无论水平还是垂直拆分，都有很多共通的缺点：

**引入分布式事务**

1. 如何处理两个系统之间的数据交互（解决方案：针对不同的场景案例，具体分析解决）

如何处理事务：例如业务逻辑复杂时，通过SOA做通用服务，再service层上做多个切面，配置多个事务

1. 例如数据量大且分析逻辑复杂时，通过使用缓冲库，缓冲表等数据库设计，最简单的业务场景就是一个系统往数据库里面写，另一个数据库进行读操作
2. 分布式锁的概念：

本质上就是一个线程在操作的时候，比如说有一个表同一时间点只能一个线程进行操作（当然借助select … for update 操作更好）

1. 例如要求实时性非常高且数据信息，业务逻辑简单单一，使用第三方数据通信组件，消息队列做回调服务，或者使用zookeeper建立分布式锁进行数据同步。或者使用netty进行通信，类似的还有webService,Restful等直接请求

**跨节点join的问题，跨节点合并，排序，分页等处理数据的问题**

1. 通用的解决方案是把数据组织好后，放到缓存里面去，定时或实时的进行同步
2. 如果要求实时性不是特别高，那么也可以使用中间库的手段去解决

多数据源管理问题（解决方案）