# Hadoop2.2.0+mahout0.9+Eclipse相关配置

## 一、准备工作

在网盘上下载后面需要用到的所有文件：<http://pan.baidu.com/s/1sjPR809>

将文件复制到/home目录下

sudo cp ~/Downloads/mahout-distribution-0.9.tar.gz /home

sudo cp ~/Downloads/apache-maven-3.2.2-bin.tar.gz /home

sudo cp ~/Downloads/hadoop-2.2.0.tar.gz /home

sudo cp ~/Downloads/jdk1.7.0\_25.tar.gz /home

**解压**

sudo tar –zxvf mahout-distribution-0.9.tar.gz

sudo tar –zxvf apache-maven-3.2.2-bin.tar.gz

sudo tar –zxvf hadoop-2.2.0.tar.gz

sudo tar –zxvf jdk1.7.0\_25.tar.gz

更改相关文件夹权限，后面可能要用到

sudo chmod -R 775 文件夹名称

### 1.环境变量配置

修改环境变量：sudo gedit /etc/profile

#set java environment

export JAVA\_HOME=/home/jdk1.7.0\_25

export JRE\_HOME=/home/jdk1.7.0\_25/jre

export HADOOP\_HOME=/home/hadoop-2.2.0

export MAHOUT\_HOME=/home/mahout-distribution-0.9

export MAVEN\_HOME=/home/apache-maven-3.2.2

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

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

保存并退出，执行以下命令使配置生效

sudo chmod +x /etc/profile

source /etc/profile

配置完毕，在命令行中使用命令“java - version”可以判断是否成功。

出现以下信息说明成功：

1. java version "1.7.0\_25"
2. Java(TM) SE Runtime Environment (build 1.7.0-b147)
3. Java HotSpot(TM) Client VM (build 21.0-b17, mixed mode)

如果JDK版本没更新，那么设置系统默认JDK

sudo update-alternatives --install /usr/bin/java java /home/jdk1.7.0\_25/bin/java 300

sudo update-alternatives --install /usr/bin/javac javac /home/jdk1.7.0\_25/bin/javac 300

sudo update-alternatives --config java

系统会列出各种JDK版本，如下所示：

1· root@linux:~$ sudo update-alternatives --config java

2· 有 3 个候选项可用于替换 java (提供 /usr/bin/java)。

3· 选择路径优先级状态

4· ------------------------------------------------------------

5·

6· \* 0    /usr/lib/jvm/java-6-openjdk/jre/bin/java   1061  自动模式

7·  1    /usr/lib/jvm/java-6-openjdk/jre/bin/java   1061  手动模式

8·  2    /usr/lib/jvm/java-6-sun/jre/bin/java      63   手动模式

9·  3    /usr/lib/jvm/java-7-sun/bin/java        300   手动模式

10· 要维持当前值[\*]请按回车键，或者键入选择的编号：3

11· update-alternatives: 使用 /usr/lib/jvm/java-7-sun/bin/java 来提供 /usr/bin/java (java)，于手动模式中。

## 二、Hadoop2.2.0配置

### 1.创建用户

首先在两台虚拟机上创建相同的用户

sudo addgroup hadoop

sudo adduser --ingroup hadoop haduser

编辑/etc/sudoers编辑文件，在root ALL=(ALL)ALL行下添加haduser ALL=(ALL)ALL。如果不添加这行，haduser将不能执行sudo操作。

确保两台虚拟机上已经安装了jdk，并正确配置了环境变量。

### 2.配置ssh免密码登录

安装ssh 所有节点用haduser用户登录：su haduser

一般系统是默认安装了ssh命令的。如果没有，或者版本比较老，则可以重新安装：

sudo apt-get install ssh

安装完成后会在~目录（当前用户主目录，即这里的/home/haduser）下产生一个隐藏文件夹.ssh（ls -a 可以查看隐藏文件）。如果没有这个文件，自己新建即可（mkdir .ssh）。

