zookeeper中的paxos算法详解

链接：http://www.aboutyun.com/thread-16575-1-1.html 分布式一致性算法——paxos
视频链接： http://www.tudou.com/v/e8zM8dAL6hM

随着大型网站的各种高并发访问、海量数据处理等场景越来越多，如何实现网站的高可用、易伸缩、可扩展、安全等目标就显得越来越重要。为了解决这样一系列问题，大型网站的架构也在不断发展。提高大型网站的高可用架构，不得不提的就是分布式。在关于分布式事务、两阶段提交协议、三阶提交协议一文中主要用于解决分布式一致性问题的集中协议，那么这篇文章主要讲解业内公认的比较难的也是最行之有效的paxos算法。
Google Chubby的作者Mike Burrows说过， there is only one consensus protocol, and that's Paxos" – all other approaches are just broken versions of Paxos. 意即世上只有一种一致性算法，那就是Paxos，所有其他一致性算法都是Paxos算法的不完整版。

背景

Paxos算法是莱斯利·兰伯特（Leslie Lamport，就是 LaTeX 中的"La"，此人现在在微软研究院）于1990年提出的一种基于消息传递的一致性算法。为描述 Paxos 算法，Lamport 讲述了这样一个故事：

在古希腊有一个岛屿叫做Paxos，这个岛屿通过议会的形式修订法律。执法者（legislators，后面称为牧师priest）在议会大厅（chamber）中表决通过法律，并通过服务员传递纸条的方式交流信息，每个执法者会将通过的法律记录在自己的账目（ledger）上。问题在于执法者和服务员都不可靠，他们随时会因为各种事情离开议会大厅、服务员也有可能重复传递消息（或者直接彻底离开），并随时可能有新的执法者（或者是刚暂时离开的）回到议会大厅进行法律表决，因此，议会协议要求保证上述情况下可以能够正确的修订法律并且不会产生冲突。

什么是paxos算法

Paxos 算法是分布式一致性算法用来解决一个分布式系统如何就某个值(决议)达成一致的问题。

paxos解决了什么问题

在关于分布式一致性的探究中我们提到过，分布式的一致性问题其实主要是指分布式系统中的数据一致性问题。所以，为了保证分布式系统的一致性，就要保证分布式系统中的数据是一致的。

在一个分布式数据库系统中，如果各节点的初始状态一致，每个节点都执行相同的操作序列，那么他们最后能得到一个一致的状态。

所以，paxos算法主要解决的问题就是如何保证分布式系统中各个节点都能执行一个相同的操作序列。

比较常见的例子，C1是一个客户端，N1、N2、N3是分布式部署的三个服务器，初始状态下N1、N2、N3三个服务器中某个数据的状态都是S0。当客户端要向服务器请求处理操作序列：op1op2op3时（op表示operation）（这里把客户端的写操作简化成向所有服务器发送相同的请操作序列，实际上可能通过Master/Slave模式处理）。如果想保证在处理完客户端的请求之后，N1、N2、N3三个服务器中的数据状态都能从S0变成S1并且一致的话（或者没有执行成功，还是S0状态），就要保证N1、N2、N3在接收并处理操作序列op1op2op3时，严格按照规定的顺序正确执行opi，要么全部执行成功，要不就全部都不执行。

所以，针对上面的场景，paxos解决的问题就是如何依次确定不可变操作opi的取值，也就是确定第i个操作什么，在确定了opi的内容之后，就可以让各个副本执行opi操作。

Paxos算法详解

Paxos是一个十分巧妙的一致性算法，但是他也十分难以理解，就连他的作者Lamport都被迫对他做过多种讲解。

我们先把前面的场景简化，把我们现在要解决的问题简化为如何确定一个不可变变量的取值（每一个不可变变量可以标识一个操作序列中的某个操作，当确保每个操作都正确之后，就可以按照顺序执行这些操作来保证数据能够准确无误的从一个状态转变成另外一个状态了）。