HBase实验报告

139 李
193 鞠

实验环境

Ubuntu 14.04

实验内容

- Ubuntu下Java, Hadoop与HBase的安装及伪分布式的部署
- 使用Java API对HBase进行基本操作
- 基于Hadoop实现MapReduce（详见MapReduce.pdf）

配置过程

一、Java的安装

安装JRE

```
sudo apt-get install default-jre
```

安装OpenJDK

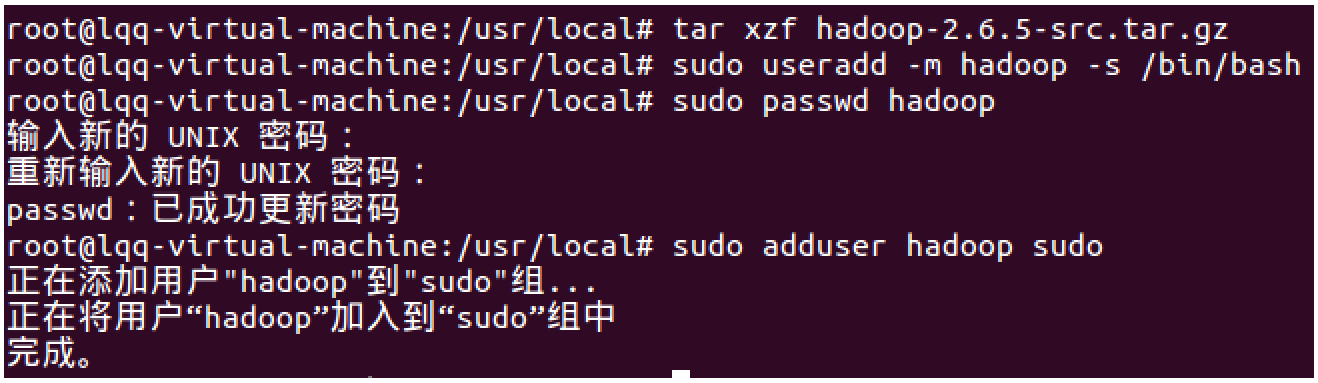
```
sudo apt-get install default-jdk
```

安装OracleJDK

```
sudo add-apt-repository ppa:webupd8team/java
sudo apt-get update
sudo apt-get install oracle-java8-installer
sudo apt-get install oracle-java8-set-default
```

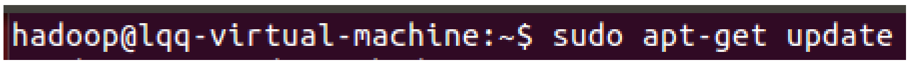
二、Hadoop的安装与伪分布式配置

首先创建一个hadoop用户，并且设置密码，为其增加sudo权限



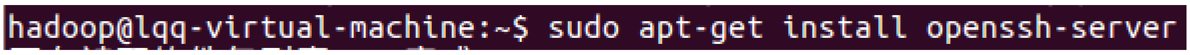
```
root@lqq-virtual-machine:/usr/local# tar xzf hadoop-2.6.5-src.tar.gz
root@lqq-virtual-machine:/usr/local# sudo useradd -m hadoop -s /bin/bash
root@lqq-virtual-machine:/usr/local# sudo passwd hadoop
输入新的 UNIX 密码：
重新输入新的 UNIX 密码：
passwd：已成功更新密码
root@lqq-virtual-machine:/usr/local# sudo adduser hadoop sudo
正在添加用户"hadoop"到"sudo"组...
正在将用户“hadoop”加入到“sudo”组中
完成。
```

