zookeeper单节点安装和伪分布式集群安装和完全分布式集群安装

1. zookeeper介绍

ZooKeeper是一个为分布式应用所设计的分布的、开源的协调服务，它主要是用来解决分布式应用中经常遇到的一些数据管理问题，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。

ZooKeeper是作为分布式协调服务，是不需要依赖于Hadoop的环境，也可以为其他的分布式环境提供服务。

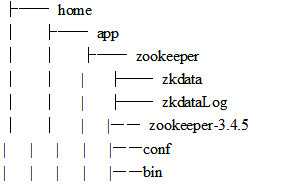
2. zookeeper单节点安装Standalones模式

创建目录

在home目录下创建app/zookeeper目录，并将zookeeper-3.4.5.tar.gz移到zookeeper

下，并在zookeeper下创建zkdata和zkdataLog子目录,也可以根据个人习惯创建目录。

目录结构：



进入到cd /home/app/zookeeper/zookeeper-3.4.5/conf/,备份该文件

cp zoo\_sample.cfg zoo.cfg , 修改配置文件conf/zoo.cfg

# vi conf/zoo.cfg

ckTime=2000

initLimit=10

syncLimit=5

dataDir=/home/app/zookeeper/zkdata

#dataLogDir=/home/app/zookeeper/zkdataLog

clientPort=2181

非常简单，我们已经配置好了的zookeeper单节点

启动zookeeper

# bin/zkServer.sh  start

JMX enabled by default

Using config: /home/app/zookeeper/zookeeper-3.4.5/bin/../conf/zoo.cfg

Usage: bin/zkServer.sh {start|start-foreground|stop|restart|status|upgrade|print-cmd}

