**Raft一致性协议**

**一、节点状态**

Follower：所有节点都以follower的状态开始。如果没收到leader消息则会变成candidate状态

Candidate：会向其他结点“拉选票”，如果得到大部分的票则成为leader。这个过程就叫做Leader选举(Leader Election)

leader：所有对系统的修改都会先经过leader。每个修改都会写一条日志(log entry)。leader收到修改请求后的过程如下，这个过程叫做日志复制

(Log Replication)：

1.复制日志到所有follower结点(replicate entry)

2.大部分结点响应时才提交日志

3.通知所有follower结点日志已提交

4.所有follower也提交日志

5.现在整个系统处于一致的状态

Raft大概将整个过程分为三个阶段，leader election，log replication

**二、Leader**

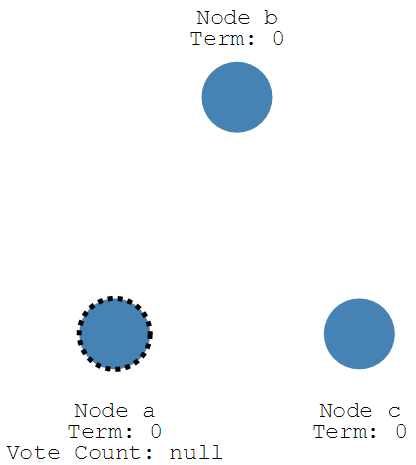
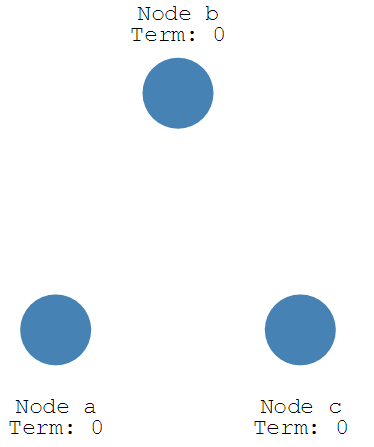


图1 图2

1）一开始所有节点a,b,c都是follower状态

2）当follower在选举超时时间(election timeout)内未收到leader的心跳消息(append entries)，则变成candidate状态。为了避免选举冲突，这个超时时间是一个150~300ms之间的随机数。此时图2中的a变成了candidate

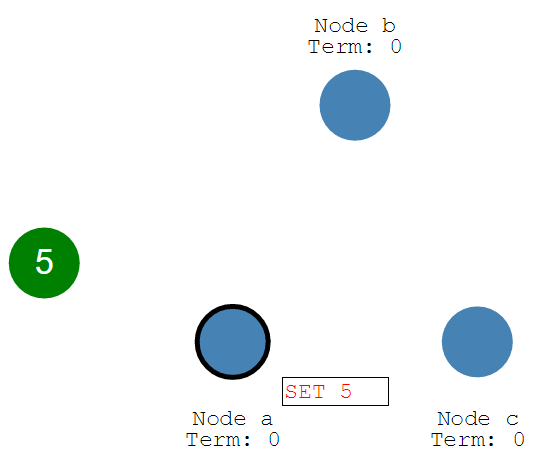
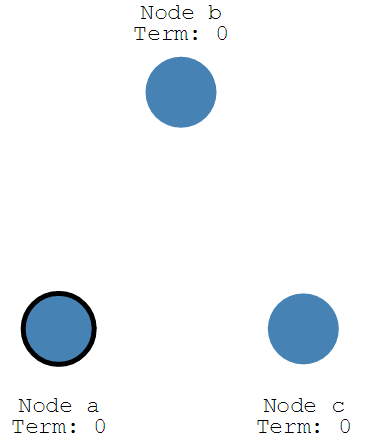


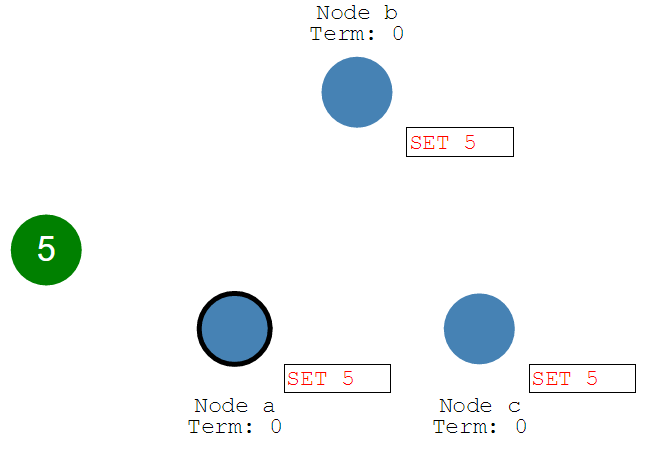
图3 图4

3）节点a会发起拉票，如果它收到大多数的投票就会成为leader,置 Term为0

4）此时所有的改变都会通过leader也就是node a。

5）此时绿色的客户端发起一个请求，将Node的数值设置为5

6）所有的改变都会以entry的形式加入到node log中



7）然后Node将log发送到其他Node中，如图5

8）Node a（Leader）等待其他节点的回复，当收到大多数节点的回复后，通知其他Node提交完成。

上述1）-8）就是Leader Election + Log Replication