**hadoop2.0集群搭建详解**

hadoop2.2.0集群搭建

    PS：apache提供的hadoop-2.2.0的安装包是在32位操作系统编译的，

           因为hadoop依赖一些C++的本地库，所以如果在64位的操作上安装

           hadoop-2.2.0就需要重新在64操作系统上重新编译

1.准备工作:(参考伪分布式搭建)

        1.1修改Linux主机名

        1.2修改IP

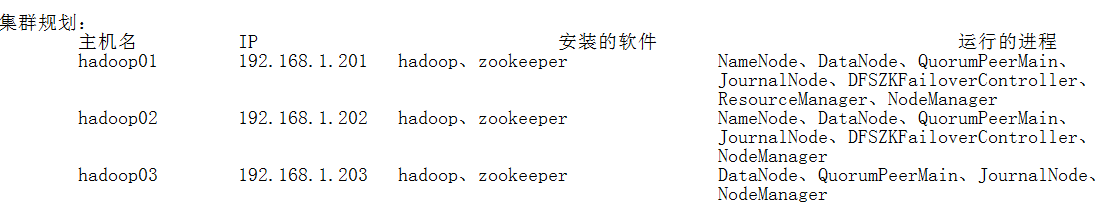
        1.3修改主机名和IP的映射关系

        1.4关闭防火墙

        1.5ssh免登陆

        1.6.安装JDK，配置环境变量等

2.集群规划：

[](http://static.oschina.net/uploads/space/2014/0331/001422_OQrF_940266.png)

            PS：

             在hadoop2.0中通常由两个NameNode组成，一个处于active状态，

                    另一个处于standby状态。Active NameNode对外提供服务，

                    而Standby NameNode则不对外提供服务，仅同步active namenode

                    的状态，以便能够在它失败时快速进行切换。

             hadoop2.0官方提供了两种HDFS HA的解决方案，一种是NFS，

                    另一种是QJM。这里我们使用简单的QJM。在该方案中，

                    主备NameNode之间通过一组JournalNode同步元数据信息，

                    一条数据只要成功写入多数JournalNode即认为写入成功。

                    通常配置奇数个JournalNode

             这里还配置了一个zookeeper集群，用于ZKFC

                    （DFSZKFailoverController）故障转移，当Active NameNode挂

                     掉了，会自动切换Standby NameNode为standby状态

3.安装步骤：

     3.1.安装配置zooekeeper集群

         3.1.1解压

         tar -zxvf zookeeper-3.4.5.tar.gz -C /cloud/

     3.1.2修改配置

         cd /cloud/zookeeper-3.4.5/conf/

         cp zoo\_sample.cfg zoo.cfg

         vim zoo.cfg

         修改：dataDir=/cloud/zookeeper-3.4.5/tmp

         在最后添加：

             server.1=hadoop01:2888:3888

             server.2=hadoop02:2888:3888

             server.3=hadoop03:2888:3888

         保存退出

         然后创建一个tmp文件夹

             mkdir /cloud/zookeeper-3.4.5/tmp

         再创建一个空文件

             touch /cloud/zookeeper-3.4.5/tmp/myid

         最后向该文件写入ID

             echo 1 > /cloud/zookeeper-3.4.5/tmp/myid

     3.1.3将配置好的zookeeper拷贝到其他节点(首先分别在hadoop02、

            hadoop03根目录下创建一个cloud目录：mkdir /cloud)

