## 对Pasxo算法的理解

姓名：周韵众

学号：2021443835

班级：大数据2021-01

Paxos算法是由莱斯利.兰伯特于1990年提出的一种基于消息传递的一致性算法，这个算法被认为是类似算法中最有效的一个。

Paxos算法解决的问题是一个分布式系统如何就某个决议达成一致。一个典型的场景是，在一个分布式数据库系统中，如果各个节点的初始状态一致，每个节点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。

Paxos算法有如下角色：

Proposer：向集群提出提案，在发生冲突的时候起到冲突调节的作用。

Acceptor：对提案进行投票，只有在达到多数派时提案才会最终被接受。

proposal：Proposal提出的议案或建议，编号n和内容value。

Paxos算法的流程分为两大阶段：

### 阶段一：（prepare）

(a) Proposer选择一个提案编号N，然后大多数的Acceptor发送编号为N的Prepare请求。

(b) 如果一个Acceptor收到一个编号为N的Prepare请求，且N大于该Acceptor已经响应过的所有Prepare请求的编号，那么它就会将它已经接受过的编号最大的提案（如果有的话）作为响应反馈给Proposer，同时该Acceptor承诺不再接受任何编号小于N的提案。

### 阶段二：（accept）

(a) 如果Proposer收到半数以上Acceptor对其发出的编号为N的Prepare请求的响应，那么它就会发送一个针对[N,V]提案的Accept请求给大多数Acceptor。注意：V就是收到的响应中编号最大的提案的value，如果响应中不包含任何提案，那么V就由Proposer自己决定。这次的大多数Acceptor与阶段一的大多数Acceptor不一定相同。如果没收到足够多的回复，则需要重回到阶段一。

(b) 如果Acceptor收到一个针对编号为N的提案的Accept请求，只要该Acceptor没有对编号大于N的Prepare请求做出过响应，它就接受该提案。否则拒绝或不回应。

该算法的流程简单来说就是，Proposer先从大多数Acceptor那里学习提案的最新内容，然后根据学习到的编号最大的提案内容组成新的提案提交，如果提案获得大多数Acceptor的投票通过就意味着提案被通过。

Paxos算法的最终目的是使每个Proposer，Acceptor都认为同一个Proposal中的value被选中。

Paxos算法在实现过程中还存在一些如活锁、效率低等问题，都需要我们用相应的方法去解决。