**系统配置**

系统配置主要分为以下几个阶段：

1. 综述
2. 安装ssh
3. 安装hadoop
4. 运行一个map/reduce任务
5. 安装Giraph
6. 运行一个Giraph任务

**综述**

这是一个手把手、一步步教你安装Giraph的教程。目标用户就是那些想用Giraph运行输入较小的任务的人。但不适合大型产品开发。

接下来，我们会教你在单机上一步步搭建一个单节点、伪分布式Hadoop集群。这个节点同时扮演master和slave。它将运行NameNode, SecondaryNameNode, JobTracker, DataNode,和TaskTracker Java这几个进程。我们也会在这个节点上使用Giraph。以下是这台机器上的基本配置：

* Ubuntu Server 14.04.2 (64-bit):
  + Admin account: sherry
  + Hostname: hdnode01
  + IP address: 192.168.56.10
  + Network mask: 255.255.255.0
* Apache Hadoop 0.20.203.0-RC1
* Apache Giraph 1.1.0

## 安装ssh

在Hadoop启动以后，Namenode是通过SSH（Secure Shell）来启动和停止各个datanode上的各种守护进程的，这就须要在节点之间执行指令的时候是不须要输入密码的形式，故我们须要配置SSH运用无密码公钥认证的形式。

**ubuntu**默认并没有安装**ssh**服务，如果通过ssh链接ubuntu，需要自己手动安装ssh-server。判断是否安装ssh服务，可以通过如下命令进行：

$ ssh localhost

若显示ssh: connect to host localhost port 22: Connection refused，表示没有还没有安装，可以通过apt安装，命令如下：

