

分布式系统-第六讲-协同与同步 课后作业

作业内容:

1. 从分布式系统的角度, 如果事件 a 和 b 具有 happensbefore 的关系, $a \rightarrow b$, 我们说 b 对 a 有因果依赖。那么, 从现实世界的角度, 事件 b 的发生一定是与 a 有关系吗? 为什么?

Answer:

从现实世界的角度, 事件 b 的发生不一定是与 a 有关系。

就像老师上课举得例子, 学生早上吃了一个鸡蛋, 然后考试考了 100 分, 吃鸡蛋和考试得 100 分在现实生活中, 两者之间没有因果依赖关系, 有的仅仅只是事件发生的先后有关系。

而在分布式系统中有逻辑时钟, 逻辑时钟中有 happen-before 关系也称因果关系:

① P1: 如果 a 和 b 是同一个进程中的两个事件, 并且 a 在 b 之前到达, 则有: $a \rightarrow b$

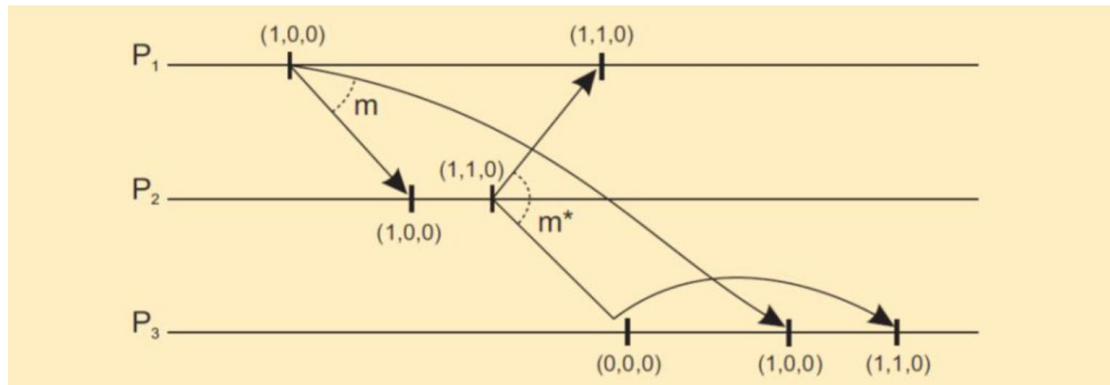
② P2: 如果 a 是消息的发送者, b 是消息的接收者, 则 $a \rightarrow b$

传递性: 如果 $a \rightarrow b$ 并且 $b \rightarrow c$ 则 $a \rightarrow c$ 本质

所有的进程并不一定在时间上达成一致, 而只需要在时间发生顺序上达成一致, 也就是需要排序

从上述定义可以看出分布式系统中的因果关系和现实世界不同, 分布式系统中的因果关系是为了确保分布式系统中时间的发生顺序, 两者是不同的概念。

2. 在强制因果有序多播的例子中, 如果发送、接收消息各作为一个事件增加时钟计数, 如何修改算法中消息交付操作才能满足要求?



Answer:

修改操作:

1、时钟仅当发送或“接受”消息时才调整 VC_j ;

2、 P_j 推迟“接受”消息 m 直到:

① $ts(m)[i] = VC_j[i]$ (m 是 P_j 应该从 P_i 接收的消息)

② $ts(m)[k] \leq VC_j[k]$, $k \neq i$ (P_j 已经接受所有 P_i 在发送 m 时已经接受的消息)

3. 基于环的选举算法中，如果两个 Election 消息同时在循环时，可以杀掉其中一个。设计一个机制实现这个功能。

Answer:

设计机制:

在基于环的选举算法中，如果两个 Election 消息同时在环中传递，可能会发生冲突。为了解决这个问题，节点可以使用两个 Election 编号来比较冲突的消息，并保留具有较大标识符的消息。选定保留的消息继续在环中传递，而被丢弃的消息则停止传递。