

04 时钟同步

- 时钟同步的需求
 - 不同计算机时钟走的快慢
 - UTC时间标准
- 时钟同步技术
 - Cristian算法
 - $t+d/2$, 减少一些延时错误
 - Berkeley算法
 - 用于集群时间同步
 - master询问所有机器时间, 再告诉各个机器如何调整时间
 - Network Time Protocol
 - 三级层次划分
 - 一级连通原子钟
 - 二级从一级或二级获取时间
 - 三级从任意服务器获取时间
 - 同步算法和Cristian类似
 - 精度: 局部1ms, 全球 10ms
 - 改变时间
 - 不能直接改时间
 - 而是改时钟频率
 - 渐进, 防止本地时钟不一致
- Lamport Clocks
 - 逻辑时间
 - 不考虑时间粒度
 - 关心事件的先后顺序, 因果关系
 - 给每个事件一个编号
 - 每个事件的编号, 都大于当前进程之前事件的编号
 - 同时大于所依赖的其他进程的事件编号
 - 优点
 - 简单, 少了很多同步
 - 缺点
 - 不同进程间的事件编号, 不能说明他们的先后关系
- 向量时钟
 - 每个事件的时间是一个向量
 - 向量的长度和总有进程数
 - 每个进程上的事件, 进程号递增
 - 当进程间发生依赖关系, 才会更新所依赖进程的事件编号
 - 优点

- 任何两个事件的编号，都基本能够说明事件的先后关系
- 缺点
 - 进程数太大后，不容易存储
 - 难以动态扩展