使用hadoop用户登录，之后首先更新一下apt



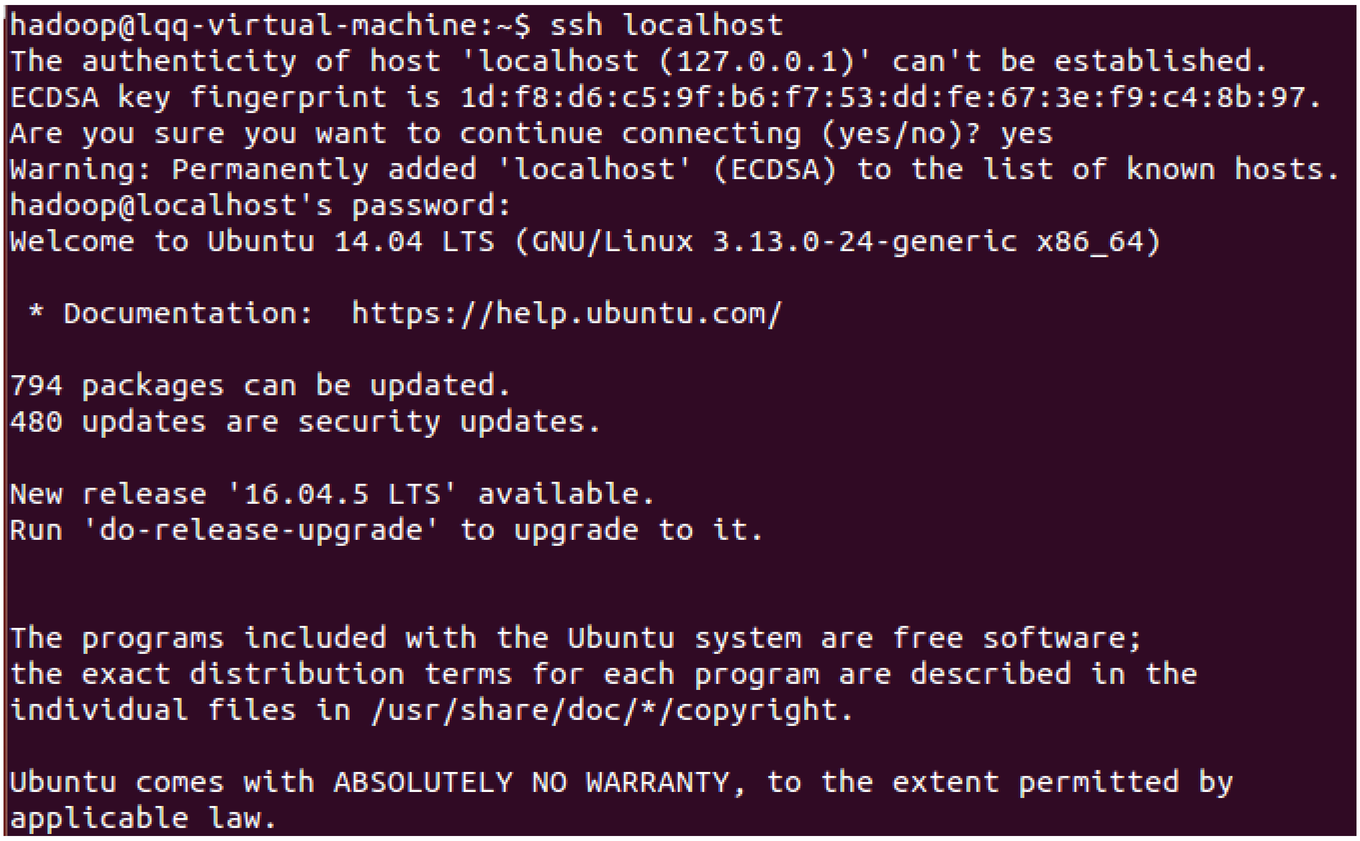
```
hadoop@lqq-virtual-machine:~$ sudo apt-get update
```

