# 一、zookeeper介绍

zookeeper是一个分布式的开源框架，它能很好的管理集群，而且提供协调分布式应用的基本服务。

它向外部应用暴露一组通用服务——分布式同步（Distributed Synchronization）、命名服务（Naming Service）、集群维护（Group Maintenance）等，简化分布式应用协调及其管理的难度，提供高性能的分布式服务。

zookeeper本身可以以standalone模式（单节点状态）安装运行，不过它的长处在于通过分布式zookeeper集群（一个leader，多个follower），基于一定的策略来保证zookeeper集群的稳定性和可用性，从而实现分布式应用的可靠性。

## 1.1、zookeeper集群角色介绍

zookeeper集群中主要有两个角色：leader和follower。

领导者（leader）,用于负责进行投票的发起和决议,更新系统状态。

学习者（learner）,包括跟随者（follower）和观察者（observer）。

其中follower用于接受客户端请求并向客户端返回结果,在选主过程中参与投票。

而observer可以接受客户端连接,将写请求转发给leader,但observer不参加投票过程,只同步leader的状态,observer的目的是为了扩展系统,提高读取速度。

## 1.2 zookeeper集群节点个数

一个zookeeper集群需要运行几个zookeeper节点呢？

你可以运行一个zookeeper节点，但那就不是集群了。如果要运行zookeeper集群的话，最好部署3，5，7个zookeeper节点。本次实验我们是以3个节点进行的。

zookeeper节点部署的越多，服务的可靠性也就越高。当然建议最好是部署奇数个，偶数个不是不可以。但是zookeeper集群是以宕机个数过半才会让整个集群宕机的，所以奇数个节点更佳。

你需要给每个zookeeper 1G左右的内存，如果可能的话，最好有独立的磁盘，因为独立磁盘可以确保zookeeper是高性能的。如果你的集群负载很重，不要把zookeeper和RegionServer运行在同一台机器上面，就像DataNodes和TaskTrackers一样。

# 二、zookeeper安装

zookeeper介绍完毕后，我们现在就来安装zookeeper。而zookeeper运行需要java的支持，所以我们要首先安装jdk，jdk的安装就在这里忽略了。

注意：zookeeper、jdk需要在每台服务器上都要安装和运行，并且关闭防火墙。

## 2.1  安装zookeeper

zookeeper的安装很简单，只需要下载zookeeper压缩包，然后解压、配置、执行就可以了。

#wget <http://mirrors.cnnic.cn/apache/zookeeper/zookeeper-3.4.8/zookeeper-3.4.8.tar.gz>

#tar -C /usr/local/ -xf zookeeper-3.4.8.tar.gz

#cd /usr/local/

#mv zookeeper-3.4.8 zookeeper

#chown root:root -R zookeeper/

#cd zookeeper

#cp conf/zoo\_sample.cfg conf/zoo.cfg

把zookeeper加入到环境变量，如下：

#vim /etc/profile

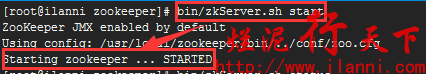
export PATH=$PATH:/usr/local/zookeeper/bin

执行source命令使刚刚修改的环境生效：

#source /etc/profile

现在在其中任意一台服务器上启动zookeeper，使用如下命令：

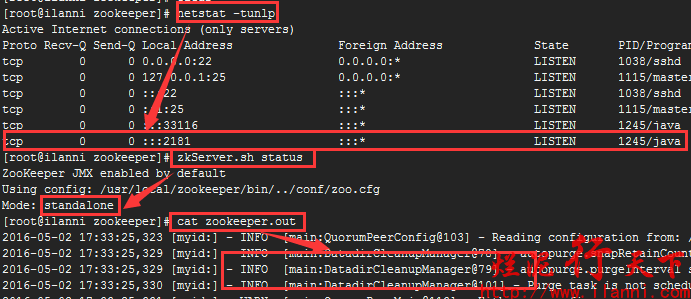
#bin/zkServer.sh start



查看zookeeper端口、状态和zookeeper日志，如下：

#netstat -tunlp

#bin/zkServer.sh status



通过上图，可以很明显的看到zookeeper监听的是TCP的2181端口，是以standalone单机模式运行的，而且通过zookeeper日志也可以看到目前zookeeper是正常运行的。

**三、zookeeper集群搭建**

注意：在搭建zookeeper集群时，一定要停止已经启动的zookeeper。

## 3.1 修改zookeeper配置文件

zookeeper集群的搭建很简单，只需要修改每台服务器的zoo.cfg配置文件然后进行相关的操作即可。

#vim /usr/local/zookeeper/conf/zoo.cfg

tickTime=2000

initLimit=10

syncLimit=5

clientPort=2181

dataLogDir=/usr/local/zookeeper/logs

dataDir=/usr/local/zookeeper/data

server.1= 192.168.0.4:2888:3888

server.2= 192.168.0.8:2888:3888

**配置文件参数说明:**

tickTime这个时间是作为zookeeper服务器之间或客户端与服务器之间维持心跳的时间间隔,也就是说每个tickTime时间就会发送一个心跳。

initLimit这个配置项是用来配置zookeeper接受客户端（这里所说的客户端不是用户连zookeeper服务器的客户端,而是zookeeper服务器集群中连接到leader的follower 服务器）初始化连接时最长能忍受多少个心跳时间间隔数。