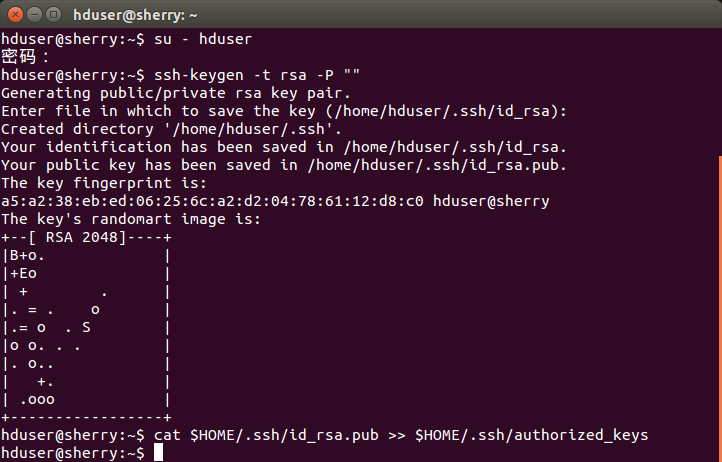
$ sudo apt-get install openssh-server

接下来**设置namenode的ssh为无需密码的、自动登录**

su – hduser

ssh-keygen -t rsa -P ""//一直回车即可

cat $HOME/.ssh/id\_rsa.pub >> $HOME/.ssh/authorized\_keys



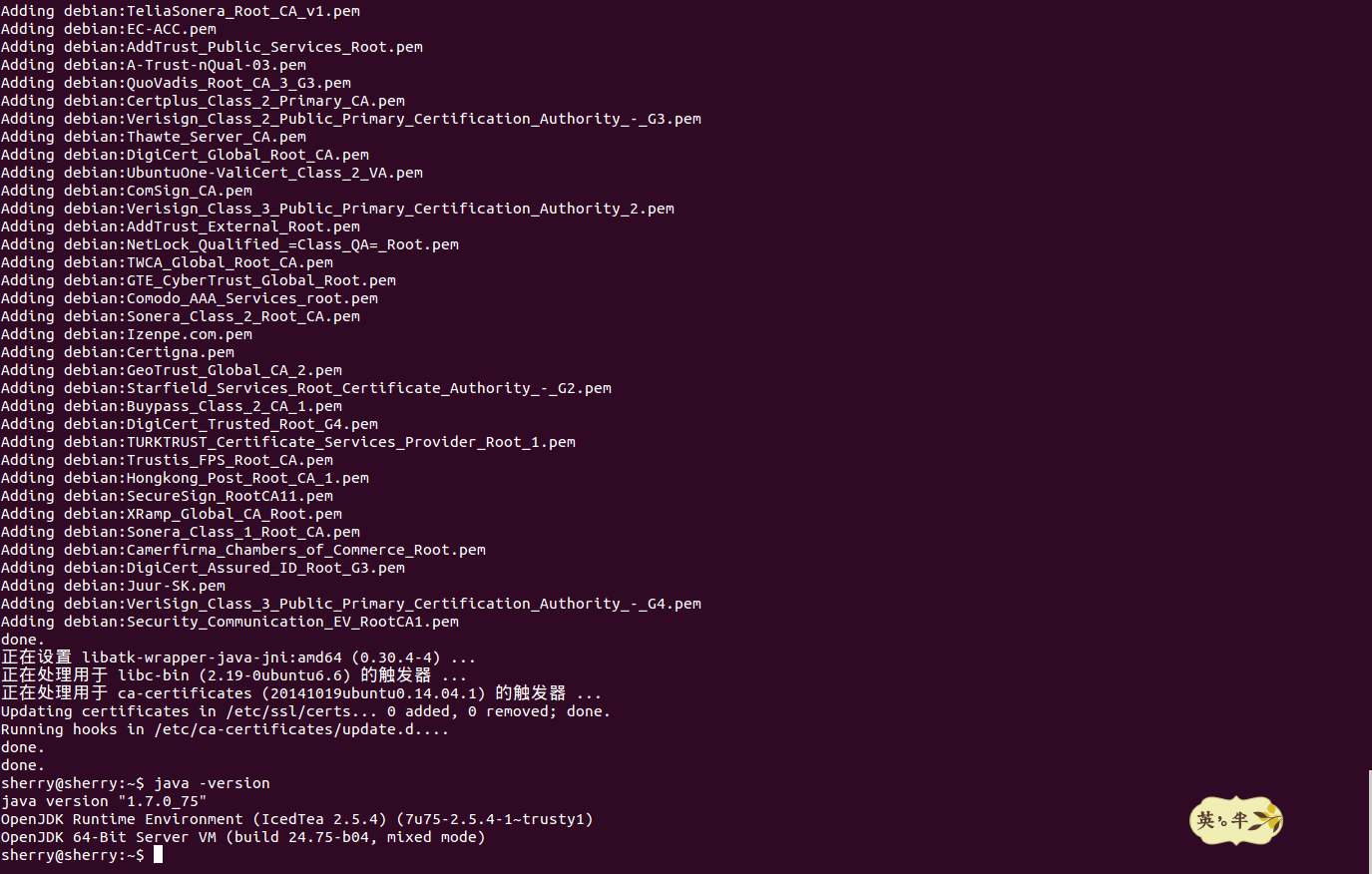
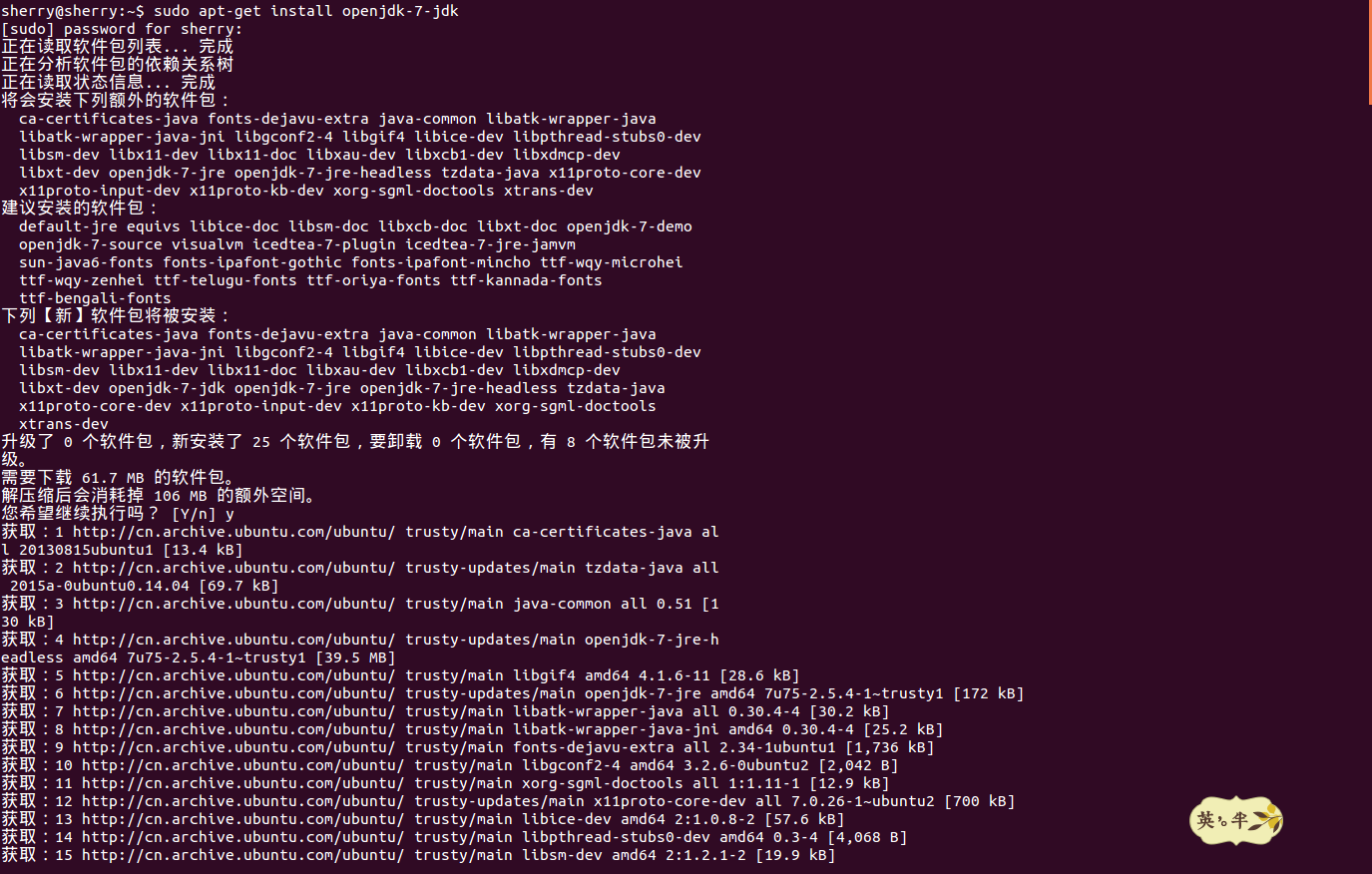
接下来，在hduser账户下使用ssh连接hdnode01，第一次在这个账户下连接ssh可能会被问到账户密码。连接之后，要把公开RSA秘钥存在$HOME/.ssh/known\_hosts中。

