**Hadoop2.7.2及Hbase1.2.1安装部署**

**1. 安装配置Linux系统**

下载linux 系统（Red Hat、centos、ubuntu）版本，然后装机

**1.1关闭防火墙**

service iptables stop

chkconfig iptables off

**1.2 设置静态IP**

# vi /etc/sysconfig/network-scripts/ifcfg-eth0

HWADDR="00:15:5D:07:F1:02"

TYPE="Ethernet"

BOOTPROTO="static" #dhcp改为static

DEFROUTE="yes"

PEERDNS="yes" gate

PEERROUTES="yes"

IPV4\_FAILURE\_FATAL="no"

IPV6INIT="yes"

IPV6\_AUTOCONF="yes"

IPV6\_DEFROUTE="yes"

IPV6\_PEERDNS="yes"

IPV6\_PEERROUTES="yes"

IPV6\_FAILURE\_FATAL="no"

NAME="eth0"

UUID="bb3a302d-dc46-461a-881e-d46cafd0eb71"

ONBOOT="yes" #开机启用本配置

IPADDR=192.168.7.106 #静态IP

GATEWAY=192.168.7.1 #默认网关

NETMASK=255.255.255.0 #子网掩码

DNS1=192.168.7.1 #DNS 配置

重新启动网卡

service network restart

**1.3集群的物理配置情况**

集群IP: 192.168.77.241 ~ 192.168.77.243

(hostname分别为master、slave1、slave2)

用户名/密码：root/lanswon

其中，master作为jobtracker和namenode，其余节点作为tasktracker和datanode

**1.4 节点角色分配**

**1.4.1 修改hosts文件**

修改每个节点的 /etc/hosts文件，追加如下内容

192.168.77.241 master

192.168.77.242 slave1

192.168.77.243 slave2

**1.4.2 修改hostname**

修改每个节点自己的hostname，如对于master,将其/etc/sysconfig/network文件改为

    NETWORKING=yes

    HOSTNAME=master

然后，运行

    hostname master

退出，再进入输入env | grep hostname发现修改生效。

对于slave1,将其/etc/sysconfig/network文件改为

    NETWORKING=yes

    HOSTNAME=slave1

然后，运行

    hostname slave1

退出，再进入输入env | grep hostname发现修改生效。

对于slave2 同理设置。

**1.5 为每个节点设置无密码ssh登录**

1、 在master节点上运行

    ssh-keygen

2、 在master节点上，首先

    cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

然后

    chmod 644 ~/.ssh/authorized\_keys

3、将master节点上的~/.ssh/ authorized\_keys文件分别scp到每个slave节点上，如

    scp ~/.ssh/ authorized\_keys root@slave1:~/.ssh

4、 修改文件权限

    chmod 600 ~/.ssh/authorized\_keys

5、 从第3步开始循环，直至配置好每个slave节点

**1.6 下载Java**

下载 jdk-7u79-linux-x64.rpm，放在主节点 /usr/java目录下，用rpm –ivh jdk-7u79-linux-x64.rpm解压，在/usr/java下生成目录jdk1.7.0\_79。

将rpm安装文件传输给其他节点(其他节点需要有相应文件夹路径)

scp /usr/java/ jdk-7u79-linux-x64.rpm root@slave1:/usr/java/

scp /usr/java/ jdk-7u79-linux-x64.rpm root@slave2:/usr/java/

重复以上操作，完成slave1，slave2的java安装。

**2.Hadoop的配置**

**2.1 下载Hadoop**

下载hadoop-2.7.2，放在/home目录下，用命令 tar -xzvf hadoop-2.7.2.tar.gz解压，在/home下生成目录hadoop-2.7.2,修改文件夹名称 mv hadoop-2.7.2 hadoop

**2.2 修改Hadoop的配置文件**

修改/etc/hadoop/core-site.xml文件

|  |
| --- |
| <configuration>  <property>  <name>fs.defaultFS</name>  <value>hdfs://master:9000</value>  </property>  <property>  <name>hadoop.tmp.dir</name>  <value>file:/home/hadoop/tmp</value>  </property>  <property>  <name>io.file.buffer.size</name>  <value>131702</value>  </property>  </configuration> |

在/etc/hadoop/hadoop-env.sh，yarn-env.sh文件里增加

  export JAVA\_HOME=/usr/java/jdk1.7.0\_79

修改/etc/hadoop/hdfs-site.xml文件：

|  |
| --- |
| <configuration>  <property>  <name>dfs.namenode.name.dir</name>  <value>file:/home/hadoop/hdata/name</value>  </property>  <property>  <name>dfs.datanode.data.dir</name>  <value>file:/home/hadoop/hdata/data</value>  </property>  <property>  <name>dfs.replication</name>  <value>2</value>  </property>  <property>  <name>dfs.namenode.secondary.http-address</name>  <value>master:9001</value>  </property>  <property>  <name>dfs.webhdfs.enabled</name>  <value>true</value>  </property>  </configuration> |

