ZooKeeper实现master选举

|  |  |  |  |
| --- | --- | --- | --- |
| 编写人 | 编写时间 | 版本 | 备注 |
| 张贵东 | 2017年8月 | V1.0 |  |

目 录

[1. 介绍 2](#_Toc490256326)

[1.1. 使用场景 2](#_Toc490256327)

[1.2. 实现原理 2](#_Toc490256328)

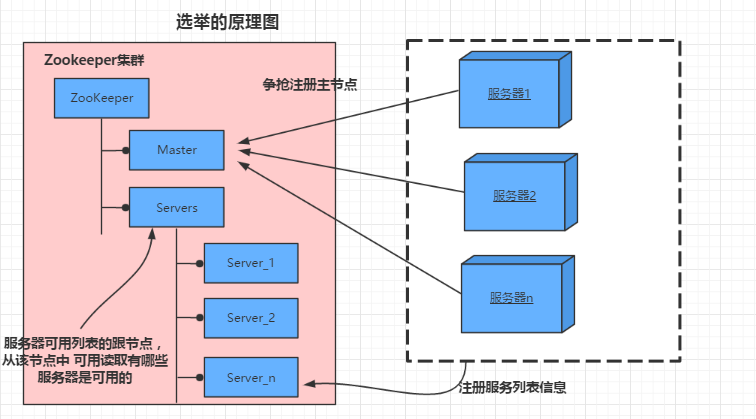
[1.3. ZooKeeper代码实现 3](#_Toc490256329)

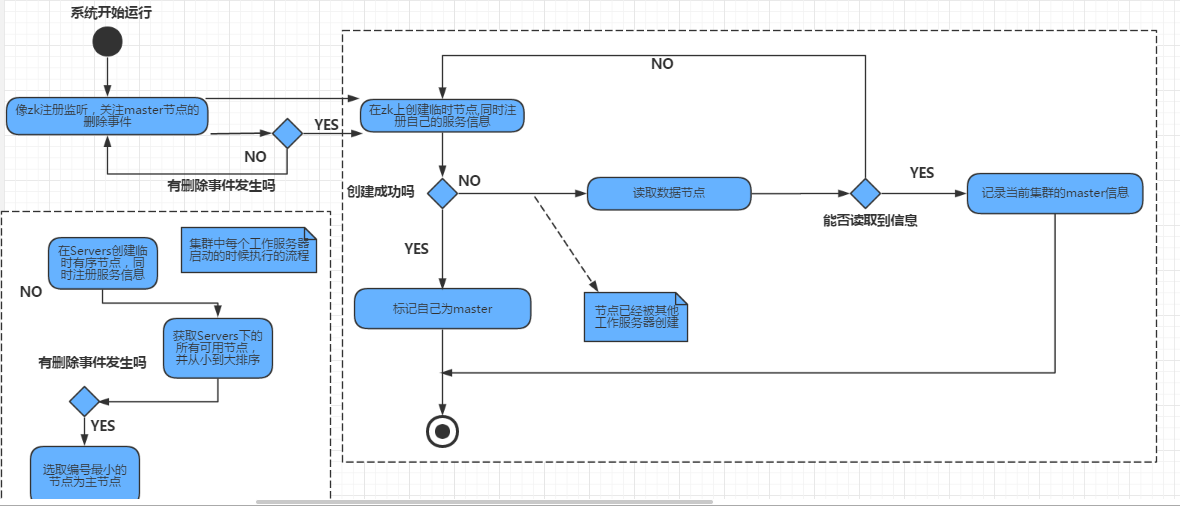
1. 介绍
   1. 使用场景

现在很多时候我们的服务需要7\*24小时工作，假如一台机器挂了，我们希望能有其它机器顶替它继续工作。此类问题现在多采用master-salve模式，也就是常说的主从模式，正常情况下主机提供服务，备机负责监听主机状态，当主机异常时，可以自动切换到备机继续提供服务(这里有点儿类似于数据库主库跟备库，备机正常情况下只监听，不工作)，这个切换过程中选出下一个主机的过程就是master选举。

* 1. 实现原理

选主原理介绍：zookeeper的节点有两种类型，持久节点跟临时节点。临时节点有个特性，就是如果注册这个节点的机器失去连接(通常是宕机)，那么这个节点会被zookeeper删除。选主过程就是利用这个特性，在服务器启动的时候，去zookeeper特定的一个目录下注册一个临时节点(这个节点作为master，谁注册了这个节点谁就是master)，注册的时候，如果发现该节点已经存在，则说明已经有别的服务器注册了(也就是有别的服务器已经抢主成功)，那么当前服务器只能放弃抢主，作为从机存在。同时，抢主失败的当前服务器需要订阅该临时节点的删除事件，以便该节点删除时(也就是注册该节点的服务器宕机了或者网络断了之类的)进行再次抢主操作。从机具体需要去哪里注册服务器列表的临时节点，节点保存什么信息，根据具体的业务不同自行约定。选主的过程，其实就是简单的争抢在zookeeper注册临时节点的操作，谁注册了约定的临时节点，谁就是master。





* 1. ZooKeeper代码实现

详见GitHub地址：<https://git.oschina.net/BJZGD/zookeeper-demo>