## 安装Hadoop

接下来我们要安装一个单节点、伪分布式的Hadoop集群。首先要安装java 1.6或以上的jdk。安装完成后要验证下是否安装成功。

sudo apt-get install openjdk-7-jdk

java –version

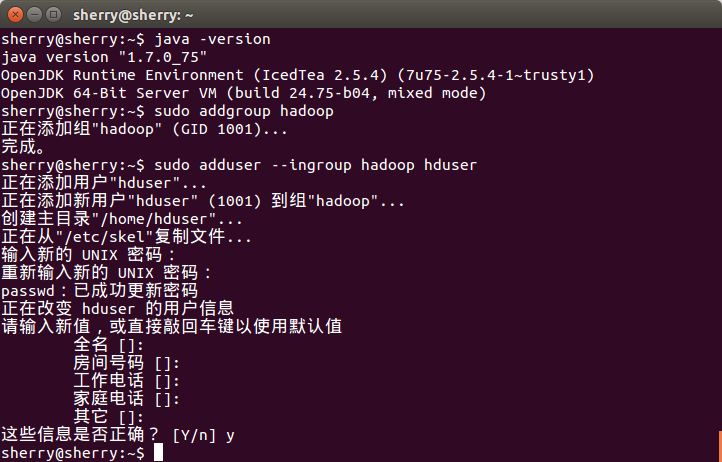


这时你会看到jave的版本信息。完整的JDK被安装在/usr/lib/jvm/java-7-openjdk-amd64，在这里你能找到Java的bin和lib目录。

接下俩要生成Hadoop组，一个新用户hduser，然后将这个用户加到新生成的组里。

sudo addgroup hadoop

sudo adduser --ingroup hadoop hduser



然后，从[Apache archives](http://archive.apache.org/dist/hadoop/core/)下载提取hadoop-0.20.203.0rc1(这时Giraph的默认版本)。

su – sherry

cd /usr/local

sudo wget http://archive.apache.org/dist/hadoop/core/hadoop-0.20.203.0/hadoop-0.20.203.0rc1.tar.gz

sudo tar xzf hadoop-0.20.203.0rc1.tar.gz

sudo mv hadoop-0.20.203.0 hadoop

sudo chown -R hduser:hadoop hadoop



安装完后切换到hduser账户，如下编辑$HOME/.bashrc:

export HADOOP\_HOME=/usr/local/hadoop

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

它会设定Hadoop/Java相关的环境变量。之后如下编辑$HADOOP\_HOME/conf/hadoop-env.sh：

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

export HADOOP\_OPTS=-Djava.net.preferIPv4Stack=true

第二行会强制Hadoop使用IPv4而不是IPv6，即使在这台机器上设置了IPv6。Hadoop会在计算时生成很多临时文件，所以需要为这些本地文件和HDFS设定存储位置。

