学习总结

1812975 曹沛鑫 2020.6.24

# 一、CentOS7下Hadoop3.2.1的安装与部署（单机模式）

1.安装准备

·下载Hadoop安装包：**hadoop-3.2.1.tar.gz**

·下载JDK：**jdk-8u211-linux-x64.tar.gz**

**·修改HOSTS配置： sudo vi /etc/hosts**

**## 在文件末尾添加一行**

**127.0.0.1 hadoop001**

**·安装JDK：将安装包解压至安装目录：注意使用压缩包的绝对路径**

**·**执行命令 vi ~/.bash\_profile 在~/.bash\_profile文件中增加Java环境变量：

export JAVA\_HOME=/usr/local/jdk1.8.0\_211

export JAVA\_BIN=$JAVA\_HOME/bin

export JAVA\_LIB=$JAVA\_HOME/lib

export CLASSPATH=.:$JAVA\_LIB/tools.jar:$JAVA\_LIB/dt.jar

PATH=$PATH:$HOME/.local/bin:$HOME/bin:$JAVA\_BIN

export PATH

（此处需要注意的事，由于需要先卸载系统自带的openjdk1.7，然后再安装jdk1.8）

·执行 source ~/.bash\_profile 命令使新添加的环境变量生效

·输入 java -version 命令查看JDK是否安装成功。

·安装Hadoop：将hadoop-3.2.1.tar.gz 解压至安装目录。接下来，编辑 ~/.bash\_profile 配置文件，增加 Hadoop 相关用户环境变量，添加后bash\_profile文件完整内容如下：

export JAVA\_HOME=/usr/local/jdk1.8.0\_211

export JAVA\_BIN=$JAVA\_HOME/bin

export JAVA\_LIB=$JAVA\_HOME/lib

export CLASSPATH=.:$JAVA\_LIB/tools.jar:$JAVA\_LIB/dt.jar  
export HADOOP\_HOME=/usr/local/hadoop-3.2.1

PATH=$PATH:$HOME/.local/bin:$HOME/bin:$JAVA\_BIN:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin  
export PATH

1. 修改配置文件（ %HADOOP\_HOME%/etc/hadoop/ 目录下）

## ·hadoop-env.sh：# The java implementation to use.

## export JAVA\_HOME=/usr/local/jdk1.8.0\_211

## # Location of Hadoop.

## export HADOOP\_HOME=/usr/local/hadoop-3.2.1

## ·**mapred-site.xml：**

<configuration>

<property>

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

<value>yarn</value>

<final>true</final>

<description>The runtime framework for executing MapReduce jobs</description>

</property>

<property>

<name>yarn.app.mapreduce.am.env</name>

<value>HADOOP\_MAPRED\_HOME=/usr/local/hadoop-3.2.1</value>

</property>

<property>

<name>mapreduce.map.env</name>

<value>HADOOP\_MAPRED\_HOME=/usr/local/hadoop-3.2.1</value>

</property>

<property>

<name>mapreduce.reduce.env</name>

<value>HADOOP\_MAPRED\_HOME=/usr/local/hadoop-3.2.1</value>

</property>

</configuration>

## ·**core-site.xml：**

<configuration>

<property>

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

<value>hdfs://hadoop001:9001</value>

</property>

</configuration>

## ·**hdfs-site.xml：**

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

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

<value>/home/hadoop/dfs/namenode</value>

</property>

<property>

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

<value>/home/hadoop/dfs/datanode</value>

</property>

<property>

<name>dfs.http.address</name>

<value>0.0.0.0:50070</value>

</property>

</configuration>

## **yarn-site.xml:**

<configuration>

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

<property>

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

<value>mapreduce\_shuffle</value>

<final>true</final>

</property>

</configuration>

## **·workers：**清空文件内容，仅保留 HOSTS 中配置的 hadoop001 主机

# 3.**启动Hadoop**

## ·**格式化NameNode：**hdfs namenode -format

## ·**启动NameNode：**hdfs --daemon start namenode

## ·**启动DataNode：**hdfs --daemon start datanode

·NameNode和DataNode启动之后，在浏览器中输入 [http://ip(该节点的ip，命令ifconfig可查看):50070](http://172.16.250.234:50070/) 访问HDFS的管理界面