在/home/hadoop目录下建立hdata文件夹，在/home/hadoop/hdata目录中创建文件夹name，data。

修改/etc/hadoop/mapred-site.xml文件：

|  |
| --- |
| <configuration>  <property>  <name>mapreduce.framework.name</name>  <value>yarn</value>  </property>  <property>  <name>mapreduce.jobhistory.address</name>  <value>master:10020</value>  </property>  <property>  <name>mapreduce.jobhistory.webapp.address</name>  <value>master:19888</value>  </property>  </configuration> |

修改/etc/hadoop/yarn-site.xml文件：

|  |
| --- |
| <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle</value>  </property>  <property>  <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>  <value>org.apache.hadoop.mapred.ShuffleHandler</value>  </property>  <property>  <name>yarn.resourcemanager.address</name>  <value>master:8032</value>  </property>  <property>  <name>yarn.resourcemanager.scheduler.address</name>  <value>master:8030</value>  </property>  <property>  <name>yarn.resourcemanager.resource-tracker.address</name>  <value>master:8031</value>  </property>  <property>  <name>yarn.resourcemanager.admin.address</name>  <value>master:8033</value>  </property>  <property>  <name>yarn.resourcemanager.webapp.address</name>  <value>master:8088</value>  </property>  <property>  <name>yarn.nodemanager.resource.memory-mb</name>  <value>1024</value>  </property> |

/etc/hadoop/slaves文件

|  |
| --- |
| slave1  slave2 |

替换/home/hadoop/lib文件夹下的native文件

从master节点上，将/home/hadoop目录依次拷贝到每个slave节点的/home目录下

scp –r /home/hadoop root@slave1:/home

scp –r /home/hadoop root@slave2:/home

**3. Zookeeper的配置**

安装HBase时，不仅需要HBase本身，还需要ZooKeeper来提供、管理和协调分布式服务。在运行HBase之前，也需要先启动ZooKeeper服务。

本次项目中，采用了Hadoop-2.7.2，HBase-1.2.2，以及ZooKeeper-3.4.6

由于ZooKeeper采用了Paxos算法来进行选举，因此实际能够有效参与选举的节点数量是奇数。所以，我们在安装HBase和ZooKeeper时，只采用奇数个节点（本文采用3个节点，分别为master、slave1、slave2）。

**3.1 安装ZooKeeper**

将zookeeper-3.4.6.tar.gz在/home下解压，生成zookeeper-3.4.6目录，修改目录名称zookeeper。在/home/zookeeper/conf中，有zoo\_sample.cfg文件，使用命令 cp zoo\_sample.cfg zoo.cfg 命令生成 zoo.cfg配置文件。编辑zoo.cfg的内容，如下：

|  |
| --- |
| # The number of milliseconds of each tick  tickTime=2000  # The number of ticks that the initial  # synchronization phase can take  initLimit=10  # The number of ticks that can pass between  # sending a request and getting an acknowledgement  syncLimit=5  # the directory where the snapshot is stored.  # do not use /tmp for storage, /tmp here is just  # example sakes.  dataDir=/home/zookeeper/zkdata  dataLogDir=/home/zookeeper/logs  # the port at which the clients will connect  clientPort=2181  server.1=master:2887:3887  server.2=slave1:2888:3888  server.3=slave2:2889:3889  # the maximum number of client connections.  # increase this if you need to handle more clients  #maxClientCnxns=60  #  # Be sure to read the maintenance section of the  # administrator guide before turning on autopurge.  #  # http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc\_maintenance  #  # The number of snapshots to retain in dataDir  #autopurge.snapRetainCount=3  # Purge task interval in hours  # Set to "0" to disable auto purge feature  #autopurge.purgeInterval=1 |

在/home/zookeeper目录下建立zkdata，logs文件夹。

从master节点上，将/home/zookeeper目录依次拷贝到每个slave节点的/home目录下

scp –r /home/zookeeper root@slave1:/home

scp –r /home/zookeeper root@slave2:/home

为了让系统能够识别zoo.cfg中的server.id=host:port:port中的server.id，在master节点的/home/zookeeper/zkdata中创建一个文件myid，里面只有一个数字1；同理，在slave1节点的/home/zookeeper/zkdata中创建一个文件myid，里面只有一个数字2；以此类推…. 直至slave2节点。注意，这里id的范围是1~255 。以上配置（zookeeper）要复制到集群中的每一个节点上去。

**4. HBase配置**

**4.1 下载hbase**

下载hbase-1.2.2-bin.tar.gz 将/home/hbase-1.2.2-bin.tar.gz解压，

tar -xzvf hbase-1.2.2-bin.tar.gz

生成/home/hbase-1.2.2。修改文件夹名称 mv hbase-1.2.2 hbase，

在hbase/lib中，查看lib的hadoop的版本的jar包是否与安装hadoop的版本一致，

find -name ‘hadoop\*jar

如果不一致，一定要替换为hadoop下的jar包版本。

创建f.sh文件

|  |
| --- |
| find -name "hadoop\*jar" | sed 's/2.5.1/2.7.2/g' | sed 's/\.\///g' > f.log  rm ./hadoop\*jar  cat ./f.log | while read Line  do  find /home/hadoop -name "$Line" | xargs -i cp {} ./  done  rm ./f.log |