su – sherry

sudo mkdir -p /app/hadoop/tmp

sudo chown hduser:hadoop /app/hadoop/tmp

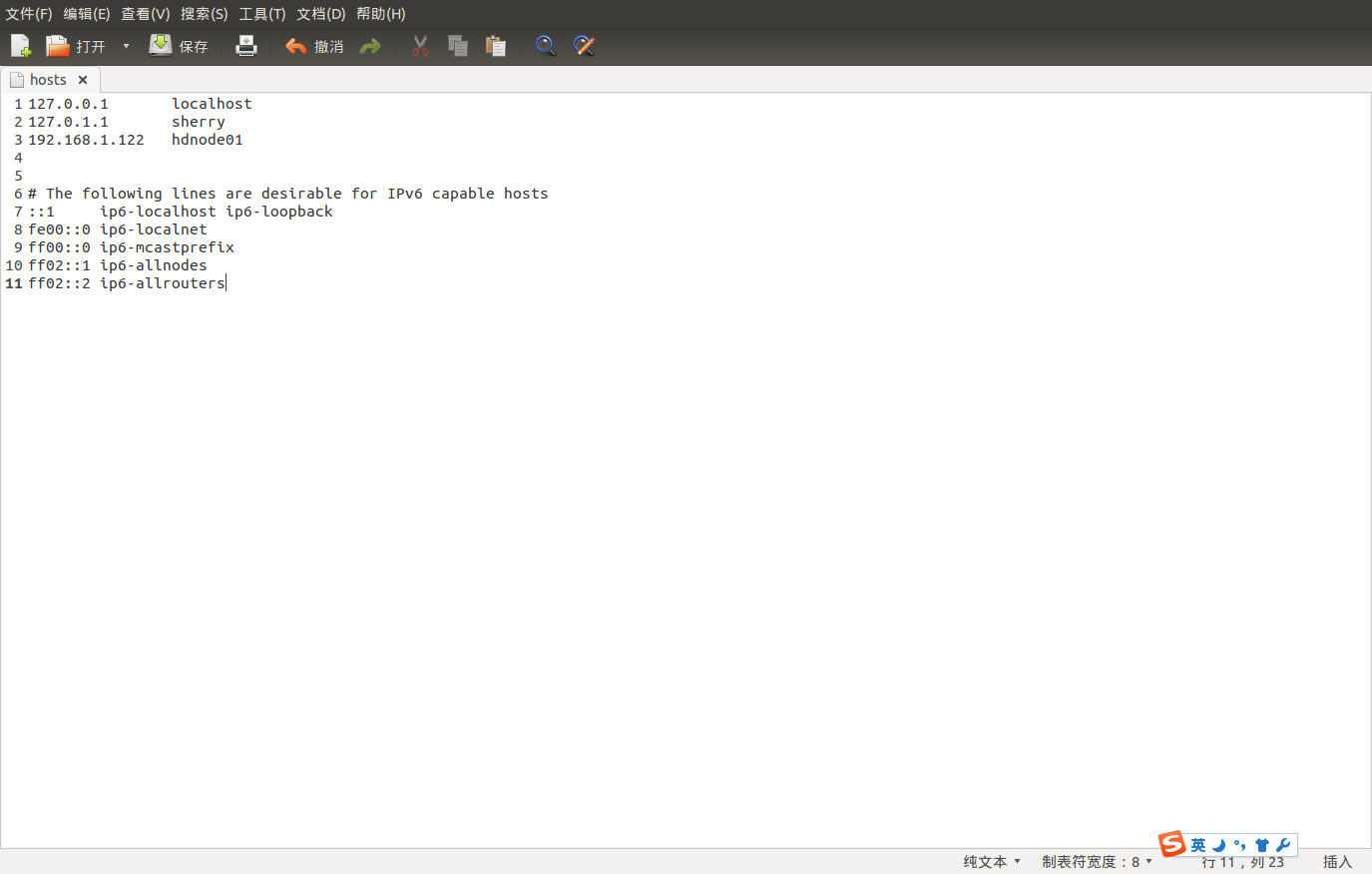
sudo chmod 750 /app/hadoop/tmp

确定/etc/hosts文件包含下面两行：

127.0.0.1 localhost

192.168.56.10 hdnode01

IP地址根据本机设置即可。



现在我们来编辑Hadoop的配置文件core-site.xml, mapred-site.xml, 和 hdfs-site.xml，它们都在$HADOOP\_HOME/conf文件夹下。