集群与单节点模式都需要用到SSH登录，所以安装一下SSH server



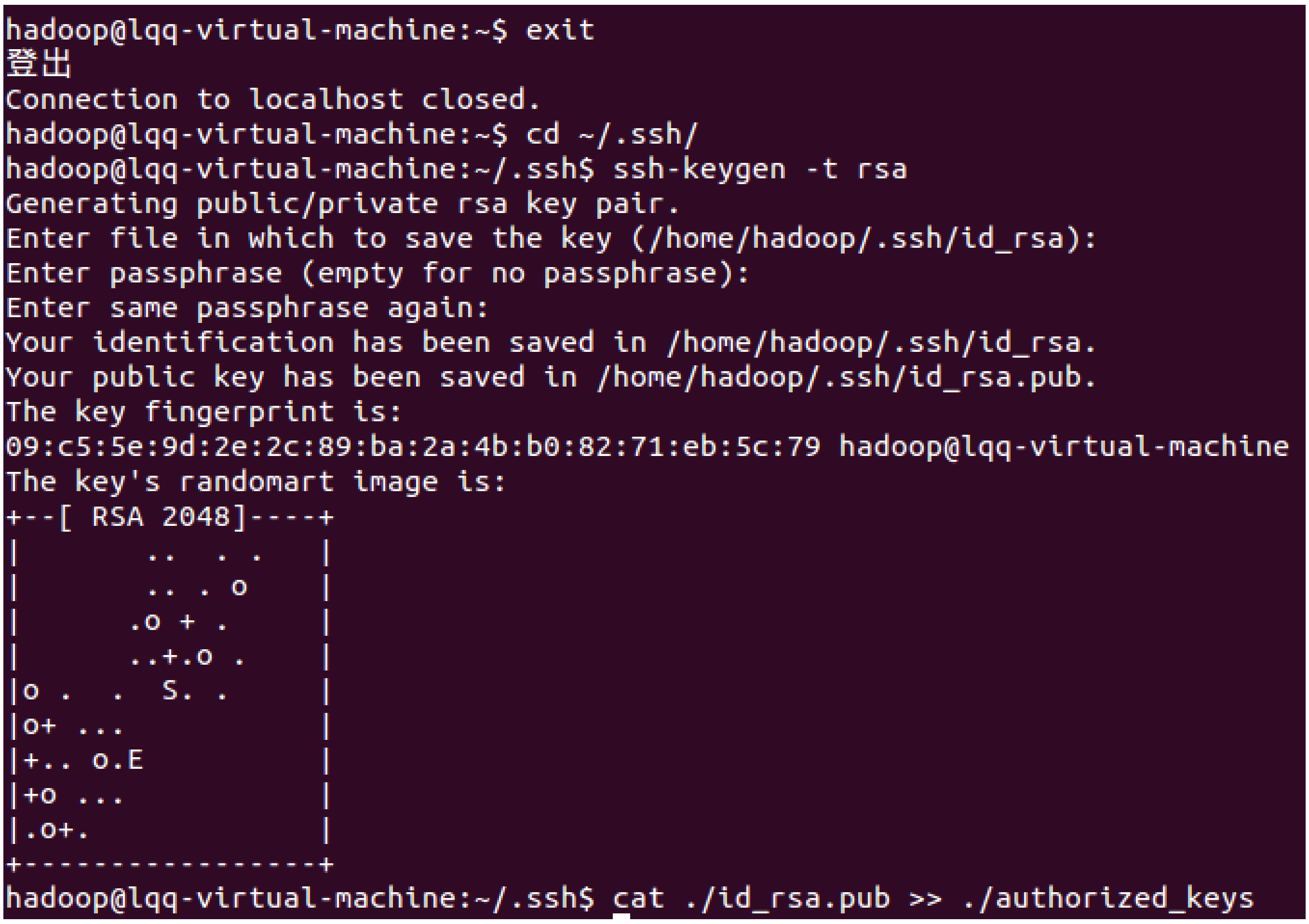
```
hadoop@lqq-virtual-machine:~$ sudo apt-get install openssh-server
```

