**图书馆集群使用说明**

1.20台机器全启动其Ubuntu系统，登陆用户名和密码为：root 941888 。

2.在SparkMaster主机上，启动命令行窗口。

3.输入命令：cd /usr/local/spark-1.6.0-bin-hadoop1/sbin 进入该文件夹

4.输入命令：start-all.sh 启动集群第一步

5.输入命令：./start-all.sh 启动集群第二步，也是最后一步

6.一般情况下，在本地Windows机上启动IDEA集成环境来编写程序，然后打成jar包，上传到个人目录下，再执行spark-submit运行jar包。如果用到其他 .jar 包，则需要把这些包添加进来再运行程序。

7.注意一下，关闭集群的方法是，在/usr/local/spark/sbin目录下，先执行 ./stop-all.sh 命令，再执行 stop-all.sh 命令。

**注意事项：**

1. 在windows装IDEA开发环境时，按照指南，javaSDK-scalaSDK-IDEA（插件、配置环境）-指南里的例子-spark自带的example包里的例子，这个顺序来。这样以后开发基本都在windows下面，在master上建自己用户目录，主要就为接受上传的jar包，并用spark-submit运行jar包。
2. 每位集群使用者，在SparkMaster上的/usr/data1目录下建立个人目录存放上传的jar包及样本数据。

|  |  |  |  |
| --- | --- | --- | --- |
| **主机名** | **IP** | **用户名** | **密码** |
| SparkMaster | 202.119.194.93 | root | 941888 |
| SparkWorker1 | 202.119.194.94 | root | 941888 |
| SparkWorker2 | 202.119.194.95 | root | 941888 |
| SparkWorker3 | 202.119.194.96 | root | 941888 |
| SparkWorker4 | 202.119.194.97 | root | 941888 |
| SparkWorker5 | 202.119.194.98 | root | 941888 |
| SparkWorker6 | 202.119.194.99 | root | 941888 |
| SparkWorker7 | 202.119.194.101 | root | 941888 |
| SparkWorker8 | 202.119.194.102 | root | 941888 |
| SparkWorker9 | 202.119.194.103 | root | 941888 |
| SparkWorker10 | 202.119.194.104 | root | 941888 |
| SparkWorker11 | 202.119.194.105 | root | 941888 |
| SparkWorker12 | 202.119.194.106 | root | 941888 |
| SparkWorker13 | 202.119.194.107 | root | 941888 |
| SparkWorker14 | 202.119.194.108 | root | 941888 |
| SparkWorker15 | 202.119.194.109 | root | 941888 |
| SparkWorker16 | 202.119.194.110 | root | 941888 |
| SparkWorker17 | 202.119.194.113 | root | 941888 |
| SparkWorker18 | 202.119.194.114 | root | 941888 |
| SparkWorker19 | 202.119.194.115 | root | 941888 |

**Spark集群搭建流程**

所用软个版本:

（1）Ubuntu14.10

（2）jdk1.7.0\_101

（3）hadoop-1.2.1

（4）scala-2.10.4

（5）spark-1.6.0

**目录**

一、安装Ubuntu14.10系统

二、安装SSH

三、安装JDK

四、安装Hadoop

五、安装Scala

六、安装Spark

七、安装Idea（在本地Windows)

**一、安装Ubuntu14.10系统**

1. 制作USB启动盘

2. 设置BIOS为U盘启动

1 Removable Device

2 Hard Device

Hard Device

1 USB

2 SATA

3. 安装成功后先连网，更新语言包

4. 设置root 权限

sudo -s //获取root权限，其中要输入开机密码

apt-get install vim //安装vim

vim /usr/share/lightdm/lightdm.conf.d/50-ubuntu.conf //打开文件，添加一句

greeter-show-manual-login=true

sudo passwd root //设置root密码

输入两次新密码，按回车完成密码设置

5. 修改hostname和hosts

vim /etc/hostname

将每台节点分别改成 ：

SparkMaster SparkWorker1 SparkWorker2 …… SparkWorker19

vim /etc/hosts

此处三台机器一致，第一行不变，第二行变成下面三行： （IP地址由实际情况而定）

202.119.194.93 SparkMaster

202.119.194.94 SparkWorker1

202.119.194.95 SparkWorker2

202.119.194.96 SparkWorker3

202.119.194.97 SparkWorker4

202.119.194.98 SparkWorker5

202.119.194.99 SparkWorker6

202.119.194.101 SparkWorker7

202.119.194.102 SparkWorker8

202.119.194.103 SparkWorker9

202.119.194.104 SparkWorker10

202.119.194.105 SparkWorker11

202.119.194.106 SparkWorker12

202.119.194.107 SparkWorker13

202.119.194.108 SparkWorker14

202.119.194.109 SparkWorker15

202.119.194.110 SparkWorker16

202.119.194.113 SparkWorker17

202.119.194.114 SparkWorker18

202.119.194.115 SparkWorker19

6. 完成后，重启，使root权限和机器名称生效。

**二、安装SSH**

1. apt-get install ssh //安装

ssh start //启动

验证是否正常启动：

ps -e |grep ssh

出现： ……………… sshd 说明已正常启动

2. 生成私钥和公钥

ssh-keygen -t rsa 加4个回车

或

ssh-keygen -t rsa -p “”