core-site.xml

<property>

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

<value>/app/hadoop/tmp</value>

</property>

<property>

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

<value>hdfs://hdnode01:54310</value>

</property>

mapred-site.xml

<property>

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

<value>hdnode01:54311</value>

</property>

<property>

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

<value>4</value>

</property>

<property>

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

<value>4</value>

</property>

Hadoop默认允许2个mapper同时运行，这里设置同时可以运行4个mapper。

相应的，对于这个单节点、伪分布式的设置来说，我们需要在mapred-site.xml中添加两个参数。

hdfs-site.xml

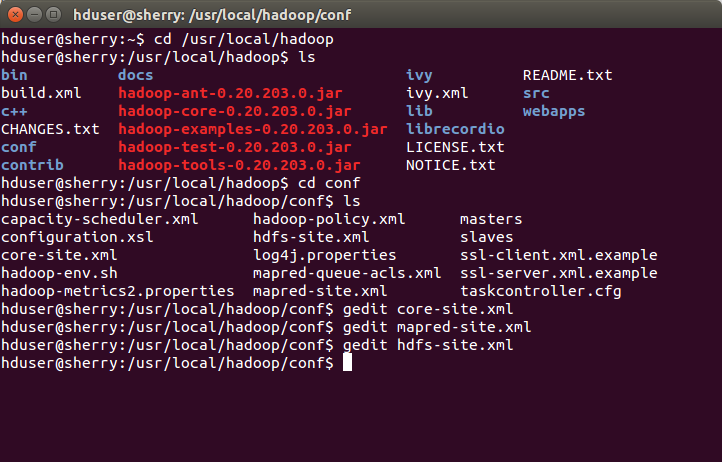
<property>

<name>dfs.replication</name>

<value>1</value>

</property>

注意到，你刚刚设置复制服务生成一份copy存储在HDFS中。这是因为你只有一个数据节点。默认值是3，如果不改的话，你会收到run-time exceptions。



编辑$HADOOP\_HOME/conf/masters：

hdnode01

相似地，编辑edit $HADOOP\_HOME/conf/slaves

hdnode01

以上设置成功后我们就得到了一个单节点、伪分布式Hadoop集群，它有一个master节点，一个slave节点，都布置在同一台物理机器上。注意，如果你想生成多个节点的分布式Hadoop集群，要在$HADOOP\_HOME/conf/slaves文件中添加其他节点名称，例如hdnode02, hdnode03, ...。详细信息参照Apache Hadoop的[官方网址](http://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/ClusterSetup.html)。

