



# Lec-03 Homework Questions

1. 请分析一下进程虚拟机，如JVM，与操作系统虚拟机，包括原生虚拟机和主机虚拟机，各有什么特点？

答案要点：

进程虚拟机是在用户空间的虚拟化，效率高，但是隔离性差。

操作系统虚拟机，是在操作系统层的虚拟化，不同的虚拟机有自己的完整的操作系统，隔离性好，支持异构，但是资源效率低。

2. 请分析讨论：基于目录服务器和基于超级服务器的两种服务器发现方法各有什么优缺点？

答案要点：

目录服务器：逻辑简单，但是需要运行大量服务器进程，可能会浪费资源。

超级服务器：超级服务器的逻辑相对复杂：对请求进行基本解析处理，而且需要超级服务器派生服务器进程；但是服务器动态派生，资源效率高。



# Lec-04 Homework Questions

## 1.请分析讨论RPC与一般的消息通信的关系、异同。

答案要点：

一般的消息通信提供的是消息的收发接口，用于对消息进行操作，通用性好。

RPC是在消息收发的基础上，实现了函数调用的语义，实现了消息操作的对上透明，方便确实有调用关系的进程间进行服务的访问。

## 2. Gossip的多播与基于Overlay的多播各适于什么样的场景？请举例说明。

答案要点：

Gossip多播操作简单、通信开销低，可扩展性好，适用于系统规模大，但是对消息交付完成度和传播时间要求不高的场景。

基于Overlay的多播，通信效率高，消息交付有很好保证，但是overlay的维护本身需要成本开销，适于规模不是很大、节点稳定的场景。

# Lec-06 课后作业

1. 从分布式系统的角度，如果事件a和b具有happens-before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？

答案要点：

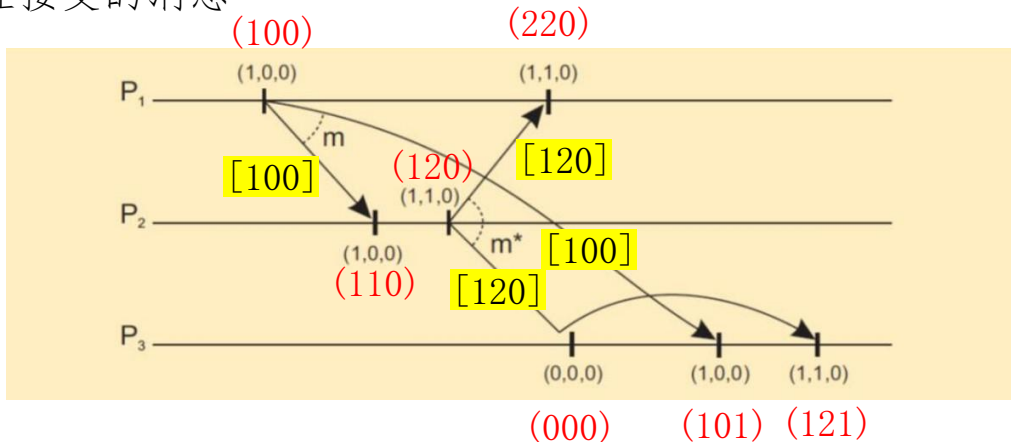
$a \rightarrow b$ ，本质上是说明b“可能”依赖于a的发生，是基于事件发生的顺序和位置而进行的一个“认定”。并不能说b事实上一定是依赖于a事件，因为现实/事实上a和b的关系取决于具体的程序或者操作的逻辑关系，这个是无法在一般化的因果关系中定义的。

# Lec-06 课后作业

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？

答案要点：P<sub>j</sub> 推迟“接受”消息m直到：

- $ts(m)[i] = VC_j[i] + 1$ ，或者：  $\{ts(m)[i] = VC_j[i] + 2, \text{ 且存在 } k, ts(m)[i] = ts(m)[k] + 1\}$
- $ts(m)[k] \leq VC_j[k], k \neq i$  // P<sub>j</sub>已经接受所有P<sub>i</sub>在发送m时已经接受的消息





# Lec-06 课后作业

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。

答案要点：

一个方案：节点设置一个election状态标志，收到election消息则将标志置为真。此状态下，如果两个election消息访问了同一个节点，则可以将第二个election消息丢弃。选举出新leader后，收到宣告消息，则将election标志复位。



# Lec-07 Homework Questions

1. 请分析讨论，与sequential consistency相比，eventual consistency的优势和价值，并通过例子进行说明。

答案要点：

Eventual consistency对一致性要求低，可以通过比较简单的机制实现，系统开销低，扩展性高，适于大型的互联网应用，譬如社交网络等。



# Lec-07 Homework Questions

2. 下面Causal consistency的操作例子，最后的一个读操作应该返回什么结果？

P1:	W(x)a		
P2:	R(x)a	W(y)b	
P3:		R(y)b	R(x)?
P4:		R(x)a	R(y)?

答案要点：

$P_3$ 已经执行操作 $R(y)b$ ，根据因果关系可知 $P_3$ 已经执行过 $W(x)a$ ，所以读操作应该返回 $R(x)a$ 。

$P_4$ 执行了操作 $R(x)a$ ，但是 $W(y)b$ 是否操作未知，此处应该返回结果 $R(y)NIL$  或者  $R(y)b$ 都是可以的。

3. 如何实现数据副本的因果一致性？给出主要思路。

答案要点：

可以通过vector clock实现对消息处理顺序的控制，达到因果一致性的要求。具体可以通过因果一致性多播控制数据副本访问请求消息的处理顺序，实现因果一致性保证。

# Lec-08 Homework Questions

1. Can the model of triple modular redundancy (三倍模块冗余, TMR) handle Byzantine failures? Why?

答案要点:

TMR与BFT问题是不一样的。TMR是假设存在一个确定的真实值，而且不考虑恶意行为，所以无法用于解决BFT问题。

2. In reliable multicasting, is it always necessary that the communication layer keeps a copy of a message for retransmission purposes?

答案要点:

用于重发的消息可以在通信层保存，也可以由应用程序自身保存。



# Lec-08 Homework Questions

3. In the two-phase commit protocol, why can blocking never be completely eliminated, even when the participants elect a new coordinator?

答案要点：

一方面，即使选出新的coordinator，也无法判断原来的coordinator在crash前做了什么决定，不能判断coordinator recover以后会做什么操作；

另一方面，coordinator的选举本身就是一个共识问题，其难度与commit问题是相当的。

4. Does a stateless server need to take checkpoints?

答案要点：

一般来说，stateless sever的状态信息较少，重新启动即可投入服务，因此checkpoint的必要性低。