

2024回忆版

2025年1月16日 11:32

Recorded by Tang Hoilap

1. 选择题：10*2points

github的一致性模型

2. 简答题

- 可拓展性的含义，如何实现可拓展性
- bully算法流程
- 可用性和可靠性的含义，为什么说可用性高的系统可靠性不一定高
- Ricart & Agrawala互斥算法流程

3. 分析题

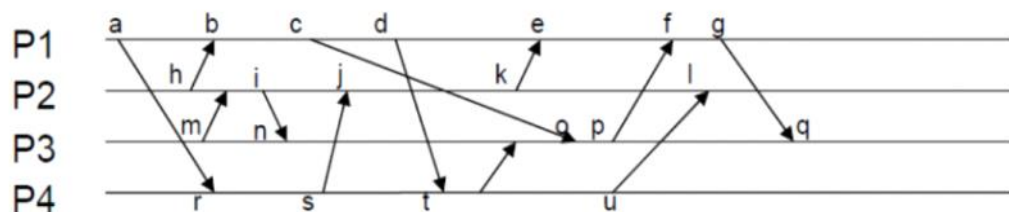
a. 时钟同步计算，类似题型如下：

在时刻 10:27:440 (小时、分钟、1/100秒)，服务器 B 从时间服务器 A 请求返回时间戳。在时刻 10:27:710，服务器 B 收到来自服务器 A 发回的时间戳 10:27:375。请解答以下几个问题：//见考点

- 找出服务 B 的时钟相对于时间服务器 A 的时钟之间的偏移量（假定在时间服务器上没有处理时间）。
- B 的时钟走的太快还是太慢了？快或慢了几秒钟？
- 如果没有时间服务器 A，服务器 B 如何调整时间？（假定服务器 B 在一个集群环境中）

Berkeley算法（在一个集群中）：时间服务器（时间守护程序）周期性地查询所有服务器，计算时间均值，再告诉所有服务器各自的调整值

b. 逻辑时钟，向量时钟的计算，题型如下：



（图非真实考试时的图，只是一个例子，摘自CPF班作业）

- 给出一幅图，写出happen before 某一个事件的所有事件
 - 给一幅图标出逻辑时钟，标出向量时钟
 - 假如对每个进程每个信息发送就会打一个检查点，判断图中一条虚线是否可以作为恢复线
- ### c. MapReduce
- 基于消息通信的一般高性能计算系统HPC使用的容错技术
 - mapreduce的特点，和一般HPC的区别
 - map和reduce的流程
 - 如果map阶段的任务比较复杂，应该怎么办？
- ### d. 一致性模型
- 几种一致性模型的强度排序（单调读，顺序一致性，因果一致性）
 - 判断一个序列是否满足因果一致性

- iii. 判断一个序列是否满足单调读, 如果不满足应该怎么修改
- e. 描述两阶段提交协议, coordinator和participant失效后如何恢复
- f. Chord
 - i. id怎么算的?
 - ii. 画指状表
 - iii. 插入某个节点时, 查找最终节点的流程
 - iv. chord的实例有哪些