

www.cnblogs.com/linbingdong/p/6253479.html

paxos算法的通俗理解:

paxos算是一种基于消息传递且具有高度容错特征的一致性算法。

Paxos算法目前在Hadoop中的Zookeeper也使用了paxos算法，在上面的各个系统中，使用的算法与原始的算法不同。因为paxos算法比较复杂，可以从生活中的小事开始。

提案=[提案编号, value]

在paxos算法中，有Proposer、Acceptor、Learners三种角色，在具体实现过程中，一个进程可能同时充当多种角色。

Proposer可以提出提案，Acceptor可以接受提案，如果某个提案被选定，那么该提案里的value就被选定了。

当Proposer发的提案被Acceptor接受（需要半数以上的Acceptor同意才行），Proposer就认为该提案里的value被选定了。

Acceptor: 只要Acceptor接受了某个提案，Acceptor就认为该提案里的value被选定了。

Learner: Acceptor告诉Learner哪个value被选定了，Learner就认为哪个value被选定了。

我们假设有一组提出提案的进程集合。在paxos算法中，一致性算法需要保证提出的这么多的提案中，只有一个提案被选定。如果没有提案被提出，就不应该有提案被选定。如果一个提案被选定，那么所有进程都应该能学习到这个被选定的提案。对于一致性算法，安全性要求如下：

1. 只有被提出的提案才能被选定，2. 只有一个提案被选定

推导过程:

假设只有一个Acceptor，所有的Proposer向其发起提案，那么当这个Acceptor宕机后，整个系统就无法工作了。所以需要有多个Acceptor。

解决方案:

1. 一个Acceptor必须接受它收到的第一个提案。
 2. 一个提案被选定，需要半数以上的Acceptor接受。
 3. 如果某个value为v的提案被选定了，那么每个编号更高的被选定提案的value必须也是v
- 由于一个提案只有被Acceptor接受了才可能被选定，因此我们可以把约束3变成约束4

4. 如果某个value为v的提案被选定了，那么每个编号更高的被Acceptor接受的提案的value必须也是v
5. 如果某个value为v的提案被选定了，那么之后的任何Proposer提出的编号更高的提案的value必须也是v。

因此Paxos算法的流程，主要分为两个阶段：

第一阶段：

1. Proposer选择一个提案编号N，然后向半数以上的Acceptor发送编号为N的Prepare请求。
2. 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。

第二阶段：

1. 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，那么它就会发一个针对[N, V]提案的Accept请求给半数以上的Acceptor。注意：V就是收到的响应中编号最大提案的value，如果响应中不包含任何提案，那么V就由Proposer自己决定。
2. 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请求作出响应，它就该接受该提案。