

• Hadoop集群模式安装

实验环境

Linux Ubuntu 16.04

实验内容

在Linux系统的服务器上，安装Hadoop3.0.0集群模式。

```
[root@Aliyun ~]# cd /data/hadoop/  
[root@Aliyun hadoop]# ls  
hadoop-3.0.0.tar.gz  jdk-8u161-linux-x64.tar.gz
```

• 1.安装jdk

将/data/hadoop目录下jdk-8u161-linux-x64.tar.gz 解压缩到/opt目录下。

```
1 | sudo tar -xzf /data/hadoop/jdk-8u161-linux-x64.tar.gz -C /opt
```

其中，tar -xzf 对文件进行解压缩，-C 指定解压后，将文件放到/opt目录下。

下面将jdk1.8.0_161目录重命名为java，执行：

```
1 | sudo mv /opt/jdk1.8.0_161/ /opt/java
```

修改java目录的所属用户和所属组：

```
1 | sudo chown -R dolphin.dolphin /opt/java
```

• 2.下面来修改环境变量

```
#java  
export JAVA_HOME=/opt/java  
export PATH=$JAVA_HOME/bin:$PATH  
#hadoop  
export HADOOP_HOME=/opt/hadoop  
export PATH=$HADOOP_HOME/bin:$PATH  
#nginx  
export NGINX_HOME=/usr/local/nginx/  
export PATH=$NGINX_HOME/sbin:$PATH
```

```
1 | sudo leafpad /etc/profile
```

末端添加如下内容：

```
1 #java
2 export JAVA_HOME=/opt/java
3 export PATH=$JAVA_HOME/bin:$PATH
```

保存并关闭编辑器

让环境变量生效。

```
1 source /etc/profile
```

刷新环境变量后，可以通过java的家目录找到java可使用的命令。利用java查看版本号命令验证是否安装成功：

```
1 java -version
```

正常结果显示如下

```
[root@Aliyun hadoop]# java -version
java version "1.8.0_161"
Java(TM) SE Runtime Environment (build 1.8.0_161-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.161-b12, mixed mode)
```

```
1 java version "1.8.0_161"
2 Java(TM) SE Runtime Environment (build 1.8.0_161-b12)
3 Java HotSpot(TM) 64-Bit Server VM (build 25.161-b12, mixed mode)
```

• 3.安装hadoop

将hadoop-3.0.0.tar.gz解压缩到/opt目录下。

```
1 sudo tar -xzf /data/hadoop/hadoop-3.0.0.tar.gz -C /opt/
```

为了便于操作，我们也将hadoop-3.0.0重命名为hadoop。

```
1 sudo mv /opt/hadoop-3.0.0/ /opt/hadoop
```

修改hadoop目录的所属用户和所属组：

```
1 sudo chown -R dolphin.dolphin /opt/hadoop
```

• 4.下面来修改环境变量

```
1 sudo leafpad /etc/profile
```

末端添加如下内容：

```
1 #hadoop
2 export HADOOP_HOME=/opt/hadoop
3 export PATH=$HADOOP_HOME/bin:$PATH
```

保存并关闭编辑器

让环境变量生效。

```
1 | source /etc/profile
```

利用hadoop查看版本号命令验证是否安装成功：

```
1 | hadoop version
```

正常结果显示如下

```
[root@Aliyun hadoop]# hadoop version
Hadoop 3.0.0
Source code repository https://git-wip-us.apache.org/repos/asf/hadoop.git -r c25427ceca461ee979d30edd7a4b0f50718e6533
Compiled by andrew on 2017-12-08T19:16Z
Compiled with protoc 2.5.0
From source with checksum 397832cb5529187dc8cd74ad54ff22
This command was run using /opt/hadoop/share/hadoop/common/hadoop-common-3.0.0.jar
```

```
1 | Hadoop 3.0.0
2 | Source code repository https://git-wip-us.apache.org/repos/asf/hadoop.git
  | -r c25427ceca461ee979d30edd7a4b0f50718e6533
3 | Compiled by andrew on 2017-12-08T19:16Z
4 | Compiled with protoc 2.5.0
5 | From source with checksum 397832cb5529187dc8cd74ad54ff22
6 | This command was run using /opt/hadoop/share/hadoop/common/hadoop-common-
  | 3.0.0.jar
```

- **5.修改hadoop hadoop-env.sh文件配置**

```
1 | leafpad /opt/hadoop/etc/hadoop/hadoop-env.sh
```

末端添加如下内容：

```
1 | export JAVA_HOME=/opt/java/
```

保存并关闭编辑器

- **6.修改hadoop core-site.xml文件配置**

```
1 | leafpad /opt/hadoop/etc/hadoop/core-site.xml
```

添加下面配置到

与

标签之间。

```
1 | <property>
2 |     <name>fs.defaultFS</name>
3 |     <value>hdfs://master:9000</value>
4 | </property>
```

保存并关闭编辑器

- **7.修改hadoop hdfs-site.xml文件配置**

```
1 | leafpad /opt/hadoop/etc/hadoop/hdfs-site.xml
```

添加下面配置到
与
标签之间。

```
1 | <property>
2 |     <name>dfs.replication</name>
3 |     <value>3</value>
4 | </property>
```