登录一下本机



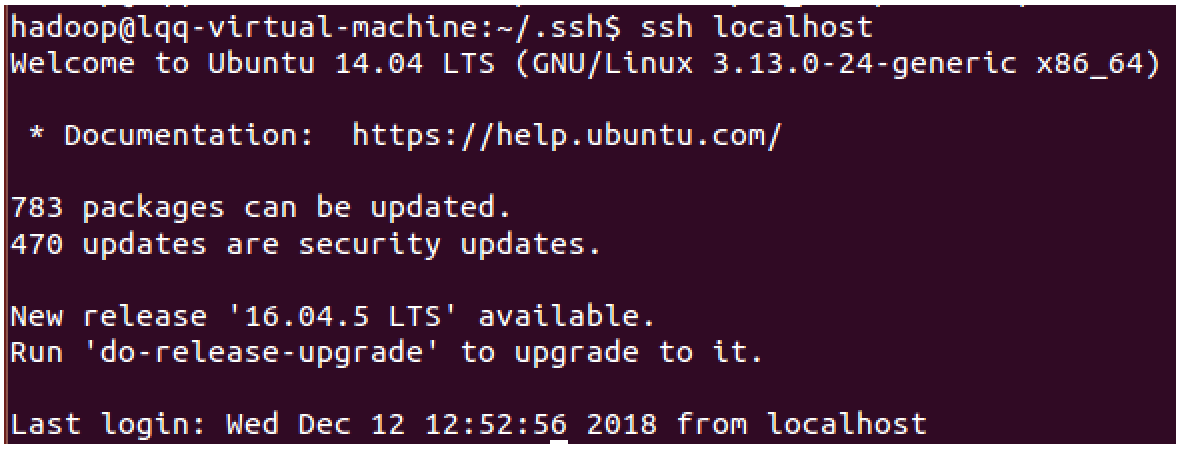
```
hadoop@lqq-virtual-machine:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is 1d:f8:d6:c5:9f:b6:f7:53:dd:fe:67:3e:f9:c4:8b:97.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
hadoop@localhost's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

794 packages can be updated.
480 updates are security updates.

New release '16.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

这次登录需要输入密码。为了方便，将SSH配置成无密码登录
利用ssh-keygen生成密钥，并将密钥加入授权



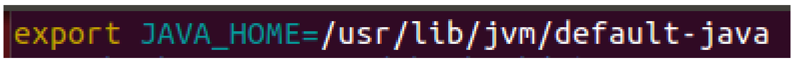
```
hadoop@lqq-virtual-machine:~$ exit
登出
Connection to localhost closed.
hadoop@lqq-virtual-machine:~$ cd ~/.ssh/
hadoop@lqq-virtual-machine:~/.ssh$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa.
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.
The key fingerprint is:
09:c5:5e:9d:2e:2c:89:ba:2a:4b:b0:82:71:eb:5c:79 hadoop@lqq-virtual-machine
The key's randomart image is:
+--[ RSA 2048]----+
|        ..   . . |
|        .. . o   |
|       .o + .    |
|       ..+.o .   |
|o .   .  S. .    |
|o+ ...           |
|+.. o.E          |
|+o ...           |
|.o+.             |
+-----------------+
hadoop@lqq-virtual-machine:~/.ssh$ cat ./id_rsa.pub >> ./authorized_keys
```

这样就可以实现无密码登录



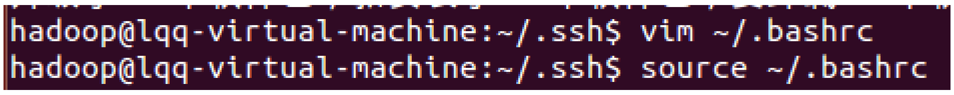
```
hadoop@lqq-virtual-machine:~/.ssh$ ssh localhost
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

783 packages can be updated.
470 updates are security updates.

New release '16.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Dec 12 12:52:56 2018 from localhost
```

之前安装JAVA的时候没有配置环境变量。在~/.bashrc中进行配置



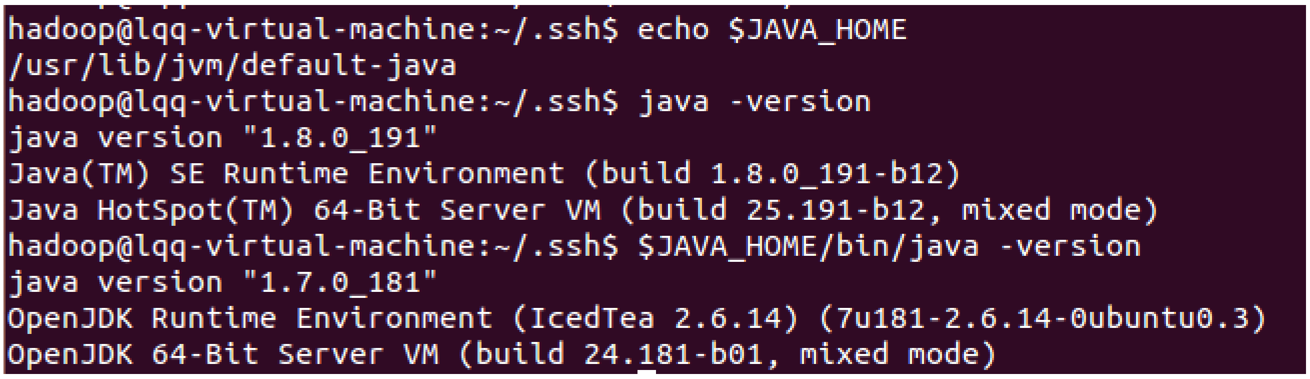
```
export JAVA_HOME=/usr/lib/jvm/default-java
```

编辑完成之后让环境变量生效



