链滴

说人话的 PAXOS 算法简介及证明

作者：skyesx

原文链接：https://ld246.com/article/1458136701468

来源网站：链滴

PAXOS是很多分布式系统的基石，获得图灵奖的Lamport的成名作就是关于paxos的算法，这算也是出了名的难理解,即使是Lamport后来写的Paxos made Simple对数学学得不太好的我来说也看云里雾里。市面上介绍Paxos的书基本都是翻译下Paxos made Simple论文的形式，而这篇论文写的如何正向推导出Paxos算法的过程，对吾等凡人来说过于弯绕，实际上，程序员直接看Paxos的代码现，再反推一下，就很容易理解了。下面直入正题。

PAXOS解决的问题是 在分布式环境中确定某一个 “不可变变量X” 的值。请多读几遍。

Paxos协议里有proposer,acceptor,learner三种角色，每中角色的扮演者都可以有多个，proposr提出关于X的值的proposal，acceptor对proposal进行表决 是否接受这个议案，learner负责统计acceptor的表决，若过半数acceptor接收了某个议案，那么关于X的值就被确定了。
Paxos算法能保证在过半数acceptor及至少一个proposer,一个learner存活的情况下，系统能正运行。
不同learner获知最终X值的时间可能不一致,即最终一致性。
根据CAP理论，paxos算法杜绝了脑裂现象，并在C和A之间保留了较好的均衡性。

下面开始描述PAXOS算法的具体运行过程：

每一个Proposer提出的建议都会被编号，且这个编号是全局严格全序的。
（至于不同proposer怎么获得这个严格全序的议案编码不在本文讨论范围，如果确实想知道&hellp;那看懂这篇文章后，后续对PAXOS算法的拓展会提到）

一）proposer:
Proposer p1在向acceptor提出关于X的第n号议案前，都会先向每一个acceptor 发送 promise求——拒绝掉所有小于编号n的promise请求及accept请求（accept请求就是真正的议案的请求）。

二）acceptor:
1、如果某个acceptor之前已经promise过其他proposer 且其议案编号 大于n,那么将会拒绝p1 promise请求；
2、如果某个acceptor当前promise拒绝的议案编号小于n，那么就会接受p1的promise请求，将当前acceptor accept的最新的议案编号及议案内容（即X的值）告诉proposer

三）proposer:
1、如果proposer p1获得了过半acceptor的承诺，则首先整合所有acceptor返回的信息，将acceptor中返回的最大编号的议案的内容作为自己的议案的内容（如果所有acceptor都没有返回议案内容那么就自己随意设定一个），然后向所有acceptor发出自己关于X的accept请求；
2、如果proposer没有获得过半数acceptor的回应，可以忽略这一次操作，申请一个新的议案编，马上重新进行一遍整个操作

四）acceptor:
1、如果acceptor收到了proposer的accept请求时，还没有收到其他proposer更高编号的promie请求时，将会批准proposer的建议，并将自己决定告知learner;
2、如果acceptor收到了proposer的accept请求时，已经收到了其他proposer更高编号的promie请求时，将会拒绝本次accept

五）learner:
Learner收到了某个acceptor的accept信息后，会统计一共收到了多少个acceptor发过来的该proposal的accept信息，若该proposal的acceptor数量达到了quorum,那么就认定该proposal对应的值 X的值。

一个分布式算法很明显需要包含以下特征：

没有确定前是未知的，一旦确定后，它是全局唯一不可变的

我们现在开始论证paxos算法能否在过半acceptor存活时，达到上述要求。
Poposer p1提出X的建议前会向所有的acceptor发出编号为n的promise请求，让acceptor不再准编号小于n的promise和建议，
p1获得了 “过半数acceptor的集合S1” 的promise后,如果某个编号为m（m<）且n-m的值最小的proposal被选定了的话(选定意味着 获得过半数acceptor的集合S2的赞同)，那么1一定能够获取到这个proposal关于X的值（因为S1与S2都是过半数集合，至少有一个acceptor的交，这个交集保存着这个proposal的建议值）

我们在程序流程里可以看到，下一次proposal的建议值必须跟当前acceptors里accept的编号最的proposal一致，因此当 m = n – 1时，n号建议的X的值必然与m相同，依此类推后续propoal关于X的建议都必然与m相同，因此X的值必然全局唯一不可变，得证。