进入.ssh文件夹

ssh-keygen -t rsa之后一路回车（产生秘钥）

以上步骤在master(10.18.2.53)和slave(10.18.2.31)上都要进行操作，然后把slave上的公钥发给master

scp id\_rsa.pub 10.18.2.53:~/.ssh/id\_pub.slave

接着在master上进行操作

cat id\_rsa.pub >> authorized\_keys把id\_rsa.pub 追加到授权的 key 里面去

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

把授权的key发给slave

scp authorized\_keys 10.18.2.31:~/.ssh

此时已经可以进行ssh的无密码登陆,查看是否可以从master主机无密码登录slave，输入命令：

ssh 10.18.2.31

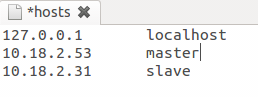
如需要可重启ssh服务/etc/init.d/ssh restart

用root身份在两台主机上分别设置：/etc/hosts 和/etc/hostname

hosts这个文件用于定义主机名和IP地址之间的映射关系。

su root

gedit /etc/hosts



/etc/hostname这个文件用于定义Ubuntu的主机名：如：master（或者slave等）

gedit /etc/hostname

如果主机名修改生效，则进行下一步。（主机命未改变则重启虚拟机）

### 3.HDFS配置

配置/home/hduser/hadoop/etc/hadoop/hadoop-env.sh

替换export JAVA\_HOME=${JAVA\_HOME}为如下：

export JAVA\_HOME=/home/jdk1.7.0\_25 (以自己的jdk为准)

同样，配置yarn-env.sh，在里面加入：

export JAVA\_HOME=/home/jdk1.7.0\_25

配置etc/hadoop/core-site.xml文件内容：

1. **<?xml** version="1.0" encoding="UTF-8"**?>**
2. **<?xml-stylesheet** type="text/xsl" href="configuration.xsl"**?>**
4. **<configuration>**
5. **<property>**
6. **<name>**fs.default.name**</name>**
7. **<value>**hdfs://master:9000/**</value>**
8. **<description>**The name of the default file system. A URI whose scheme and authority determine the FileSystem implementation. The uri's scheme determines the config property (fs.SCHEME.impl) naming the FileSystem implementation class. The uri's authority is used to determine the host, port, etc. for a filesystem.**</description>**
9. **</property>**
10. **<property>**
11. **<name>**dfs.replication**</name>**
12. **<value>**3**</value>**
13. **</property>**
14. **<property>**
15. **<name>**hadoop.tmp.dir**</name>**
16. **<value>**/tmp/hadoop-${user.name}**</value>**
17. **<description></description>**
18. **</property>**
19. **</configuration>**

配置etc/hadoop/hdfs-site.xml文件内容：

1. **<?xml** version="1.0" encoding="UTF-8"**?>**
2. **<?xml-stylesheet** type="text/xsl" href="configuration.xsl"**?>**
4. **<configuration>**
5. **<property>**
6. **<name>**dfs.namenode.name.dir**</name>**
7. **<value>**/home/haduser/hadoop/storage/hadoop2/hdfs/name**</value>**
8. **<description>**Path on the local filesystem where the NameNode stores the namespace and transactions logs persistently.**</description>**
9. **</property>**
10. **<property>**
11. **<name>**dfs.datanode.data.dir**</name>**
12. **<value>**/home/haduser/hadoop/storage/hadoop2/hdfs/data1,/home/haduser/hadoop/storage/hadoop2/hdfs/data2,/home/haduser/hadoop/storage/hadoop2/hdfs/data3**</value>**
13. **<description>**Comma separated list of paths on the local filesystem of a DataNode where it should store its blocks.**</description>**
14. **</property>**
15. **<property>**
16. **<name>**hadoop.tmp.dir**</name>**
17. **<value>**/home/haduser/hadoop/storage/hadoop2/hdfs/tmp/hadoop-${user.name}**</value>**
18. **<description>**A base for other temporary directories.**</description>**
19. **</property>**
20. **</configuration>**