```
hadoop@lqq-virtual-machine:~/.ssh$ vim ~/.bashrc
hadoop@lqq-virtual-machine:~/.ssh$ source ~/.bashrc
```

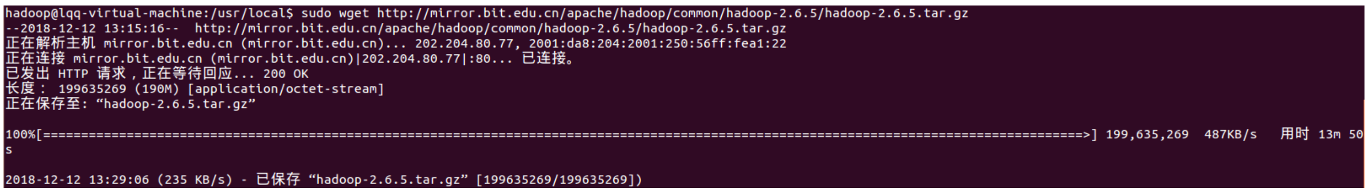
之后对其进行检验



```
hadoop@lqq-virtual-machine:~/.ssh$ echo $JAVA_HOME
/usr/lib/jvm/default-java
hadoop@lqq-virtual-machine:~/.ssh$ java -version
java version "1.8.0_191"
Java(TM) SE Runtime Environment (build 1.8.0_191-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.191-b12, mixed mode)
hadoop@lqq-virtual-machine:~/.ssh$ $JAVA_HOME/bin/java -version
java version "1.7.0_181"
OpenJDK Runtime Environment (IcedTea 2.6.14) (7u181-2.6.14-0ubuntu0.3)
OpenJDK 64-Bit Server VM (build 24.181-b01, mixed mode)
```

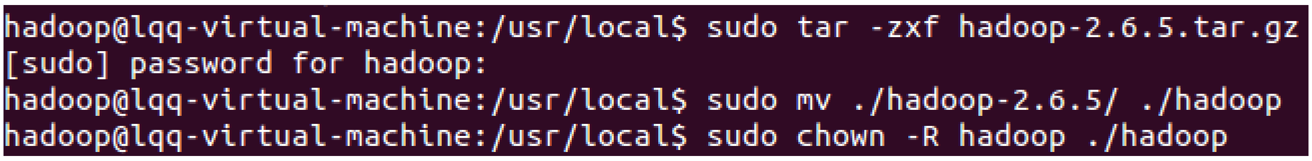
可以看到，配置是正确的

下载hadoop-2.6.5



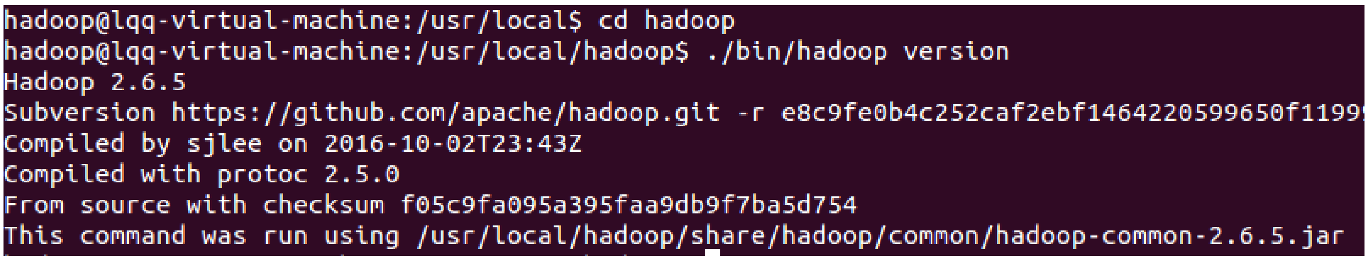
```
hadoop@lqq-virtual-machine:/usr/local$ sudo wget http://mirror.bit.edu.cn/apache/hadoop/common/hadoop-2.6.5/hadoop-2.6.5.tar.gz
--2018-12-12 13:15:16--  http://mirror.bit.edu.cn/apache/hadoop/common/hadoop-2.6.5/hadoop-2.6.5.tar.gz
正在解析主机 mirror.bit.edu.cn (mirror.bit.edu.cn)... 202.204.80.77, 2001:da8:204:2001:250:56ff:fea1:22
正在连接 mirror.bit.edu.cn (mirror.bit.edu.cn)|202.204.80.77|:80... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度： 199635269 (190M) [application/octet-stream]
正在保存至: “hadoop-2.6.5.tar.gz”