             scp -r /cloud/zookeeper-3.4.5/ hadoop02:/cloud/

             scp -r /cloud/zookeeper-3.4.5/ hadoop03:/cloud/

     注意：修改hadoop02、hadoop03对应

               /cloud/zookeeper-3.4.5/tmp/myid内容

         hadoop02：

             echo 2 > /cloud/zookeeper-3.4.5/tmp/myid

         hadoop03：

             echo 3 > /cloud/zookeeper-3.4.5/tmp/myid

     3.2.安装配置hadoop集群

         3.2.1解压

         tar -zxvf hadoop-2.2.0.tar.gz -C /cloud/

     3.2.2配置HDFS（hadoop2.0所有的配置文件都在

            $HADOOP\_HOME/etc/hadoop目录下）

     将hadoop添加到环境变量中

         vim /etc/profile

         export JAVA\_HOME=/usr/java/jdk1.6.0\_45

         export HADOOP\_HOME=/cloud/hadoop-2.2.0

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

         cd /cloud/hadoop-2.2.0/etc/hadoop

     3.2.2.1修改hadoo-env.sh

         export JAVA\_HOME=/usr/java/jdk1.6.0\_45

3,2.2.2修改core-site.xml

     <configuration>

         <!-- 指定hdfs的nameservice为ns1 -->

         <property>

             <name>fs.defaultFS</name>

             <value>hdfs://ns1</value>

         </property>

     <!-- 指定hadoop临时目录 -->

         <property>

             <name>hadoop.tmp.dir</name>

             <value>/cloud/hadoop-2.2.0/tmp</value>

         </property>

         <!-- 指定zookeeper地址 -->

         <property>

             <name>ha.zookeeper.quorum</name>

             <value>

                 hadoop01:2181,hadoop02:2181,hadoop03:2181

            </value>

         </property>

     </configuration>

3,2.2.3修改hdfs-site.xml

     <configuration>

         <!--指定hdfs的nameservice为ns1，

         需要和core-site.xml中的保持一致 -->

         <property>

             <name>dfs.nameservices</name>

             <value>ns1</value>

         </property>

         <!-- ns1下面有两个NameNode，分别是nn1，nn2 -->

         <property>

             <name>dfs.ha.namenodes.ns1</name>

             <value>nn1,nn2</value>

         </property>

         <!-- nn1的RPC通信地址 -->

         <property>

             <name>

                 dfs.namenode.rpc-address.ns1.nn1

             </name>

             <value>hadoop01:9000</value>

         </property>

         <!-- nn1的http通信地址 -->

         <property>

             <name>

                 dfs.namenode.http-address.ns1.nn1

             </name>

             <value>hadoop01:50070</value>

         </property>

         <!-- nn2的RPC通信地址 -->

         <property>

             <name>

                 dfs.namenode.rpc-address.ns1.nn2

             </name>

             <value>hadoop02:9000</value>

         </property>

         <!-- nn2的http通信地址 -->

         <property>

             <name>

                 dfs.namenode.http-address.ns1.nn2

             </name>

             <value>hadoop02:50070</value>

         </property>

         <!-- 指定NameNode的元数据在

                JournalNode上的存放位置 -->

         <property>

             <name>

                 dfs.namenode.shared.edits.dir

            </name>

             <value>

                 qjournal://hadoop01:8485;hadoop02:8485;hadoop03:8485/ns1

             </value>

         </property>

         <!-- 指定JournalNode在本地磁盘存放数据的位置 -->

         <property>

             <name>dfs.journalnode.edits.dir</name>

             <value>/cloud/hadoop-2.2.0/journal</value>

         </property>

         <!-- 开启NameNode失败自动切换 -->

         <property>

             <name>

                 dfs.ha.automatic-failover.enabled

              </name>

             <value>true</value>

         </property>

         <!-- 配置失败自动切换实现方式 -->

         <property>

             <name>

                 dfs.client.failover.proxy.provider.ns1

             </name>

             <value>

                org.apache.hadoop.hdfs.server.namenode.ha.

             ConfiguredFailoverProxyProvider

            </value>

         </property>

         <!-- 配置隔离机制 -->

         <property>

             <name>dfs.ha.fencing.methods</name>

             <value>sshfence</value>

         </property>

         <!-- 使用隔离机制时需要ssh免登陆 -->

         <property>

             <name>

                 dfs.ha.fencing.ssh.private-key-files

             </name>

             <value>/root/.ssh/id\_rsa</value>

         </property>

     </configuration>

     3.2.2.4修改slaves

     hadoop01

     hadoop02

     hadoop03

     3.2.3配置YARN

         3.2.3.1修改yarn-site.xml

     <configuration>

         <!-- 指定resourcemanager地址 -->

         <property>

             <name>

                 yarn.resourcemanager.hostname

             </name>

             <value>hadoop01</value>

         </property>

         <!-- 指定nodemanager启动时加载

                 server的方式为shuffle server -->

         <property>

             <name>

                 yarn.nodemanager.aux-services

             </name>

             <value>mapreduce\_shuffle</value>

         </property>

     </configuration>

     3.2.3.2修改mapred-site.xml

         <configuration>

             <!-- 指定mr框架为yarn方式 -->

             <property>

                 <name>

                      mapreduce.framework.name

                  </name>

                 <value>yarn</value>

             </property>

         </configuration>

     3.2.4将配置好的hadoop拷贝到其他节点

         scp -r /cloud/hadoop-2.2.0/ hadoo02:/cloud/

         scp -r /cloud/hadoop-2.2.0/ hadoo03:/cloud/

     3.2.5启动zookeeper集群

           （分别在hadoop01、hadoop02、hadoop03上启动zk）

         cd /cloud/zookeeper-3.4.5/bin/

         ./zkServer.sh start

         查看状态：

         ./zkServer.sh status

         （一个leader，两个follower）

     3.2.6启动journalnode（在hadoop01上启动所有journalnode）

         cd /cloud/hadoop-2.2.0

         sbin/hadoop-daemons.sh start journalnode

         (运行jps命令检验，多了JournalNode进程)

     3.2.7格式化HDFS

         在hadoop01上执行命令:

         hadoop namenode -format

         格式化后会在根据core-site.xml中的hadoop.tmp.dir

         配置生成个文件，这里我配置的是/cloud/hadoop-2.2.0/tmp，

         然后将/cloud/hadoop-2.2.0/tmp拷贝到hadoop02的

         /cloud/hadoop-2.2.0/下。

         scp -r tmp/ hadoop02:/cloud/hadoop-2.2.0/

     3.2.8格式化ZK(在hadoop01上执行即可)

         hdfs zkfc -formatZK

     3.2.9启动HDFS(在hadoop01上执行)

         sbin/start-dfs.sh

     3.3.10启动YARN(在hadoop01上执行)

         sbin/start-yarn.sh

    到此，hadoop2.2.0配置完毕，可以统计浏览器访问:

         http://192.168.1.201:50070

         NameNode 'hadoop01:9000' (active)

         http://192.168.1.202:50070

         NameNode 'hadoop02:9000' (standby)

    验证HDFS HA

         首先向hdfs上传一个文件

             hadoop fs -put /etc/profile /profile

             hadoop fs -ls /

         然后再kill掉active的NameNode

             kill -9 <pid of NN>

         通过浏览器访问：http://192.168.1.202:50070

             NameNode 'hadoop02:9000' (active)

             这个时候hadoop02上的NameNode变成了active

         在执行命令：

             hadoop fs -ls /

             -rw-r--r--   3 root supergroup       1926 2014-02-06 15:36 /profile

     刚才上传的文件依然存在！！！

         手动启动那个挂掉的NameNode

         sbin/hadoop-daemon.sh start namenode

         通过浏览器访问：http://192.168.1.201:50070

         NameNode 'hadoop01:9000' (standby)

    验证YARN：

         运行一下hadoop提供的demo中的WordCount程序：

             hadoop jar share/hadoop/mapreduce/hadoop-mapreduce

            -examples-2.2.0.jar wordcount /profile /out

OK，大功告成！！！