conan@conan:~/zookeeper/zookeeper-3.4.5$ bin/zkServer.sh start

JMX enabled by default

Using config: /home/app/zookeeper/zookeeper-3.4.5/bin/../conf/zoo.cfg

Starting zookeeper ... STARTED

#zk的服务显示为QuorumPeerMain

# jps 或#ps -ef|grep java

5321 QuorumPeerMain

5338 Jps

#查看运行状态

~ bin/zkServer.sh status

JMX enabled by default

Using config: /home/app/zookeeper/zookeeper-3.4.5/bin/../conf/zoo.cfg

Mode: standalone 节点的时，Mode会显示为standalone

停止ZooKeeper服务

# bin/zkServer.sh stop

JMX enabled by default

Using config: /home/app/zookeeper/zookeeper-3.4.5//bin/../conf/zoo.cfg

Stopping zookeeper ... STOPPED

3. zookeeper伪分布式集群安装

所谓“伪分布式集群”就是在，在一台PC中，启动多个ZooKeeper的实例。“完全分布

式集群”是每台PC，启动一个ZooKeeper实例。由于我的测试环境PC数量有限，所以在一台PC中，启动3个ZooKeeper的实例。

创建环境目录

~ mkdir /home/app/zookeeper/zkdata1

~ mkdir /home/app/zookeeper/zkdata2

~ mkdir /home/app/zookeeper/zkdata3

#新建myid文件

~ echo "1" > /home/app/zookeeper/zkdata1/myid

~ echo "2" > /home/app/zookeeper/zkdata2/myid

~ echo "3" > /home/app/zookeeper/zkdata3/myid

分别修改配置文件

修改：dataDir,clientPort两个参数

增加：集群的实例，server.X，”X”表示每个目录中的myid的值

~ vi  /home/app/zookeeper/zookeeper-3.4.5/conf/zoo1.cfg

ckTime=2000

initLimit=10

syncLimit=5

dataDir=/home/app/zookeeper/zkdata1

clientPort=2181

server.1=192.168.1.108:2888:3888

server.2=192.168.1.108:2889:3889

server.3=192.168.1.108:2890:3890

~ vi  /home/app/zookeeper/zookeeper-3.4.5/conf/zoo2.cfg

ckTime=2000

initLimit=10

syncLimit=5

dataDir=/home/app/zookeeper/zkdata2

clientPort=2182

server.1=192.168.1.108:2888:3888

server.2=192.168.1.108:2889:3889

server.3=192.168.1.108:2890:3890

~ vi  /home/app/zookeeper/zookeeper-3.4.5/conf/zoo3.cfg

tickTime=2000

initLimit=10

syncLimit=5

dataDir=/home/app/zookeeper/zkdata3

clientPort=2183

server.1=192.168.1.108:2888:3888

server.2=192.168.1.108:2889:3889

server.3=192.168.1.108:2890:3890

3个节点的ZooKeeper集群配置完成，接下来我们的启动服务。

启动集群

~ /home/app/zookeeper/zookeeper-3.4.5/bin/zkServer.sh start zoo1.cfg

~ /home/app/zookeeper/zookeeper-3.4.5/bin/zkServer.sh start zoo2.cfg

~ /home/app/zookeeper/zookeeper-3.4.5/bin/zkServer.sh start zoo3.cfg

~ jps

5422 QuorumPeerMain

5395 QuorumPeerMain

5463 QuorumPeerMa

#查看节点状态

--如果在当前目录下无法查看，则要用全路径查看

~ /home/app/zookeeper/zookeeper-3.4.5/bin/zkServer.sh status zoo1.cfg

MX enabled by default

JUsing config: /home/app/zookeeper/zookeeper-3.4.5/bin/../conf/zoo1.cfg

Mode: follower

~ /home/app/zookeeper/zookeeper-3.4.5/bin/zkServer.sh status zoo2.cfg

MX enabled by default

JUsing config: /home/app/zookeeper/zookeeper-3.4.5/bin/../conf/zoo2.cfg

Mode: leader

~ /home/app/zookeeper/zookeeper-3.4.5/bin/zkServer.sh status zoo3.cfg

MX enabled by default

JUsing config: /home/app/zookeeper/zookeeper-3.4.5/bin/../conf/zoo3.cfg

Mode: follower

我们可以看到zoo2是leader，zoo1和zoo3是follower(主，接收和响应客户端的读写请求，向从改善数据)和follower（从leader同步数据和当leader服务停止后投票从新选择leader）

4. zookeeper完全分布式集群安装

所谓“伪分布式集群”就是在，在一台PC中，启动多个ZooKeeper的实例。“完全分布

式集群”是每台PC，启动一个ZooKeeper实例。我们现在有3台服务器，分别是：192.168.0.181，192.168.0.182，192.168.0.183。每台服务器创建相同的环境目录~ mkdir /home/app/zookeeper/zkdata

#新建myid文件

192.168.0.181服务器：

~ echo "1" > /home/app/zookeeper/zkdata/myid

192.168.0.182服务器：

~ echo "2" > /home/app/zookeeper/zkdata/myid

192.168.0.183服务器：

~ echo "3" > /home/app/zookeeper/zkdata/myid

分别修改配置文件

增加：集群的实例，server.X，”X”表示每个目录中的myid的值

一、192.168.0.181服务器上：

~ vi  /home/app/zookeeper/zookeeper-3.4.5/conf/zoo.cfg

tickTime=2000

initLimit=10

syncLimit=5

dataDir=/home/app/zookeeper/zkdata

#dataLogDir=/home/app/zookeeper/zkdataLog

clientPort=2181

server.1=192.168.0.181:2888:3888

server.2=192.168.0.182:2888:3888

server.3=192.168.0.183:2888:3888

二、192.168.0.182服务器上：

~ vi  /home/app/zookeeper/zookeeper-3.4.5/conf/zoo.cfg

tickTime=2000

initLimit=10

syncLimit=5

dataDir=/home/app/zookeeper/zkdata

#dataLogDir=/home/app/zookeeper/zkdataLog

clientPort=2181

server.1=192.168.0.181:2888:3888

server.2=192.168.0.182:2888:3888

server.3=192.168.0.183:2888:3888

三、192.168.0.183服务器上：

~ vi  /home/app/zookeeper/zookeeper-3.4.5/conf/zoo.cfg

tickTime=2000

initLimit=10

syncLimit=5

dataDir=/home/app/zookeeper/zkdata

#dataLogDir=/home/app/zookeeper/zkdataLog

clientPort=2181

server.1=192.168.0.181:2888:3888

server.2=192.168.0.182:2888:3888

server.3=192.168.0.183:2888:3888

启动和查看节点状态参照第3点伪分布式集群的启动和查看。