100%[==================================================================================================================>] 199,635,269  487KB/s   用时 13m 50
s

2018-12-12 13:29:06 (235 KB/s) - 已保存 “hadoop-2.6.5.tar.gz” [199635269/199635269])
```

将压缩包进行解压，修改文件夹名称为hadoop，并且修改文件权限



```
hadoop@lqq-virtual-machine:/usr/local$ sudo tar -zxf hadoop-2.6.5.tar.gz
[sudo] password for hadoop:
hadoop@lqq-virtual-machine:/usr/local$ sudo mv ./hadoop-2.6.5/ ./hadoop
hadoop@lqq-virtual-machine:/usr/local$ sudo chown -R hadoop ./hadoop
```

接下来检验一下hadoop是否可用

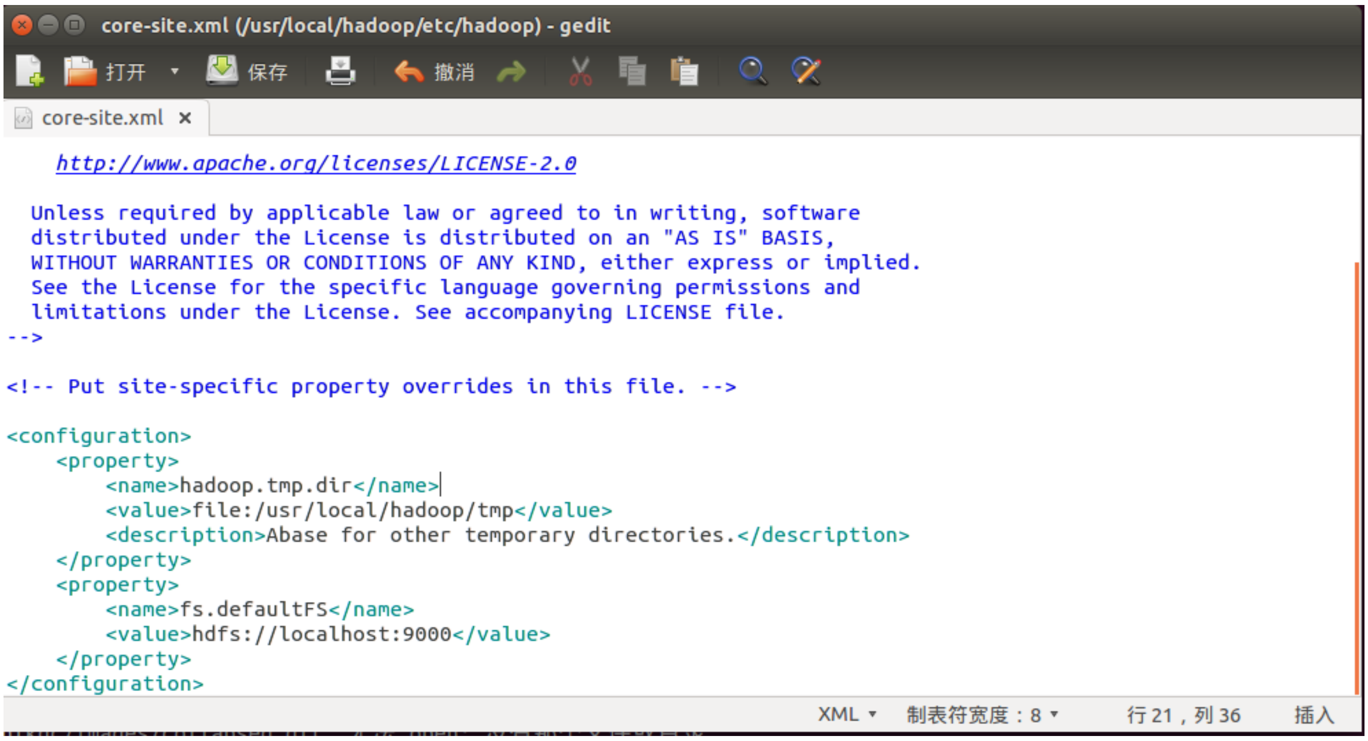