### 4.YARN安装配置

配置etc/hadoop/yarn-site.xml文件内容：

1. **<?xml** version="1.0"**?>**
3. **<configuration>**
4. **<property>**
5. **<name>**yarn.resourcemanager.resource-tracker.address**</name>**
6. **<value>**master:8031**</value>**
7. **<description>**host is the hostname of the resource manager and
8. port is the port on which the NodeManagers contact the Resource Manager.
9. **</description>**
10. **</property>**
12. **<property>**
13. **<name>**yarn.resourcemanager.scheduler.address**</name>**
14. **<value>**master:8030**</value>**
15. **<description>**host is the hostname of the resourcemanager and port is the port
16. on which the Applications in the cluster talk to the Resource Manager.
17. **</description>**
18. **</property>**
20. **<property>**
21. **<name>**yarn.resourcemanager.scheduler.class**</name>**
22. **<value>**org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.CapacityScheduler**</value>**
23. **<description>**In case you do not want to use the default scheduler**</description>**
24. **</property>**
26. **<property>**
27. **<name>**yarn.resourcemanager.address**</name>**
28. **<value>**master:8032**</value>**
29. **<description>**the host is the hostname of the ResourceManager and the port is the port on
30. which the clients can talk to the Resource Manager. **</description>**
31. **</property>**
33. **<property>**
34. **<name>**yarn.nodemanager.local-dirs**</name>**
35. **<value>**${hadoop.tmp.dir}/nodemanager/local**</value>**
36. **<description>**the local directories used by the nodemanager**</description>**
37. **</property>**
39. **<property>**
40. **<name>**yarn.nodemanager.address**</name>**
41. **<value>**0.0.0.0:8034**</value>**
42. **<description>**the nodemanagers bind to this port**</description>**
43. **</property>**
45. **<property>**
46. **<name>**yarn.nodemanager.resource.memory-mb**</name>**
47. **<value>**10240**</value>**
48. **<description>**the amount of memory on the NodeManager in GB**</description>**
49. **</property>**
51. **<property>**
52. **<name>**yarn.nodemanager.remote-app-log-dir**</name>**
53. **<value>**${hadoop.tmp.dir}/nodemanager/remote**</value>**
54. **<description>**directory on hdfs where the application logs are moved to **</description>**
55. **</property>**
57. **<property>**
58. **<name>**yarn.nodemanager.log-dirs**</name>**
59. **<value>**${hadoop.tmp.dir}/nodemanager/logs**</value>**
60. **<description>**the directories used by Nodemanagers as log directories**</description>**
61. **</property>**
63. **<property>**
64. **<name>**yarn.nodemanager.aux-services**</name>**
65. **<value>**mapreduce\_shuffle**</value>**
66. **<description>**shuffle service that needs to be set for Map Reduce to run **</description>**
67. **</property>**
68. **</configuration>**

注意此配置的最后一个property的value值，只能是A-Za-z0-9\_，不能以数字开头，否则可能造成nodemanager无法正常启动。

配置文件slaves （这个文件里面保存所有slave节点）

写入以下内容：

slave

配置完成后，复制到其他节点（slave）

sudo scp –r /home/hadoop-2.2.0/ slave:/home/

注意：只需要原样复制即可，不必改动上面的xml配置文件。切记...

### **5.启动HDFS集群和YARN集群**

首先，需要格式化HDFS，执行如下命令：

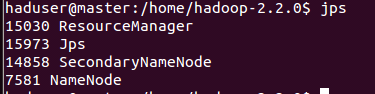
haduser@master:/home/hadoop-2.2.0$ bin/hdfs namenode –format

