分布式系统作业五

数据科学与计算机学院　17大数据与人工智能

17341015　陈鸿峥

问题 1. 通常认为弱一致性模型给程序开发人员增加了额外的负担。这一命题在哪种程度上确实是正确的？

解答. 因为在弱一致性模型下，程序员需要很小心地处理共享数据来实现同步，无论是什么粒度的锁，都是需要程序员硬编码入程序中的，这时候当然就给程序员增加了额外的负担了，这也是为什么并行程序这么难写。在内存模型上无法解决的问题，就只能丢给程序员来解决了。

问题 2. 你会选择哪种类型的一致性来实现电子股票市场？试解释你的答案。

解答. 因果一致性即可。有因果关系的操作需要保持一定次序，而没有因果关系的操作都可以并发执行，并以任何次序被观察。

问题 3. 请列举全序广播的实现方法，解释基本原理并比较它们之间的不同点。

解答. 这里主要列举三种 [1]：

- 排序器算法：特殊的排序器进程为每一条应用程序消息发送额外的排序消息来指明其顺序，但这样子做会产生额外的排序和通信开销，不利于在高负载情况下工作。在具体实现上又可分为：
  - 首先发送操作给排序器(sequencer)，然后再由排序器将分配好的全局序号和操作广播给所有成员
  - 在排序器已经分配好全局序号并广播后，才将操作广播出去
  - 从排序器获取全局序号，然后再广播该操作
- 逻辑时钟算法：使用通信的历史信息建立消息的全序关系，但组成员需要发送空消息保持较高消息发送频率，容易导致网络阻塞
- 优先级算法：使用令牌环实现消息排序，但算法的延迟会随着成员数目的增加而线性增长，不具有可扩展性

问题 4. 请列举顺序一致性、因果一致性、单调读、单调写、读写一致性、写读一致性的具体场景。

解答.

- 顺序一致性：在微博上发表消息，微博会将其放入一个处理序列，并推送给不同的用户。不同用户将会在不同时间看到我的消息，但每个用户都以**同样的顺序**看到我的操作，而不会乱序。

- 因果一致性：比如豆瓣日记的评论机制，只有设置日记非私密**后**，日记才下的评论才可见，即日记可见与日记下的评论具有**因果关系**。只有当前提条件成立，后续的操作才会被看见。
- 单调读：E-mail不管在哪里登录，返回的都是**最近**阅读/**最新**的邮件，那些收到而未读的邮件也不会因为多次登录而改变记录。
- 单调写：软件更新总要等前一次更新**完**，才可以进行当前次更新
- 读写一致性：Markdown编辑器的实时预览功能，可以保证当前写操作结果总会被**后续读操作**看见。
- 写读一致性：各种公告板/贴吧，只有读到原来的消息**才能**看到它的回应（写）消息。

参考文献

[1] 李磊, 王怀民, 史殿习. 一种高性能的全序组播算法 [J]. 计算机研究与发展. 2007.