# Hadoop-Mahout配置

## 文件准备

Hadoop2.7安装包下载：<http://mirror.bit.edu.cn/apache/hadoop/common/>

Mahout 安装包下载：<http://archive.apache.org/dist/mahout/>

JDK1.8

## 集群ssh无密匙登录

首先输入ssh localhost,验证在为配置前是无法通过ssh连接本机的

ssh localhost

不能免密继续：

ssh-keygen -t dsa -P '' -f /root/.ssh/id\_dsa

ssh-keygen代表生成密钥;-t表示生成密钥的类型;-P提供密语；-f指定生成的文件.这个命令执行完毕后会在.ssh文件夹下生成两个文件，分别是id\_dsa、id\_dsa.pub,这是SSH的一对私钥和公钥，就像是钥匙和锁。

下一步将id\_dsa.pub追加到授权的key中,键入一下命令：

cat /root/.ssh/id\_dsa.pub >> /root/.ssh/authorized\_keys

此时，免密码登录本机就配置完成了

### 将本机的SSH公钥copy到其他三台虚拟机上并输入相应虚拟机的的密码

ssh-copy-id -i /root/.ssh/id\_dsa.pub root@hadoop.slave1

#提示输入hadoop.slave1的密码

## Hadoop集群安装

解压hadoop安装包

### 环境变量配置

export HADOOP\_HOME=/usr/local/hadoop/hadoop-2.7.0

export HADOOP\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

export YARN\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

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

修改后保存执行

source /etc/profile

### Hadoop参数配置

目录$HADOOP\_HOME/etc/hadoop下文件配置：

#### Slaves

集群节点的hostname

#### core-site.xml

<property>

<name>fs.default.name</name>

<value>hdfs://nn00.docker.wd.com:9000</value>

<final>true</final>

</property>

<property>

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

<value>/home/hadoop/data/hadoop\_${user.name}</value>

</property>

<property>

<name>io.file.buffer.size</name>

<value>131072</value>

</property>

#### hdfs-site.xml

<property>

<name>dfs.namenode.secondary.http-address</name>

<value>nn00.docker.wd.com:50090</value>

</property>

<property>

<name>dfs.name.dir</name>

<value>/usr/local/hadoop/name</value>

<final>true</final>

</property>

<property>

<name>dfs.data.dir</name>

<value>/usr/local/hadoop/data</value>

<final>true</final>

</property>

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

<property>

<name>dfs.block.size</name>

<value>134217728</value>

</property>

<property>

<name>dfs.namenode.handler.count</name>

<value>40</value>

</property>

<!--The number of server threads for the datanode -->

<property>

<name>dfs.datanode.handler.count</name>

<value>64</value>

</property>

#### yarn-site.xml

<!-- Site specific YARN configuration properties -->

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.resourcemanager.address</name>

<value>nn00.docker.wd.com:8032</value>

</property>

<property>

<name>yarn.resourcemanager.scheduler.address</name>

<value>nn00.docker.wd.com:8030</value>

</property>

<property>

<name>yarn.resourcemanager.resource-tracker.address</name>

<value>nn00.docker.wd.com:8031</value>

</property>

<property>

<name>yarn.resourcemanager.admin.address</name>

<value>nn00.docker.wd.com:8033</value>

</property>

<property>

<name>yarn.resourcemanager.webapp.address</name>

<value>nn00.docker.wd.com:8088</value>

</property>

<property>

<name>yarn.nodemanager.vmem-check-enabled</name>

<value>false</value>

<description>Whether virtual memory limits will be enforced for containers</description>

</property>

<property>

<name>yarn.nodemanager.vmem-pmem-ratio</name>

<value>4</value>

<description>Ratio between virtual memory to physical memory when setting memory limits for containers</description>

</property>

<property>

<name>yarn.resourcemanager.hostname</name>

<value>nn00.docker.wd.com</value>

</property>

#### mapred-site.xml<1000W人数配置>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

<!--themost important properties for jobhistory -->

<property>

<name>mapreduce.jobhistory.address</name>

<value>192.168.1.47:10020</value>

</property>

<property>

<name>mapreduce.jobhistory.webapp.address</name>

<value>192.168.1.47:19888</value>

</property>

<property>

<name>mapred.job.tracker</name>

<value>192.168.1.47:9001</value>

</property>

<property>

<name>mapred.tasktracker.map.tasks.maximum</name>

<value>16</value>

</property>

<property>

<name>mapred.tasktracker.reduce.tasks.maximum</name>

<value>8</value>

</property>

<!--property>

<name>mapred.child.java.opts</name>

<value>-Xmx1024M</value>

</property-->

<property>

<name>mapred.map.child.java.opts</name>

<value>-Xmx1024m</value>

</property>

<property>

<name>mapred.reduce.child.java.opts</name>

<value>-Xmx2048m</value>

</property>

<property>

<name>mapred.reduce.tasks</name>

<value>8</value>

</property>

<property>

<name>mapred.map.tasks</name>

<value>16</value>

</property>

<property>

<name>io.sort.mb</name>

<value>200</value>

</property>

<property>

<name>io.sort.factor</name>

<value>100</value>

</property>

<property>

<name>mapred.compress.map.output</name>

<value>true</value>

</property>

<property>

<name>mapred.reduce.parallel.copies</name>

<value>10</value>

</property>

<property>

<name>mapred.job.reuse.jvm.num.tasks</name>

<value>20</value>

</property>

<property>

<name>mapreduce.map.memory.mb</name>

<value>2048</value>

</property>

<property>

<name>mapreduce.reduce.memory.mb</name>

<value>3072</value>

</property>

<!--property>

<name>mapreduce.jobtracker.handler.count</name>

<value>128</value>

</property>

<property>

<name>yarn.app.mapreduce.am.resource.mb</name>

<value>3072</value>

</property-->

<property>

<name>yarn.nodemanager.resource.memory-mb</name>

<value>30720</value>

</property>

<property>

<name>yarn.nodemanager.resource.cpu-vcores</name>

<value>24</value>

</property>

## Mahout安装

解压mahout安装包

配置环境变量：

export MAHOUT\_HOME=/usr/local/mahout/apache-mahout-distribution-0.13.0

export MAHOUT\_CONF\_DIR=/usr/local/mahout/apache-mahout-distribution-0.13.0/conf

export PATH=$PATH:$MAHOUT\_HOME/bin

保存后执行

source /etc/profile

## 启动Hadoop,执行命令

### 格式化节点

hdfs namenode –format

### 启动集群

sh $HADOOP\_HOME/sbin/start-all.sh

运行指令jps查看运行状态

访问8088端口地址查看页面

### Mahout执行指令

创建hdfs文件目录： hadoop fs –mkdir 目录

将目标文件放入：hdfs中 hadoop fs –put 目标文件 hdfs目录

执行mahout运算 ：mahout recommenditembased --booleanData true --numRecommendations 50 --input hdfs文件 --output output目录 --tempDir temp目录 --similarityClassname SIMILARITY\_COOCCURRENCE

运算结束传出文件：hadoop fs –getmerge output目录 输出文件