如果格式化正常，日志中不会出现异常信息，可以继续启动集群相关服务

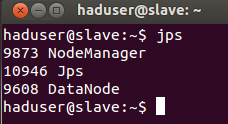
启动HDFS集群和YARN集群，执行如下命令：

haduser@master:/home/hadoop-2.2.0$ sbin/start-all.sh

可以在master结点上看到如下几个进程：



在slave结点上看到如下进程：



如果相应的进程不能启动，则关闭进程

sbin/stop-all.sh 删除格式化后生产的文件夹/hadoop/storage

再重复以上步骤。

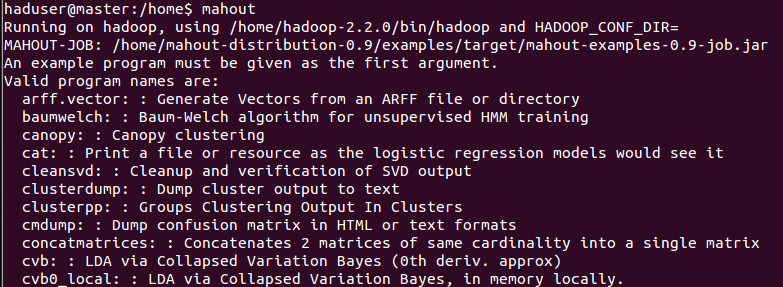
### 6.测试hadoop

现在进入mahout的安装文件夹进行打包

cd /home/ mahout-distribution-0.9

mvn package

这个过程可能需要10几分钟，结束后可以看到mahout相关的包已经下好。



现在在hadoop2.2.0上运行一个简单的例子

hadoop fs -mkdir testdata

hadoop fs -put ~/Downloads/synthetic\_control.data testdata

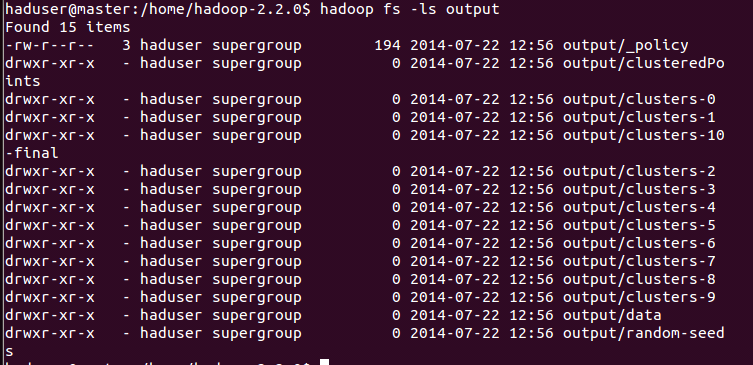


hadoop fs -ls testdata hadoop jar /home/mahout-distribution-0.9/examples/target/mahout-examples-0.9-job.jar org.apache.mahout.clustering.syntheticcontrol.kmeans.Job

红字部分是一条完整的命令（完成计算后会有数据直接显示）

查看结果

hadoop fs -ls output

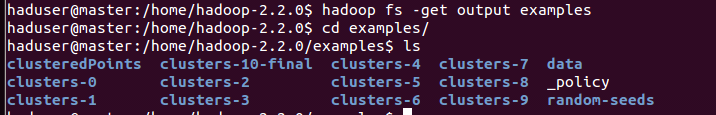


hadoop fs -get output

cd /home/examples/

ls

显示以下文件夹那么刚才的例子运行已经成功了



## 三、Eclipse配置

如果要用maven插件则需要在官网上下载最新的Eclipse版本，老版本的Eclipse的maven插件老是下不了= =。。。

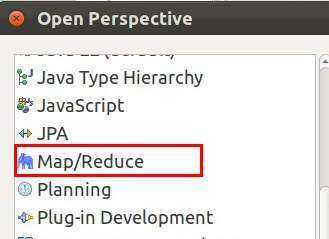
### 1.hadoop插件

下载地址：[http://www.eclipse.org/downloads/](http://www.eclipse.org/downloads/" \t "_blank)

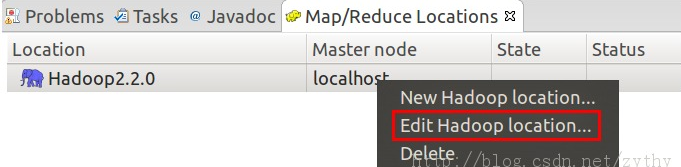
现在安装hadoop插件

sudo scp ~/Downloads/hadoop-eclipse-plugin-2.2.0.jar /home/eclipse/plugins

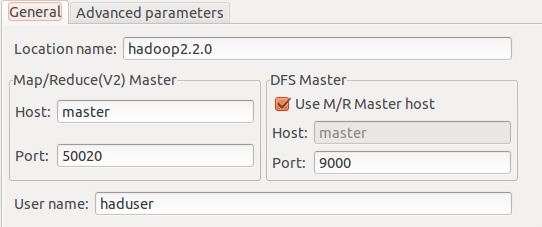
启动Eclipse通过Open Perspective菜单打开Map Reduce视图，如下：