生成的私钥id\_rsa、公钥id\_rsa.pub保存在 /root/.ssh

3. 20台机器都生成公、私钥后，将19台从机的公钥id\_rsa.pub复制到主机的 /root/.ssh ，

为避免名称冲突，分别以id\_rsa.pub.SparkWorker1、id\_rsa.pub.SparkWorker2保存

cat id\_rsa.pub >> /root/.ssh/id\_rsa.SparkWorker1

然后，将20台机器的公钥集中在一起，保存在authorized\_keys

cat id\_rsa.pub >> authorized\_keys

cat id\_rsa.pub.SparkWorker1 >> authorized\_keys

cat id\_rsa.pub.SparkWorker2 >> authorized\_keys

……

cat id\_rsa.pub.SparkWorker19 >> authorized\_keys

最后，将存有20台机器公钥的authorized\_keys复制到19从机上

cat authorized\_keys >> /root/.ssh/authorized\_keys

这一步的意义就在于让所有的机器拥有包括自身在内的所有机器的公钥，以便于相互间无密码通信。

4. 完成后验证

ssh SparkWorker1 验证无密码通信

**三、安装JDK**

1. 获取java-7-openjdk-amd64.tar.gz

2. 解压

tar -xvf java-7-openjdk-amd64.tar.gz

1. 在 /usr/lib 中新建 jvm 文件夹，将解压的jdk剪切到 /usr/lib/jvm 中。

4. 修改环境变量

vim ~/.bashrc

添加：

export JAVA\_HOME=/usr/lib/jvm/java-7-openjdk-amd64

export JRE\_HOME=${JAVA\_HOME}/jre

export CLASS\_PATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib

export PATH= ${JAVA\_HOME}/bin:$PATH

然后

source ~/.bashrc //使修改生效

5. 分别验证java 版本

java -version

6. 将~/.bashrc 复制到另外两台机器上

scp ~/.bashrc [root@SparkWorker1](mailto:root@SparkWorker1):~/.bashrc

scp ~/.bashrc [root@SparkWorker2](mailto:root@SparkWorker2):~/.bashrc

……

scp ~/.bashrc [root@SparkWorker1](mailto:root@SparkWorker2)9:~/.bashrc

并使其生效：source ~/.bashrc

8. 安装完java后，再安装一个小东西

apt-get install rsync

**四、安装Hadoop**

1. 获取 hadoop-1.2.1.tar.gz

2. 解压

tar -xzf hadoop-1.2.1.tar.gz

3. 解压为 hadoop-1.2.1 ,移动到 /usr/local/

4. 配置“HADOOP\_HOME”,并把“HADOOP\_HOME”下的bin加入PATH中。

vim ~/.bashrc

添加：

export HADOOP\_HOME=/usr/local/hadoop

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

然后

source ~/.bashrc //使修改生效

5. 另外19台，用 scp 复制一下即可，并使之生效。

6. 到 /usr/local/hadoop-1.2.1/ 下新建文件夹

mkdir hdfs

mkdir tmp

mkdir hdfs/data

mkdir hdfs/name

7. 到 /usr/local/hadoop-1.2.1/conf, 修改 hadoop-env.sh

vim hadoop-env.sh

将 export JAVA\_HOME=${JAVA\_HOME} 改为

export JAVA\_HOME=/usr/lib/jvm/java-7-openjdk-amd64

8. 修改 slaves

vim slaves

将 localhost 改为

SparkWorker1

SparkWorker2

……

SparkWorker19

9. 修改 core-site.xml

<configuration>

<property>

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

<value>hdfs://202.119.194.93:9000</value>

</property>

<property>

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

<value>/usr/local/hadoop-1.2.1/tmp</value>

</property>

</configuration>

10. 修改 hdfs-site.xml

<configuration>

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

<property>

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

<value>/usr/local/hadoop-1.2.1/hdfs/name</value>

</property>

<property>

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

<value>/usr/local/hadoop-1.2.1/hdfs/data</value>

</property>

</configuration>

11. 修改 mapred-site.xml

复制一份mapred-site.xml.template命名为mapred-site.xml, 打开

<configuration>

<property>

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

<value>202.119.194.93:9001</value>

</property>

</configuration>

12. 另外19台，复制一下。

13. 格式化hdfs文件系统

cd /usr/local/hadoop-1.2,1/bin

hadoop namenode -format

进入 hadoop-1.2.1/bin

./start-all.sh

用jps 查看进程

访问 ：[http://202.119.194.93:50070](http://SparkMaster:50070/)/dfshealth.jsp

访问： <http://202.119.194.93:50030/jobtracker.jsp>

14. 验证hadoop 分布式集群

hadoop fs -mkdir -p /data/wordcount

hadoop fs -mkdir -p /output/wordcount

/usr/local/hadoop-1.2.1# hadoop fs -put ../conf/\*.xml /data/wordcount/ //上传文件

/usr/local/hadoop-1.2.1# hadoop jar hadoop-examples-1.2.1.jar wordcount /data/wordcount /output/wordcount