使用以下命令初始化HDFS：

$HADOOP\_HOME/bin/hadoop namenode -format

然后启动HDFS和map/reduce进程：

$HADOOP\_HOME/bin/start-dfs.sh

$HADOOP\_HOME/bin/start-mapred.sh

输入如下命令，确保所有Java的进程在hdnode01节点上。应该输出如下几行(进程号不用管)。

jps

9079 NameNode

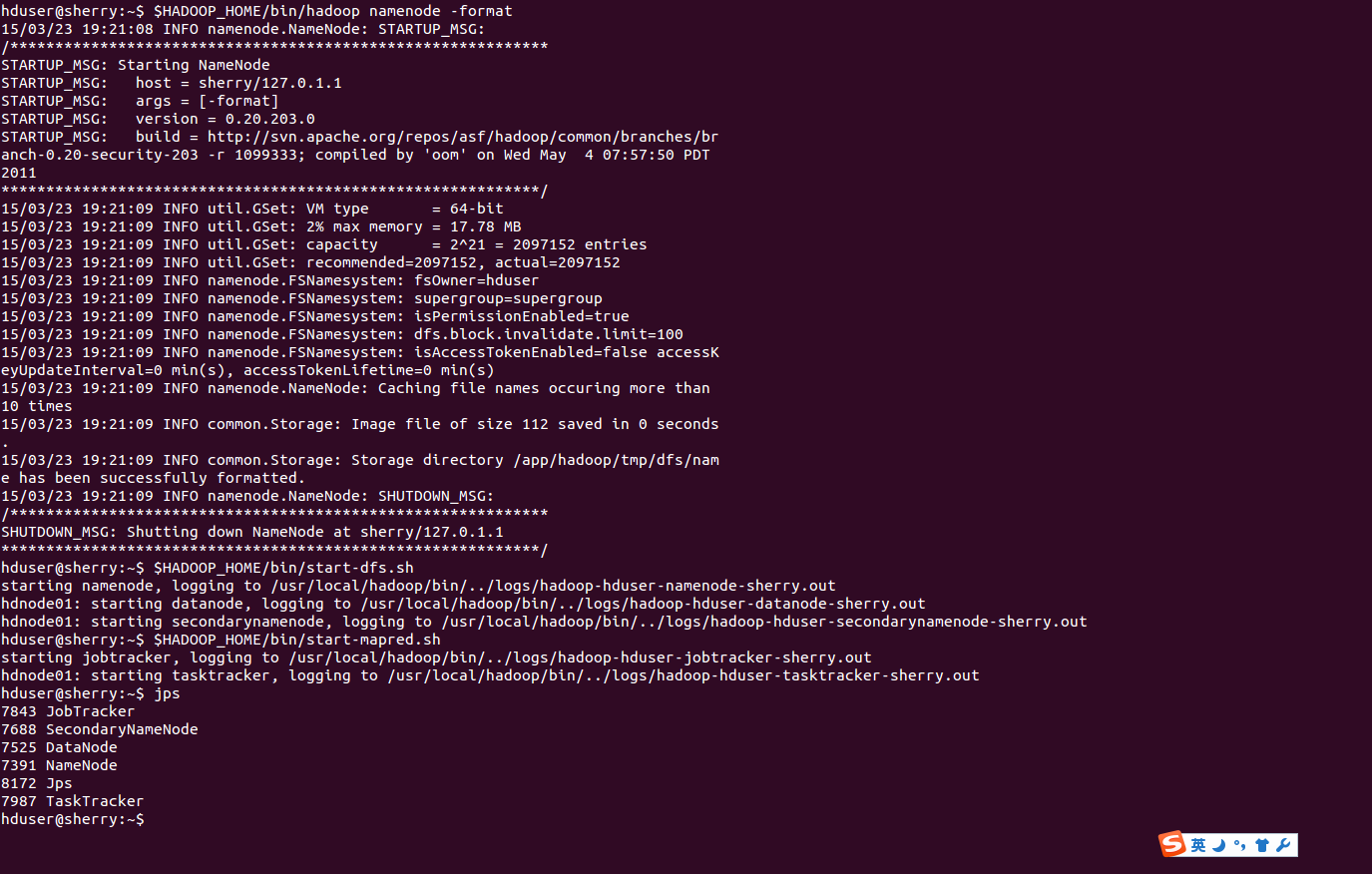
9560 JobTracker

9263 DataNode

9453 SecondaryNameNode

16316 Jps

9745 TaskTracker



可以使用$HADOOP\_HOME/bin/stop-\*.sh脚本终止进程。关于Hadoop的配置工作到这里完成了。