保存并关闭编辑器

- **8.修改hadoop yarn-site.xml文件配置**

```
1 | leafpad /opt/hadoop/etc/hadoop/yarn-site.xml
```

添加下面配置到
与
标签之间。

```
1 | <property>
2 |     <name>yarn.nodemanager.aux-services</name>
3 |     <value>mapreduce_shuffle</value>
4 | </property>
5 | <property>
6 |     <name>yarn.nodemanager.env-whitelist</name>
7 |     <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,C
8 |     LASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
9 | </property>
```

保存并关闭编辑器

- **9.mapred-site.xml文件配置**

```
1 | leafpad /opt/hadoop/etc/hadoop/mapred-site.xml
```

添加下面配置到
与
标签之间。

```
1 | <property>
2 |     <name>mapreduce.framework.name</name>
3 |     <value>yarn</value>
4 | </property>
```

保存并关闭编辑器

- **10.修改hadoop workers文件配置**

```
1 | leafpad /opt/hadoop/etc/hadoop/workers
```

覆盖写入主节点映射名和从节点映射名：

```
1 | master
2 | slave1
3 | slave2
```

保存并关闭编辑器

- **11.修改hosts文件**

查看master ip地址

```
1 | ifconfig eth0|sed -n '2p'|awk -F " " '{print $2}'|awk -F ":" '{print $2}'
```

记录下显示的ip, 例: 172.18.0.4

打开slave1 节点, 做如上操作, 记录下显示的ip, 例: 172.18.0.3

打开slave2 节点, 做如上操作, 记录下显示的ip, 例: 172.18.0.2

编辑/etc/hosts文件:

```
1 | sudo leafpad /etc/hosts
```

添加master IP地址对应本机映射名和其它节点IP地址对应映射名(如下只是样式, 请写入实验时您的正确IP):

```
1 | 172.18.0.4 master
2 | 172.18.0.3 slave1
3 | 172.18.0.2 slave2
```

- **12.创建公钥**

在dolphin用户下创建公钥:

```
1 | ssh-keygen -t rsa
```

出现如下内容:

Enter file in which to save the key (/home/dolphin/.ssh/id_rsa):

回车即可, 出现如下内容:

Enter passphrase (empty for no passphrase):

直接回车, 出现内容:

Enter same passphrase again:

直接回车, 创建完成。

- **13.拷贝公钥**

提示：命令执行过程中需要输入“yes”和密码“dolphin”。三台节点请依次执行完成。

```
1 | ssh-copy-id master
```

```
1 | ssh-copy-id slave1
```

```
1 | ssh-copy-id slave2
```

测试连接是否正常：

```
1 | ssh master
```

- **14.拷贝文件到所有从节点**

```
1 | scp -r /opt/java/ /opt/hadoop/ slave1:/tmp/
```

```
1 | scp -r /opt/java/ /opt/hadoop/ slave2:/tmp/
```

至此，主节点配置完成。

现在，请去slave1和slave2依次完成节点配置。

以下内容在所有从节点配置完成之后回来继续进行！

- **15.格式化分布式文件系统**

```
1 | hdfs namenode -format
```

- **16.启动Hadoop**

```
1 | /opt/hadoop/sbin/start-all.sh
```

- **17.查看Hadoop进程**

在hadoop主节点执行：

```
jps
```

输出结果必须包含6个进程，结果如下：

```
1 | 2529 DataNode
2 | 2756 SecondaryNameNode
3 | 3269 NodeManager
4 | 3449 Jps
5 | 2986 ResourceManager
6 | 2412 NameNode
```

在hadoop从节点执行同样的操作：

```
jps
```

输出结果必须包含3个进程，具体如下：

```
1 2529 DataNode
2 3449 Jps
3 2412 NameNode
```

- 18.在命令行中输入以下代码，打开Hadoop WebUI管理界面：

```
1 firefox http://master:8088
```

- 19.测试HDFS集群以及MapReduce任务程序

利用Hadoop自带的WordCount示例程序进行检查集群；在主节点进行如下操作，创建HDFS目录：

```
1 hadoop fs -mkdir /dolphin/
```

```
1 hadoop fs -mkdir /dolphin/input
```

创建测试文件

```
1 leafpad /home/dolphin/test
```

添加下面文字

dolphin

保存并关闭编辑器

将测试文件上传到Hadoop HDFS集群目录：

```
1 hadoop fs -put /home/dolphin/test /dolphin/input
```

执行wordcount程序：

```
1 hadoop jar /opt/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.0.0.jar wordcount /dolphin/input/ /dolphin/out/
```

查看执行结果：

```
1 hadoop fs -ls /dolphin/out/
```

如果列表中结果包含“_SUCCESS”文件，代码集群运行成功。

```
[root@Aliyun sbin]# hadoop fs -ls /shenhao/out/
Found 2 items
-rw-r--r--  1 root supergroup      0 2021-03-24 20:24 /shenhao/out/_SUCCESS
-rw-r--r--  1 root supergroup    17 2021-03-24 20:24 /shenhao/out/part-r-00000
```

查看具体的执行结果，可以用如下命令：

```
1 hadoop fs -text /dolphin/out/part-r-00000
```

```
[root@Aliyun sbin]# hadoop fs -text /shenhao/out/part-r-00000
Hello 1
world! 1
[root@Aliyun sbin]#
```

到此，集群安装完成。