给文件授权，可以执行的权限

chmod u+x f.sh

执行文件

./f.sh

jar 包替换成功；/home/hbase/lib 目录下还有个 slf4j-log4j12-XXX.jar，在机器有装hadoop时，由于classpath中会有hadoop中的这个jar包，会有冲突，直接删除掉

rm –f slf4j-log4j12-1.7.5.jar

**4.2 配置hbase文件**  
打开conf/hbase-env.sh文件，文件里增加

  export JAVA\_HOME=/usr/java/jdk1.7.0\_79

export HBASE\_CLASSPATH=/home/hadoop/etc/hadoop        

  export  HBASE\_MANAGES\_ZK=false

第一个参数指定了JDK路径；

第二个参数指定了 hadoop 的配置文件路径；

第三个参数设置使用 hbase 默认自带的 Zookeeper;

这表示我们不使用hbase-1.2.2自带的zookeeper，而是使用我们刚才自己安装的zookeeper-3.4.6（具体指定使用哪个zookeeper是通过/etc/profile中的ZOOKEEPER\_HOME变量来指定的）。

下面，打开conf/hbase-site.xml文件，添加以下内容:

|  |
| --- |
| <configuration>  <property>  <name>hbase.rootdir</name>  <value>hdfs://master:9000/hbase</value>  </property>  <property>  <name>hbase.cluster.distributed</name>  <value>true</value>  </property>  <property>  <name>dfs.replication</name>  <value>2</value>  </property>  <property>  <name>hbase.tmp.dir</name>  <value>/home/hbase/tmp</value>  </property>  <property>  <name>hbase.zookeeper.quorum</name>  <value>master,slave1,slave2</value>  </property>  <property>  <name>hbase.zookeeper.property.dataDir</name>  <value>/home/zookeeper/zkdata</value>  </property>  </configuration> |

其中，通过hbase.rootdir指定HDFS的一个namenode节点，并让HBase将数据写入HDFS中的一个目录（本例中就是hdfs://master:9000/hbase）；hbase.zookeeper.quorum指定了哪些节点运行Quorum Peer（即QuorumPeerMain进程），个数需为奇数。

下面，打开文件conf/regionservers，添加以下内容：

|  |
| --- |
| slave1  slave2 |

该文件指定了HRegionServer进程将在哪些节点上运行。

从master节点上，将/home/hbase目录依次拷贝到每个slave节点的/home目录下

scp –r /home/hbase root@slave1:/home

scp –r /home/hbase root@slave2:/home

**5. 启动配置**

**5.1 配置profile文件**

在/etc/profile的最后添加：

|  |
| --- |
| # set path  export JAVA\_HOME=/usr/java/jdk1.7.0\_79  export HADOOP\_HOME=/home/hadoop  export HBASE\_HOME=/home/hbase  export ZOOKEEPER\_HOME=/home/zookeeper  export PATH=$JAVA\_HOME/bin:$HADOOP\_HOME/bin:$HBASE\_HOME/bin:$ZOOKEEPER\_HOME/bin:$ZOOKEEPER\_HOME/conf:$PATH |

然后，将/etc/profile文件 传到slave1,slave2节点上

scp /etc/profile root@slave1:/etc/

scp /etc/profile root@slave2:/etc/

然后，source /etc/profile

这个操作要在集群的每一个节点上进行。

**5.2集群启动顺序**

Hadoop、ZooKeeper和HBase之间应该按照顺序启动和关闭：启动Hadoop—>启动ZooKeeper集群—>启动HBase—>停止HBase—>停止ZooKeeper集群—>停止Hadoop。

**5.3 启动Hadoop**

在master节点的/home/hadoop目录下进行HDFS的格式化

      bin/hdfs namenode –format

格式化后，HDFS中的内容会全部丢失

然后，开启Hadoop集群  
      sbin/start-all.sh

此时，Hadoop集群已经启动，可以提交运算的作业了。

http://192.168.33.50:8088

http://192.168.33.50:50070

**5.4 启动ZooKeeper**

在每一个节点的/home/zookeeper/bin中执行命令：zkServer.sh  start

这样就启动了zookeeper集群中的各个节点。

输入命令./zkServer.sh  status

可以查看当前节点在zookeeper集群中的角色是leader还是follower。

若启动出现问题，可以使用调试命令

zkServer.sh  start-foreground

**5.4 启动HBase**

启动HBase。进入/home/hbase/bin目录下，输入命令  
        start-hbase.sh

在HBase集群启动之后，键入命令

        hbase  shell

启动HBase的控制台界面

在浏览器中查看HBase系统的运行状态，地址为http://192.168.33.50:16010 /master-status，其中192.168.33.50是master所在节点的IP。