# Zookeeper使用

## Zookeeper 简介

Zookeeper是一个高效的分布式协调服务，它暴露了一些公共服务，比如命名，配置管理，同步控制，群组服务等。我们可以使用Zookeeper来实现比如达成共识，集群管理，leader选举等。

Zookeeper是一个高可用的分布式管理与协调框架， 基于ZAB算法（源自消息广播协议）的实现。该框架能够很好地保护分布式环境中的数据的一致性。也正是基于这样的特性，使得Zookeeper成为了解决分布式一致性问题的利器。

顺序性：从一个客户端发起的事务请求，最终将会严格地按照其发起的顺序被执行应用到zookeeper中去。

原子性：所有事务情字u的处理结果在整个集群中所有机器上的应用情况是一致的，也就是说，要么整个集群所有的机器都成功应用了某一事务，要么有没有应用，一定不会出现部分机器应有了该事务，而另一部分没有应用的情况。

单一视图：无论客户端连接的是哪一个zookeeper服务器，其看到的服务器端数据模型都是一致的。

可靠性：一旦服务器成功地应用了一个事务，并完成对客户端的响应，那么该事务所引起的服务器端状态将会被一致保留下来。除非有另外的一个事务对其更改。

实时性：通常所说的实时性就是指一档事务被成功应用，那么客服端就能立刻从服务器上获取变更后的新数据，zookeeper仅仅能保证在一段时间内，客户端最终能从服务器端读取最新的数据状态

## Zookeeper组成

Zookeeper server 根据其身份分为三种，leader，follower，observer，其中follower和observer统称为learner（学习者）。

leader：负责客户端的writer类型请求

follower：负责客户端的reader类型请求，参与leader选举等

observer：特殊的“follower”其可以接收客户端reader请求，但不参与选举。（扩容系统支撑能力，提高读取速度，因为它不接受任何同步的写入请求只负责与leader同步数据）

## Zookeeper 应用场景

典型应用场景

Zookeeper 从设计模式的角度来看，是一个基于观察者模式设计的分布式服务管理框架，它负责存储和管理大家都关心的数据，然后就接受观察者的注册，一旦这些数据的状态发生变化，zookeeper就将负责通知已经在zookeeper上注册的那些观察者做出相应的反应，从而实现集群中类似Master/Slave管理模式。

配置模式、集群管理、发布与订阅、数据库切换、分布式日志和收集、分布式锁队列管理等等

## Zookeeper集群搭建

1 一共三个节点（zookeeper服务器集群规模不小于三个节点）要求服务器之间系统时间保持一致。

2 进行解压 tar zookeeper-3.4.5.tar.gz

3 重命名 mv zookeeper-3.4.5 zookeeper

4 修改环境变量 vi /etc/profile

export ZOOKEEPER\_HOME=/usr/local/zookeeper

export PATH=.:$HADOOP\_HOME/bin:$ZOOKEEPER\_HOME/bin:$JAVA\_HOME/...

5 刷新 source /etc/profile

6 到zookeeper下修改配置文件

cd /usr/local/zookeeper/conf

mv zoo\_sample.conf zoo.cfg

7 修改conf vi zoo.cfg 修改两处

1. dataDir = /usr/local/zookeeper/data
2. 在最后面添加

server.0 =xxx:2888:3888 （xxx主机名或IP地址）

server.1=hadoop1:2888:3888

server.2=hadoop2:2888:3888

8 服务器标识配置

创建文件夹：mkdir data

创建文件myid并填写内容为0：vi myid（内容为服务器标识：0） 3台服务器每一台都需要创建一个myid的文件，并且在文件中进行复制 server.0 设置0 .。。。。

9 进行复制zookeeper目录到hadoop01和hadoop01还有/etc/profile文件

10 把hadoop01、hadoop02中的myid文件的值修改为1和2路径（vi /usr/local/zookeeper/data/myid）

11 启动zookeeper

路径： /usr/local/zookeeper/bin

执行： zookeeperServer.sh start (注意这里3台机器都要进行启动)

状态：zookeeperServer.sh status(在三个节点上检验zookeeper的mode,一个leader和两 个follower)

12 关闭防火墙

[root@localhost zookeeper-3.4.10]# firewall-cmd --zone=public --add-port=2888/tcp --permanent

FirewallD is not running

[root@localhost zookeeper-3.4.10]# firewall-cmd --zone=public --add-port=3888/tcp --permanent

FirewallD is not running

## ZooKeeper操作shell

使用zkCli.sh命令进入zookeeper客户端

zkCli.sh -timeout 5000 -server 127.0.0.1:2181

查找：ls / ls/zookeeper

创建并赋值： create /test hadoop

获取： get /test

设值： se /test xxxx

可以看到zookeeper集群的数据一致性

rmr /path 递归删除节点

delete /path/child 删除指定某个节点

1. Watcher、ZK状态

zookeeper有watch时间，是一次性触发的，当watch监视的数据发生变化的时候，通知设置了该watch的client，即watcher

同样，其watcher是监听数据发送了某些变化，那就一定会有对应的事件类型喝状态类型。

事件类型：（zonde节点相关的）

EventType.NodeCreated

EventType.NodeDataChanged

EventType.NodeChildredChanged

EventType.NodeDeleted

状态类型（是客户端实例相关的）

KeeperState.Disconnected

KeeperState.SynConnected

KeeperState.AuthFailed

KeeperState.Expired