完成 ！！！

**五、安装Scala**

1. 获取 scala-2.10.4.tgz

2. 解压

tar -zxf scala-2.10.4.tgz

3. 解压名为 scala-2.10.4, 剪切到 /usr/lib/

4. 配置“SCALA\_HOME”,并把“SCALA\_HOME”下的bin加入PATH中。

export SCALA\_HOME=/usr/lib/scala-2.10.4

export PATH=${SCALA\_HOME}/bin:${JAVA\_HOME}/bin:${HADOOP\_HOME}/bin:$PATH

source ~/.bashrc 使生效

5. 另外19台，用 scp 复制一下即可，并使之生效。

6. 分别验证其scala 版本

scala -version

**六、安装Spark**

1. 获取 spark-1.2.1-bin-hadoop1.tgz

2. 解压

tar -zxf spark-1.2.1-bin-hadoop1.tgz

3. 解压名为 spark-1.2.1-bin-hadoop1, 剪切到 /usr/local/

4. 配置“SPARK\_HOME”,并把“SPARK\_HOME”下的bin加入PATH中。

export SPARK\_HOME=/usr/local/-1.2.1-bin-hadoop1

export PATH=${SPARK\_HOME}/bin:${SCALA\_HOME}/bin:${JAVA\_HOME}/bin:${HADOOP\_HOME}/bin:$PATH

source ~/.bashrc 使生效

5. 另外19台，用 scp 复制一下即可，并使之生效。

6. 配置spark 的conf目录

cd /usr/local/spark-1.2.1-bin-hadoop1/conf

修改slaves

vim slaves

将 localhost 改为:

SparkWorker1

SparkWorker2

……

SparkWorker19

7. 把spark-env.sh.template复制到spark-env.sh

cp spark-env.sh.template spark-env.sh

vim spark-env.sh

添加如下信息：

export JAVA\_HOME=/usr/lib/jvm/java-7-openjdk-amd64

export SCALA\_HOME=/usr/lib/jvm/scala-2.10.4

export SPARK\_MASTER\_IP=202.119.194.93

export SPARK\_WORKER\_MEMORY=4g

export HADOOP\_CONF\_DIR=/usr/local/hadoop-1.2.1

8. 另外19台，用 scp 复制一下即可，并使之生效。

9. 至此，Spark安装完毕，可以启动了。

cd /usr/local/hadoop-1.2.1/sbin

./start-dfs.sh

./mr-jobhistory-deamon.sh start historyserver

cd /usr/local/spark-1.2.1-bin-hadoop1/sbin

./start-all.sh

用jps 查看进程信息

访问：[http://202.119.194.93:8080](http://SparkMaster:8080/)

cd ..

cd bin

spark-shell //启动 spark-shell

访问：[http://202.119.194.93:4040](http://SparkMaster:4040/)

10. 测试Spark集群

cd /usr/local/spark-1.2.1-bin-hadoop1

hadoop fs -put README.md /data/

启动 spark-shell

val rdd = sc.textFile(“hdfs://202.119.194.93:9000/data/README.md”)

val count = rdd.flatMap(\_.split(“ “)).map((\_,1)).reduceByKey(\_+\_).collect

完成！！！

**七、安装Idea（在本地Windows)**

1. 获取 ideaIC-13.1.5.tar.gz

2. 解压

tar -xzf ideaIC-13.1.5.tar.gz

3. 重命名为 idea, 剪切到 /usr/local/

4. 选 Configure

选 Plugins

点击左下角的 Install JetBrains plugin

在左上方的输入框中输入 scala 来查找 scala 插件

点击右侧的 Install plugin

安完后重启 idea.sh

5. 这次选 Create New Project->Scala->SBT->起个名->finish

6. 添加jdk

File->Project Structure->Project->New 加入jdk路径 ：/usr/lib/jvm/java-7-openjdk-amd64

7. 重进工程，等待SBT加载

8. 设置黑色背景

File->Settings->Appearance->Theme 选择 Darcula

重新启动

9. idea里编程要加两句：

conf.setMaster("spark://202.119.194.93:7077")

sc.addJar("/usr/local/data1/个人目录名/helloworld.jar")

10. 读取hdfs上的文件：

val rdd1= sc.textFile("hdfs://202.119.194.93:9000/data/README.md")

往hdfs上添加文件，用命令：

hadoop fs -copyFromLocal README.md /data

hadoop fs -put README.md /data／

11.提交jar包

在IDEA上打包程序的过程   
 File - project structure - Artifacts - Jar - FromModuleWithDependencies - Mainclass

Buid - Build Artifacts - Build

注意：可以把Spark和Scala的jar去掉, jar包会在系统集群上寻找，减少包大小

/usr/local/spark-1.6.0-bin-hadoop1/bin/spark-submit   
 –class Spark.Word（package和object）   
 –master spark://202.119.194.93:7077   
 /usr/local/data1/个人目录/WordCount.jar