## Running a map/reduce job

接下来我们要运行一个map/reduce任务来测试下安装是否正确。我们使用WordCount这个例子，它从文本文件中读取单词，计算单词一共出现多少次。结果的输出文件也是文本文件，每一行包括一个单词和出现次数，由Tab隔开。这个例子可以从$HADOOP\_HOME/hadoop-examples-0.20.203.0.jar找到。让我们开始吧。

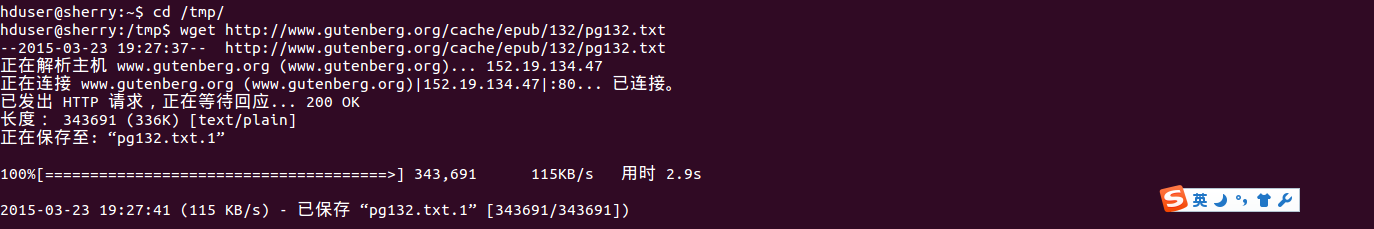
首先，将一个大的UTF-8文本文件下载到临时文件夹中，复制到HDFS，确保复制成功。

cd /tmp/

wget http://www.gutenberg.org/cache/epub/132/pg132.txt

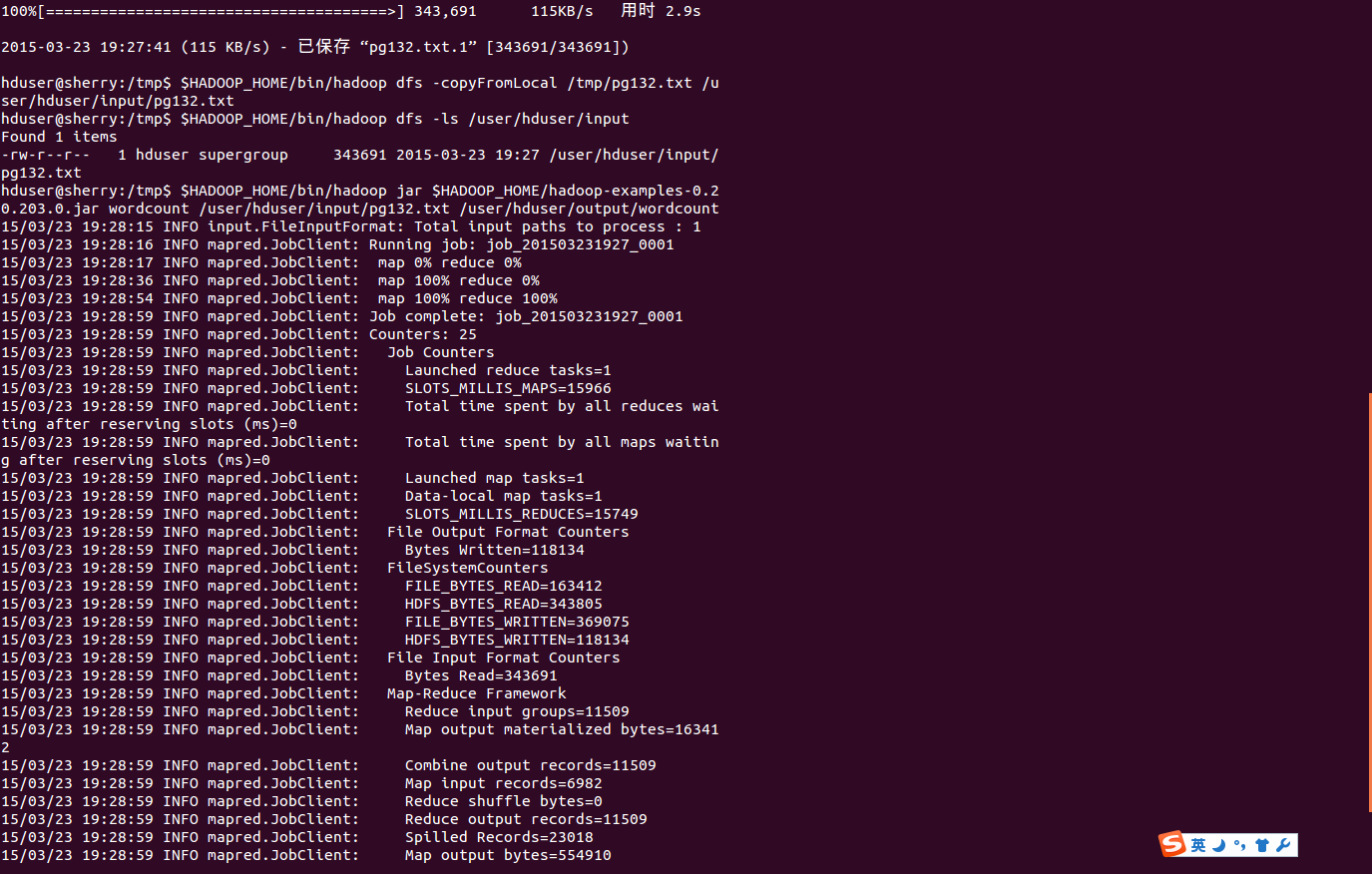
$HADOOP\_HOME/bin/hadoop dfs -copyFromLocal /tmp/pg132.txt /user/hduser/input/pg132.txt

