链滴

zookeeper 的容错与脑裂问题

作者：howie404

原文链接：https://ld246.com/article/1563777437594

来源网站：链滴

一 、zookeeper的容错机制

1.由于在增删改查操作中需要半数以上的服务器通过,来分析一下情况

1主节点挂了，另外一个备节点因为没有过半，无法对外提供集群服务，容错数为0

3节点zookeeper，一个主节点挂了，另外两个备节点过半，顺利选出Leader对外提供集群服务，容数为1

5节点zookeeper，两个主节点挂了，另外三个备节点过半，对外提供集群服务，容错数为2

6节点zookeeper，两个主节点挂了，另外四个备节点过半，对外提供集群服务，容错数为2，

起第三个主节点，另外三个备节点没有过半，也就无法梳理选举出Leader。

2.总结:

 如果有2个zookeeper，那么只要有1个死了zookeeper就不能用了，因为1没有过半，所以2个zooeeper的死亡容忍度为0；同理，要是有3个zookeeper，一个死了，还剩下2个正常的，过半了，所以个zookeeper的容忍度为1；同理你多列举几个：2->0;3->1;4->1;5->2;6->2会发现一个规律，2n和n-1的容忍度是一样的，都是n-1，所以为了更加高效，何必增加那一个不必要的zookeeper呢。

二、zookeeper脑裂（Split-Brain）问题

1.什么是脑裂?

 比如当你的 cluster 里面有两个结点，它们都知道在这个 cluster 里需要选举出一个 master。那当它们两之间的通信完全没有问题的时候，就会达成共识，选出其中一个作为 master。但是如果它之间的通信出了问题，那么两个结点都会觉得现在没有 master，所以每个都把自己选举成 master。是 cluster 里面就会有两个 master。

 对于Zookeeper来说有一个很重要的问题，就是到底是根据一个什么样的情况来判断一个节点死亡own掉了。 在分布式系统中这些都是有监控者来判断的，但是监控者也很难判定其他的节点的状态，一一个可靠的途径就是心跳，Zookeeper也是使用心跳来判断客户端是否仍然活着。

 使用ZooKeeper来做master HA基本都是同样的方式，每个节点都尝试注册一个象征master的临节点其他没有注册成功的则成为slaver，并且通过watch机制监控着master所创建的临时节点，Zookeeper通过内部心跳机制来确定master的状态，一旦master出现意外Zookeeper能很快获悉并且通知他的slaver，其他slaver在之后作出相关反应。这样就完成了一个切换。这种模式也是比较通用的模，基本大部分都是这样实现的，但是这里面有个很严重的问题，如果注意不到会导致短暂的时间内系出现脑裂，因为心跳出现超时可能是master挂了，但是也可能是master，zookeeper之间网络出现问题，也同样可能导致。这种情况就是假死，master并未死掉，但是与ZooKeeper之间的网络出现题导致Zookeeper认为其挂掉了然后通知其他节点进行切换，这样slaver中就有一个成为了master，是原本的master并未死掉，这时候client也获得master切换的消息，但是仍然会有一些延时，zookeeer需要通讯需要一个一个通知，这时候整个系统就很混乱可能有一部分client已经通知到了连接到新master上去了，有的client仍然连接在老的master上如果同时有两个client需要对master的同一个数更新并且刚好这两个client此刻分别连接在新老的master上，就会出现很严重问题。

2.总结

原文链接：zookeeper 的容错与脑裂问题

- 假死：由于心跳超时（网络原因导致的）认为master死了，但其实master还存活着。

- 脑裂：由于假死会发起新的master选举，选举出一个新的master，但旧的master网络又通了，导出现了两个master ，有的客户端连接到老的master 有的客户端链接到新的master。

3.出现原因

- 在搭建hadoop的HA集群环境后，由于两个namenode的状态不一，当active的namenode由于络等原因出现假死状态，standby接收不到active的心跳，因此判断active的namenode宕机，但实上active并没有死亡。此时standby的namenode就会切换成active的状态，保证服务能够正常使用若原来的namenode复活，此时在整个集群中就出现2个active状态的namenode，该状态成为脑裂脑裂现象可能导致这2个namenode争抢资源，从节点不知道该连接哪一台namenode，导致节点的据不统一，这在企业生产中是不可以容忍的。

4.解决方案（三种）

- 1、添加心跳线。

原来两个namenode之间只有一条心跳线路，此时若断开，则接收不到心跳报告，判断对方经死亡。此时若有2条心跳线路，一条断开，另一条仍然能够接收心跳报告，能保证集群服务正常运。2条心跳线路同时断开的可能性比1条心跳线路断开的小得多。再有，心跳线路之间也可以HA（高用），这两条心跳线路之间也可以互相检测，若一条断开，则另一条马上起作用。正常情况下，则不作用，节约资源。

- 2、启用磁盘锁。

由于两个active会争抢资源，导致从节点不知道该连接哪一台namenode，可以使用磁盘锁形式，保证集群中只能有一台namenode获取磁盘锁，对外提供服务，避免数据错乱的情况发生。但，也会存在一个问题，若该namenode节点宕机，则不能主动释放锁，那么其他的namenode就永远取不了共享资源。因此，在HA上使用"智能锁"就成为了必要措施。"智能锁"是指active的namenode测到了心跳线全部断开时才启动磁盘锁，正常情况下不上锁。保证了假死状态下，仍然只有一台namenode的节点提供服务。

- 3、设置仲裁机制

脑裂导致的后果最主要的原因就是从节点不知道该连接哪一台namenode，此时如果有一方决定谁留下，谁放弃就最好了。因此出现了仲裁机制，比如提供一个参考的IP地址，当出现脑裂现象，双方接收不到对方的心跳机制，但是能同时ping参考IP，如果有一方ping不通，那么表示该节点网已经出现问题，则该节点需要自行退出争抢资源的行列，或者更好的方法是直接强制重启，这样能更的释放曾经占有的共享资源，将服务的提供功能让给功能更全面的namenode节点。

以上的3种方式可以同时使用，这样更能减少集群中脑裂情况的发生。但是还是不能保证完全不出现如果仲裁机制中2台机器同时宕机，那么此时集群中没有namenode可以使用。此时需要运维人员人的抢修，或者提供一台新的机器作为namenode，这个时间是不可避免的。希望未来能有更好的解决法，能彻底杜绝这类情况的发生吧~