当已经超过10个心跳的时间（也就是tickTime）长度后 zookeeper 服务器还没有收到客户端的返回信息,那么表明这个客户端连接失败。总的时间长度就是 10\*2000=20秒。

syncLimit这个配置项标识leader与follower之间发送消息,请求和应答时间长度,最长不能超过多少个tickTime的时间长度,总的时间长度就是5\*2000=10秒。

dataDir顾名思义就是zookeeper保存数据的目录,默认情况下zookeeper将写数据的日志文件也保存在这个目录里。

clientPort这个端口就是客户端连接Zookeeper服务器的端口,Zookeeper会监听这个端口接受客户端的访问请求。

server.A=B:C:D中的A是一个数字,表示这个是第几号服务器,B是这个服务器的IP地址，C第一个端口用来集群成员的信息交换,表示这个服务器与集群中的leader服务器交换信息的端口，D是在leader挂掉时专门用来进行选举leader所用的端口。

## 3.2 创建相关目录

配置文件修改完毕后，创建zookeeper数据目录和日志目录，使用如下命令：

#mkdir -p /usr/local/zookeeper/logs

#mkdir -p /usr/local/zookeeper/data

## 3.3 创建ServerID标识

除了修改zoo.cfg配置文件外,zookeeper集群模式下还要配置一个myid文件,这个文件放在dataDir目录下。这个文件里面有一个数据就是A的值（该A就是zoo.cfg文件中server.A=B:C:D中的A）,在zoo.cfg文件中配置的dataDir路径中创建myid文件。

在192.168.0.4服务器上创建myid文件，并设置为1，同时与zoo.cfg文件里面的server.1对应，如下：

#echo “1” > /usr/local/zookeeper/data/myid

在192.168.0.8服务器上创建myid文件，并设置为2，同时与zoo.cfg文件里面的server.2对应，如下：

#echo “2” > /usr/local/zookeeper/data/myid

**四、启动并查看zookeeper集群**

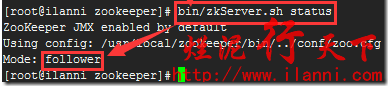
#cd /usr/local/zookeeper

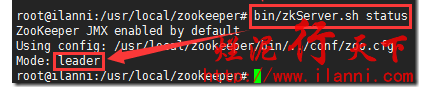
#bin/zkServer.sh start

注意:启动第一台zookeeper的时候可能会报错，等最后一台zookeeper全部启动之后就不会报错了。

zookeeper启动完毕后，查看各个服务器上zookeeper的状态。如下：

#bin/zkServer.sh status





通过上图，可以很明显的看出这两台服务器zookeeper的状态是follower模式和leader模式。

这说明zookeeper集群已经成功搭建。

**五、连接zookeeper集群**

对于客户端来说，zookeeper集群是一个整体，连接到zookeeper集群实际上感觉在独享整个集群的服务，所以可以在任何一个节点上建立到服务器集群的连接，例如在0.8上进行登录0.4：

#bin/zkCli.sh -server 192.168.0.4:2181

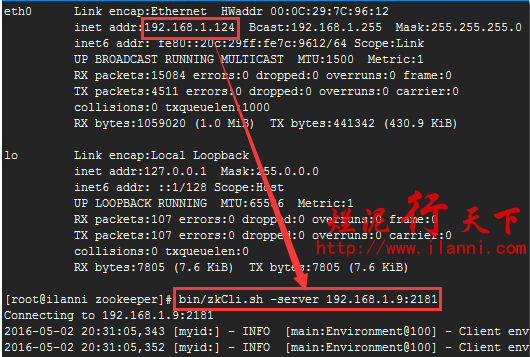
注意：下面的1.9的ip相当于0.4, 1.124相当于0.8，本人比较懒，没有用自己的截图。

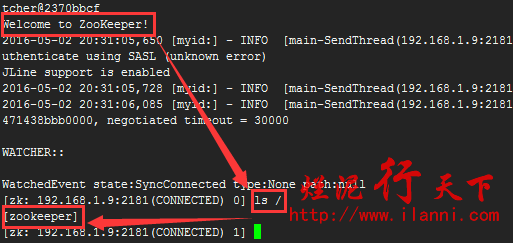
ZooKeeper启动后，在本机监听一个端口，可以通过nc向此端口发指令进行验证，如果启动成功会返回imok。

#echo ruok | nc [zookeeper\_server\_ip] [zookeeper\_server\_port]

举例：

#echo ruok | nc 192.168.0.5 2181





通过上图，可以很明显的看出在192.168.1.124这台机器上连接192.168.1.9服务器上的zookeeper是正常的，而且当前根路径为/zookeeper。