$HADOOP\_HOME/bin/hadoop dfs -ls /user/hduser/input



之后使用如下命令运行wordcount例子。

$HADOOP\_HOME/bin/hadoop jar $HADOOP\_HOME/hadoop-examples-0.20.203.0.jar wordcount /user/hduser/input/pg132.txt /user/hduser/output/wordcount



你可以使用如下的Web地址监测任务的处理过程。

NameNode daemon: <http://hdnode01:50070>

JobTracker daemon: <http://hdnode01:50030>

TaskTracker daemon: <http://hdnode01:50060>

一旦任务完成，你可以通过如下命令查看结果：

$HADOOP\_HOME/bin/hadoop dfs -cat /user/hduser/output/wordcount/p\* | less



## 安装Giraph

接下来我们安装Giraph。首先要安装Git和Maven 3，一定要保证安装Maven 3以上版本。

su – sherry

sudo apt-get install git

sudo apt-get install maven

mvn –version

安装Giraph，从github上clone。

cd /usr/local/

sudo git clone https://github.com/apache/giraph.git

sudo chown -R hduser:hadoop giraph

su - hduser

编辑hduser的$HOME/.bashrc文件，完成后保存：

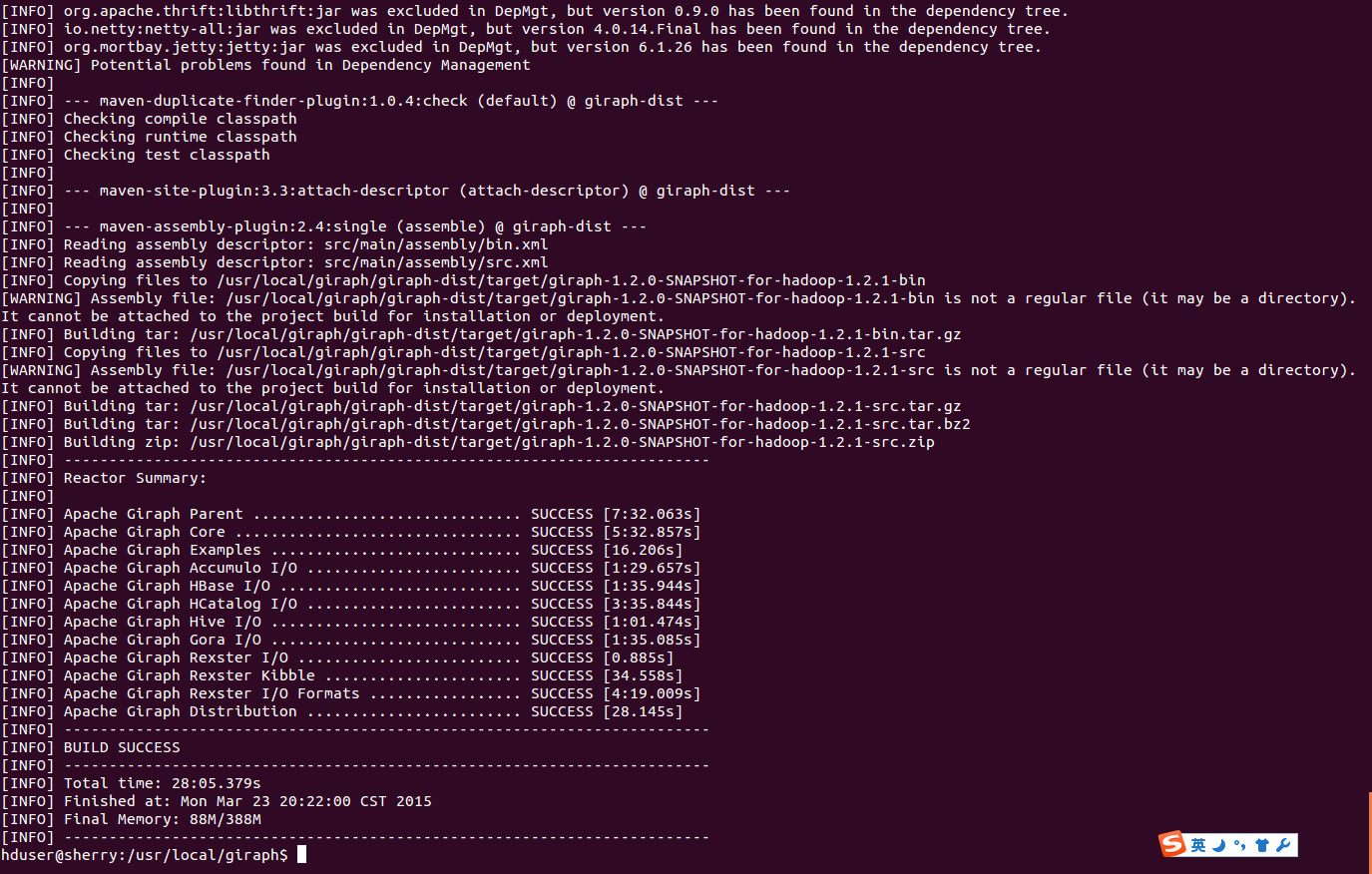
export GIRAPH\_HOME=/usr/local/giraph

使用如下命令将Giraph打包进JAR文件。

source $HOME/.bashrc

cd $GIRAPH\_HOME

mvn package –DskipTests



第一次运行的时候时间比较长，因为Maven要将JAR的插件和其他文件下载到本地。完成后，你会得到Giraph的核心JAR，它存储在$GIRAPH\_HOME/giraph-core/target/giraph-1.1.0-for-hadoop-0.20.203.0-jar-with-dependencies.jar and Giraph examples JAR $GIRAPH\_HOME/giraph-examples/target/giraph-examples-1.1.0-for-hadoop-0.20.203.0-jar-with-dependencies.jar。这样Giraph就安装完成了。

## 运行一个Giraph任务

Giraph和Hadoop安装成功后，就可以运行第一个Giraph任务了。我们会使用SimpleShortestPathsComputation这个例子，它从输入文件中读取图，计算源节点到其他所有节点的最短路径。源节点总是输入文件的第一个节点。首先，在/tmp/tiny\_graph.txt文件中输入一个图。

[0,0,[[1,1],[3,3]]]

[1,0,[[0,1],[2,2],[3,1]]]

[2,0,[[1,2],[4,4]]]

[3,0,[[0,3],[1,1],[4,4]]]

[4,0,[[3,4],[2,4]]]

每一行的格式是：[source\_id,source\_value,[[dest\_id, edge\_value],...]]。在这个图中，一共有5个节点和12条有向边。

$HADOOP\_HOME/bin/hadoop dfs -copyFromLocal /tmp/tiny\_graph.txt /user/hduser/input/tiny\_graph.txt

$HADOOP\_HOME/bin/hadoop dfs -ls /user/hduser/input

运行任务。

$HADOOP\_HOME/bin/hadoop jar $GIRAPH\_HOME/giraph-examples/target/giraph-examples-1.1.0-for-hadoop-0.20.203.0-jar-with-dependencies.jar org.apache.giraph.GiraphRunner org.apache.giraph.examples.SimpleShortestPathsComputation -vif org.apache.giraph.io.formats.JsonLongDoubleFloatDoubleVertexInputFormat -vip /user/hduser/input/tiny\_graph.txt -vof org.apache.giraph.io.formats.IdWithValueTextOutputFormat -op /user/hduser/output/shortestpaths -w 1

像之前一样，你可以使用JobTracker在网页里监测工作过程。任务完成后，使用以下命令查看结果。

$HADOOP\_HOME/bin/hadoop dfs -cat /user/hduser/output/shortestpaths/p\* | less