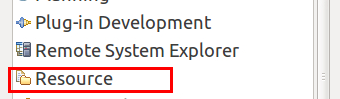
选中大象图标，右键点击Edit Hadoop Location编辑Hadoop配置信息：



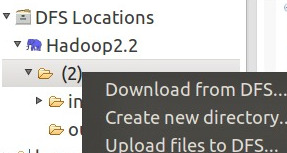
填写正确的Map/Reduce和HDFS信息。（具体根据您的配置而定）



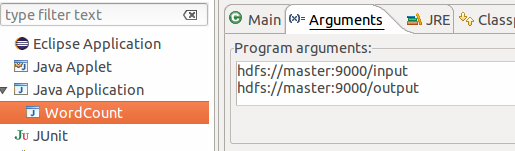
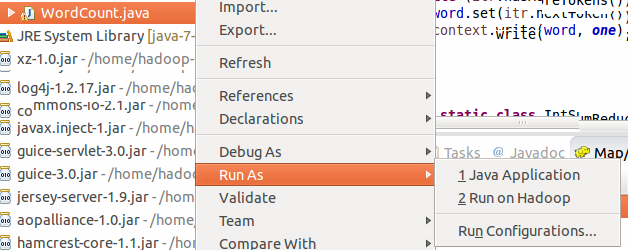
通过Eclipse访问DFS打开Resource视图，即可看到DFS：



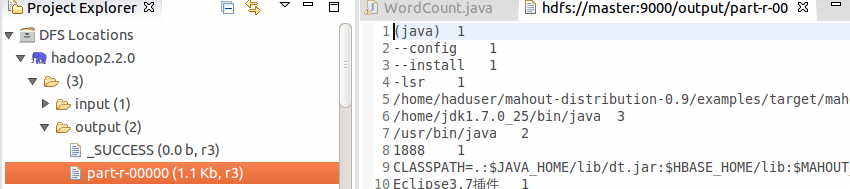
此时，你可对DFS进行一些操作，比如上传本地文件到HDFS等等：



现在运行一个简单的Wordcount例子（代码在网盘里有），在运行之前先配置程序参数

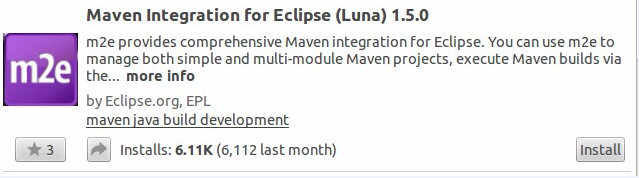


刷新后在output里可以看到结果，如下图

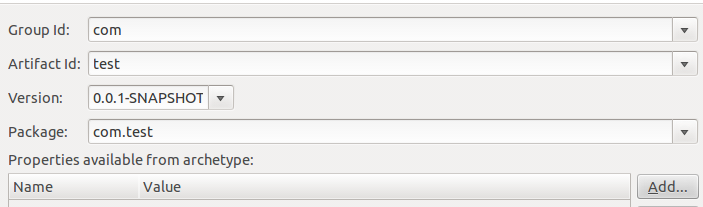
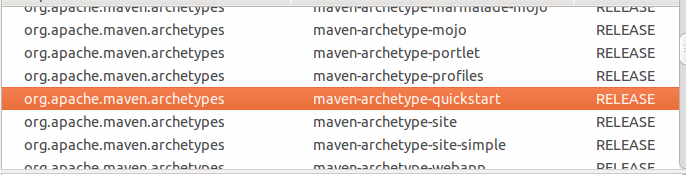
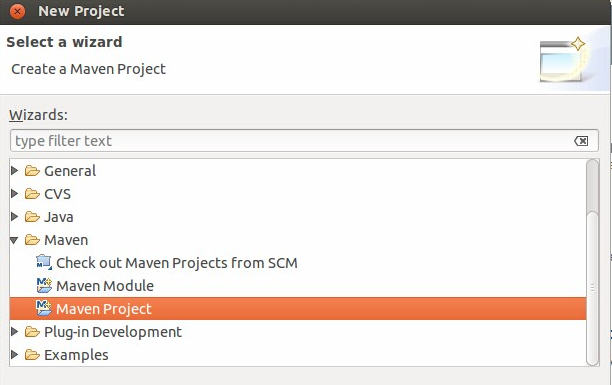


