**MongoDB的选举过程**

[十月 20, 2014](http://www.mongoing.com/archives/295)[贝贝的博客](http://www.mongoing.com/beibei)[mongodb](http://www.mongoing.com/archives/tag/mongodb)、[复制集](http://www.mongoing.com/archives/tag/%e5%a4%8d%e5%88%b6%e9%9b%86)、[选举](http://www.mongoing.com/archives/tag/%e9%80%89%e4%b8%be)[贝贝](http://www.mongoing.com/archives/author/xbsura)

科普文,MongoDB的选举过程.  
MongoDB的复制集具有自动容忍部分节点宕机的功能,在复制集出现问题时时,会触发选举相关的过程,完成主从节点自动切换.  
每个复制集成员都会在后台运行与复制集所有节点的心跳线程,在两种情况下会触发状态检测过程:

* 复制集成员心跳检测结果发生变化,比如某个节点挂了或者新增节点.
* 超过4s没有执行状态检测过程.

在状态检测过程大致包含以下步骤:

* 检测自身是否处于选举过程,如果是,退出本次过程.
* 维护一个主节点的备用列表,列表中所有节点都可能被选举为主节点,每个节点都会检测自身以及全局条件是否满足:
  + 是否看见复制集中是否有Majority在线.
  + 自身priority大于0.
  + 自身不为arbiter.
  + 自身opTime不能落后于最新节点10s以上.
  + 自身存储的集群程序按信息为最新.

如果所有条件满足,则将自身添加到主节点备用列表中,否则,将自身从列表中移除.

* 检测以下条件,若都满足,将主节点将为从节点(如果要降级的主节点是自身,直接调用降级方法,如果不为自身,调用replSetStepDown命令将复制集主节点降级为从节点.):
  + 集群中主节点存在.
  + “主节点的备用列表”中存在比当前的主节点priority更高的节点.
  + “主节点的备用列表”中priority最高的节点,其opTime要比其他所有节点最新的opTime落后10s以内.
* 检测自身是否为主,若为主,且自身无法看见复制集的Majority在线,将自身降级为从.
* 如果看不见集群中有主节点存在,检测自身是否在”主节点的备用列表”,若不在,打印log并退出此流程.
* 若自身在”主节点的备用列表”中,开始判断自身可否向复制集中发送选举自身为主节点的通知,判断过程包含:
  + 自身是否可以看见复制集中的Majority在线.
  + 自身是否在”主节点的备用列表”.  
    若条件满足,则设置”自身已经在选举过程中”标识位为true,并进入”选举自身为主节点”方法.
* 方法中会验证自身是否满足以下条件:
  + 此线程拿到了线程锁.
  + 此节点没有被配置slaveDelay选项或者配置的slaveDelay为0.
  + 此节点没有被配置为arbiter.  
    若满足,则调用环境检测,若以下条件被触发,则**不发送**“选举我为主节点”投票:
  + 当前时间小于steppedDown的结束冻结时间(为执行steppedDown时的时间+冻结设定时间,内部调用为60s).
  + 自己的opTime不是所有节点最新的.
    - 若有节点opTime比自己新,直接退出此流程.
    - 如果其他最新的节点最多与自己一样新,每有一个这样的节点,随机sleep一段时间,之后继续判断.
  + 自己上线5分钟内且复制集中不是所有节点在线.
* 如无其他问题,尝试获取自己进行投票时的票数,在此过程中,会判断自己在30s内是否进行过投票,如进行过,直接退出整个过程.
* 经过以上种种复杂的检测,终于可以向复制集发送”选举我为主节点”的投票.
* 发送之后,会接收来自所有节点的投票,若得票数小于等于一半,不将自己变为主节点,若超过一半,设置自己为主节点.  
  投票结束后,设置”自身已经在选举过程中”标识位为false.  
  可以看到,上面的判断逻辑有一些是重复判断,不过不影响最终结果,可能与判断逻辑较为复杂有关系,在每个决定之前都要验证所有条件是否满足,防止有条件被漏掉.  
  在复制集中的节点收到其他节点发送的”选举我为主节点”投票信息时,会有以下的判断:
* 若自身存储的复制集配置版本过低,不投票.
* 若发起请求的节点存储的复制集配置版本过低,投反对票.
* 如果自身所在的复制集没有发起投票的节点,投反对票.
* 复制集中存在主节点,投反对票.
* 可参与选举的节点中有priority高于请求为主的节点存在时,投反对票.  
  如果所有条件通过,获取自身的投票数(同样会判断自身在30s内是否参加过投票,若参加过,不再投票),投出票数.  
  需要说一下的是,一个反对会将最终票数**减10000**,即在绝大多数情况下,只要有节点反对,请求的节点就不能成为主节点.  
  **选举过程很复杂,实际使用中总结为两点:**
* **一般情况下需要5s左右进行选主.**
* **如果新选举出的主节点立马挂掉,至少需要30s时间重新选主.**