**4.本机SSH免密码登陆配置**

·SELinux：修改 /etc/selinux/config 文件中的 SELINUX=enforcing 修改为 SELINUX=disabled ，然后重启

·防火墙：开启：systemctl start firewalld.service

关闭：systemctl stop firewalld.service

·配置sshd：编辑 /etc/ssh/sshd\_config 文件，使用命令：vi /etc/ssh/sshd\_config ，去掉以下3行的 “#” 注释：

RSAAuthentication yes

PubkeyAuthentication yes

AuthorizedKeysFile      .ssh/authorized\_keys

重启 sshd 服务，使用命令：systemctl restart sshd.service

·生成密钥：从 root 用户切换到hadoop 用户，可使用命令：su hadoop。然后， 执行命令 ssh-keygen -t rsa 来生成秘钥。无需指定口令密码，遇到询问直接回车，命令执行完毕后会在 hadoop 用户的家目录中（/home/hadoop/.ssh）生成两个文件：id\_rsa(私钥）和id\_rsa.pub（公钥）

·将公钥导入认证文件：cat /home/george/.ssh/id\_rsa.pub >> /home/george/.ssh/authorized\_keys

·设置认证文件访问权限：chmod 700 /home/george/.ssh

chmod 600 /home/george/.ssh/authorized\_keys

5.启动Yarn：start-yarn.sh。启动成功之后，在浏览器中输入 http://ip:8088 便可以查看YARN的管理界面了

二、wordcount测试

1.Hadoop3.2.1版本中etc/hadoop 目录下有一个 shellprofile.d 文件夹会导致测试失败，执行测试前记得删掉这个文件夹

2.输入命令：hdfs dfs -mkdir /input *# 在HDFS根目录下创建input目录*

3.输入命令：hdfs dfs -mkdir /output *# 在HDFS根目录下创建output目录*

4.输入命令：hdfs dfs -ls / *# 查看HDFS根目录下文件列表*

5.输入命令：hdfs dfs -put /usr/local/hadoop-3.2.1/etc/hadoop /input*# 将本地主机上的 hadoop 文件夹上传到 HDFS的 input目录*

6.输入命令：dfs dfs -ls /input/hadoop *# 查看HDFS中/input/hadoop目录下文件列表*

7.输入命令：hadoop jar /usr/local/hadoop-3.2.1/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.1.jar grep /input/hadoop /output/demo 'dfs[a-z.]+' *# 使用 hadoop 提供的mapreduce示例程序，统计上述32个文件中匹配'dfs[a-z.]+'的字符串出现的次数*

8.输入命令：hdfs dfs -ls /output/demo *# 查看结果存放目录/output/demo 下的文件列表*

9.输入命令：hdfs dfs -get /output/demo /tmp/ *# 将结果文件目录从HDFS拷贝到本地主机的 /tmp/ 目录下*

10.输入命令：cat /tmp/demo/part-r-00000 #查看统计结果，显示共31个文件目录

三、MapReduce简介

·定义combiner方法可优化reduce效率，但适用范围具有局限性，如可用在max方法，不能用在mean方法

1. 总结
2. 在文件格式化时，我看到shutdown就以为格式化失败，查了一下午bug，结果发现可以直接启动Hadoop成功，我太蠢了，要是敢试试就不用浪费这么多时间了。
3. 进一步巩固了linux命令
4. 怀疑伪分布式和集群全分布式不能完成，因为一旦同时开启多个虚拟机就会卡死，可以尝试小组内成员共同完成。
5. 不能看懂Java MapReducer程序