### 2.maven插件

在线安装maven插件

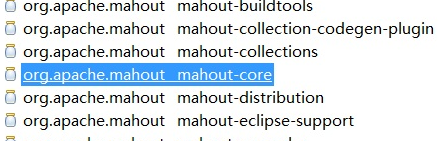


重启Eclipse后按照以下步奏可以新建一个maven项目



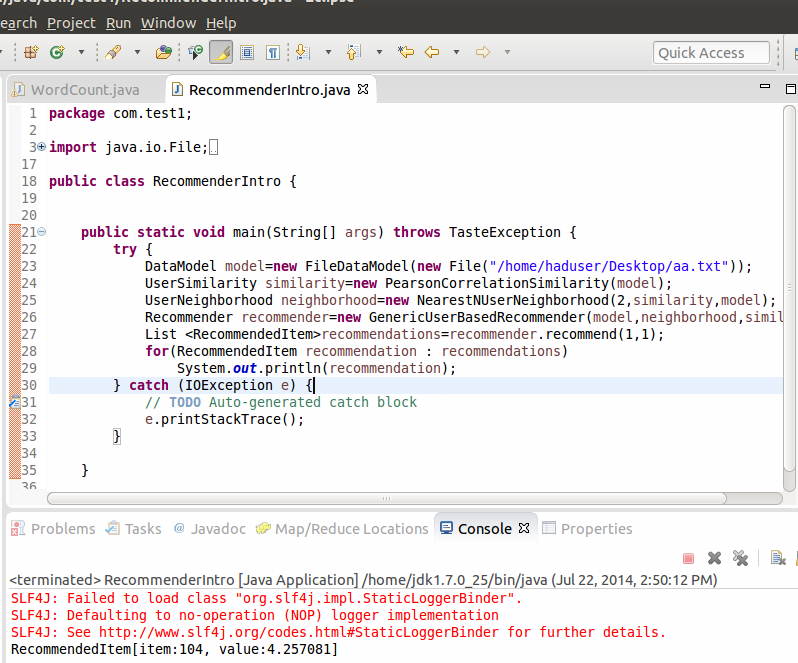
打开pom.xml 加入一些必备的jar。

点击 pom.xml 下的dependencies选项卡，在dependencies出点击add。 在弹出的对话框中输入”mahout“ 稍等片刻就会出来很多的jar包，选择合适的mahout包，一般来说，如果做的很简单的程序，选择mahout-core 就行，如果需要分布计算则需要加入 hadoop



运行一个简单的例子，

1. 1,101,5
2. 1,102,3
3. 1,103,2.5
4. 2,101,2
5. 2,102,2.5
6. 2,103,5
7. 2,104,2
8. 3,101,2.5
9. 3,104,4
10. 3,105,4.5
11. 3,107,5
12. 4,101,5
13. 4,103,3
14. 4,104,4.5
15. 4,106,4
16. 5,101,4
17. 5,102,3
18. 5,103,2
19. 5,104,4
20. 5,105,3.5
21. 5,106,4



如果缺少相应的jar包，可以在/home/mahout-distribution-0.9文件下导入相关包

## 四、总结

在以上的配置过程中遇到的大多数问题都是关于权限的问题，大家多注意一下，以上配置有什么不对的欢迎大家提出，我好及时整理修改。

参考资料网址：1.http://www.ituring.com.cn/ARTICLE/63927 (hadoop)

2.http://blog.csdn.net/cruise\_h/article/details/18716645 (hadoop)

3.http://www.micmiu.com/bigdata/hadoop/hadoop2x-cluster-setup (hadoop)

4.http://blog.csdn.net/microfhu/article/details/7667393 (jdk)