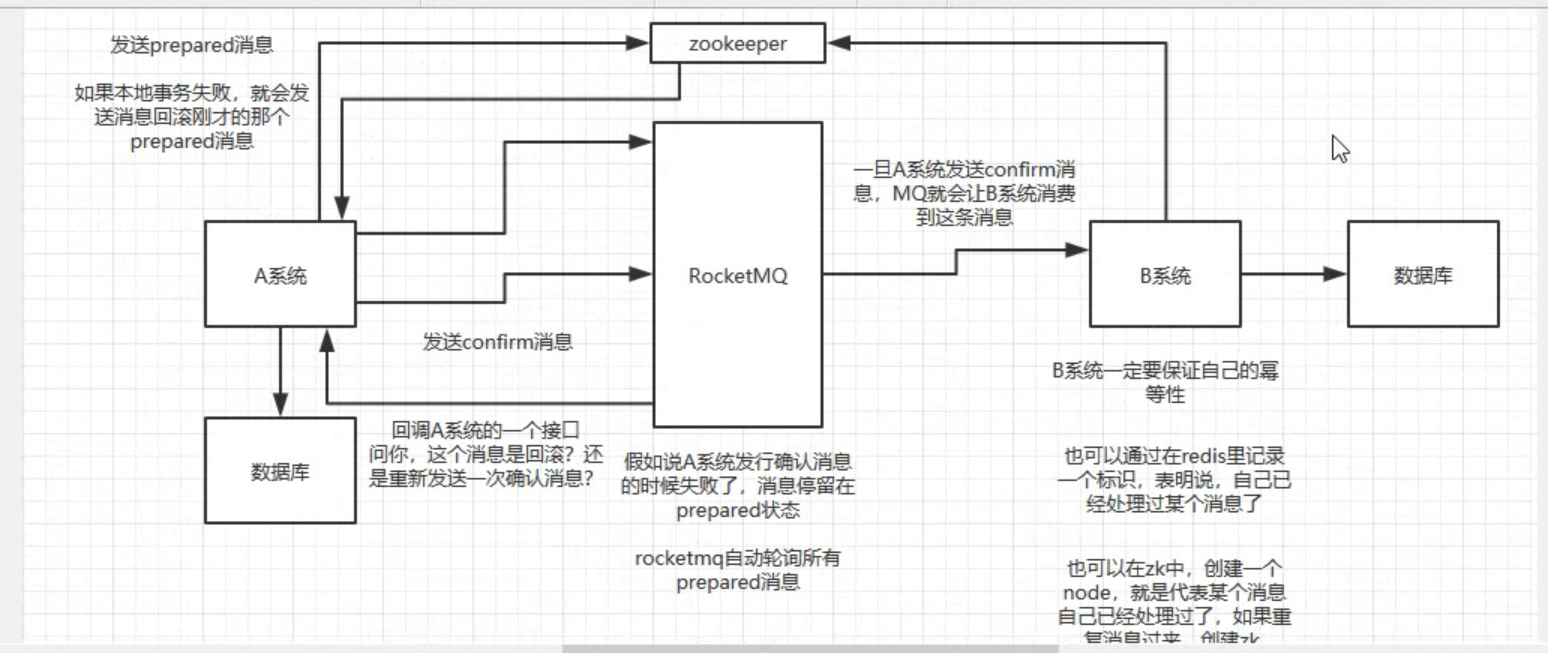
分布式事务：

1. 两阶段提交(可以解决一个系统操作多个数据库，但是通常情况不允许一个服务直连别的数据库)
2. TCC方案 (try confirm cancel)
   1. 实现复杂 但是能100%保证事务一致性
   2. 适用于核心场景或者是与资金相关的场景
3. 本地消息表



依赖于数据库的消息表 不适合高并发场景

1. 可靠消息最终一致方案(最优)



1. 分布式CAP
   1. 一致性 C
      1. 分布式系统中所有节点数据保持一致
   2. 可用性 A
      1. 任何一个请求，都要得到正常数据的响应
   3. 分区容错性 P
      1. 部分故障不会影响整体服务
2. 一致性和可用性，为什么不可能同时成立？答案很简单，因为可能通信失败（即出现分区容错）。

如果保证 G2 的一致性，那么 G1 必须在写操作时，锁定 G2 的读操作和写操作。只有数据同步后，才能重新开放读写。锁定期间，G2 不能读写，没有可用性

如果保证 G2 的可用性，那么势必不能锁定 G2，所以一致性不成立。

1. 综上所述，G2 无法同时做到一致性和可用性。系统设计时只能选择一个目标。如果追求一致性，那么无法保证所有节点的可用性；如果追求所有节点的可用性，那就没法做到一致性