

09 分布式并发管理

- 今天的假设
 - 忽略失败，可以从log恢复失败
 - 关注并发
- 数据库事务
 - Atomicity，事务中的sql要么都执行，要么都不执行
 - Consistency，事务修改的所有相关数据保持一致
 - Isolation，两个事务不能同时执行，特殊：只读事务可并行
 - Durability，写成功
- 加锁实现事务
 - 其中注意死锁
- 将事务分解为两个阶段
 - 准备
 - 提交或放弃
- 分布式两阶段提交
 - 客户端发起请求
 - 协调者询问所有参与者，是否可以提交
 - 参与者都回复结果
 - 此时参与者要锁数据
 - 如果所有参与者都说可以，协调者发送提交；否则发送放弃
 - 如果有参与者写失败怎么办？
 - 问题：使用超时处理失败，性能会很差
 - 之后分析各种失败情况
 - 总结使用gossip protocol解决问题
 - 假设没有拜占庭失败模型
- gossip protocol
 - 询问其他参与者
 - 如果有参与者没回复，或者否定，可以安全放弃
 - 如果没有收到参与者的回复，或者其他人也说了OK，则等待
- 3 phase commit protocol
 - 1：协调者询问是否可以提交，参与者回复
 - 2：协调者发送准备提交，参与者确认
 - 3：协调者发送提交，参与者回复提交结果
 - 动机
 - 2PC在commit阶段，协调者和一个参与者失败会造成阻塞
 - 不好
 - 消息多，延时大