```
hadoop@lqq-virtual-machine:/usr/local$ cd hadoop
hadoop@lqq-virtual-machine:/usr/local/hadoop$ ./bin/hadoop version
Hadoop 2.6.5
Subversion https://github.com/apache/hadoop.git -r e8c9fe0b4c252caf2ebf1464220599650f1199
Compiled by sjlee on 2016-10-02T23:43Z
Compiled with protoc 2.5.0
From source with checksum f05c9fa095a395faa9db9f7ba5d754
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-2.6.5.jar
```

检验成功，显示当前hadoop版本信息

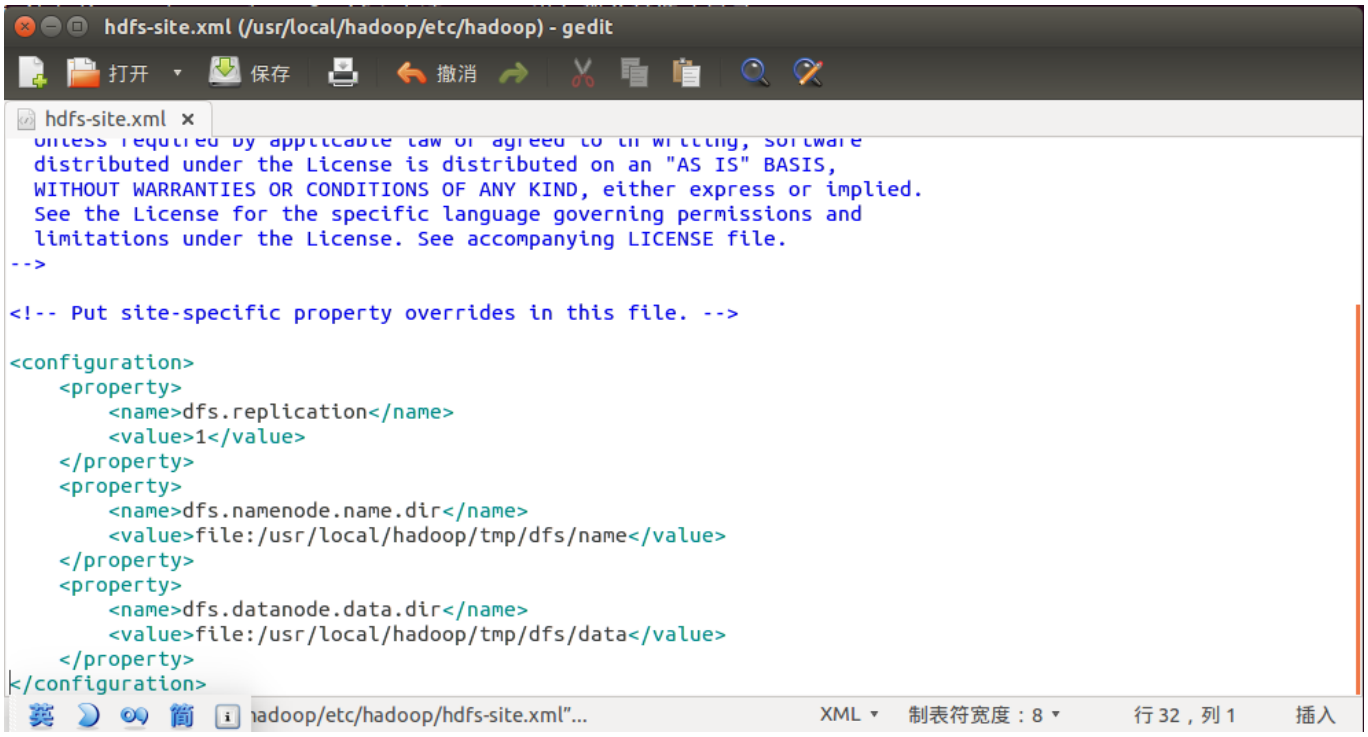
接下来进行伪分布式的配置。Hadoop可以在单节点上以伪分布式的方式运行。Hadoop进程以分离的Java进程来运行。需要对hadoop/etc/hadoop中的两个配置文件进行修改

对core-site.xml进行修改



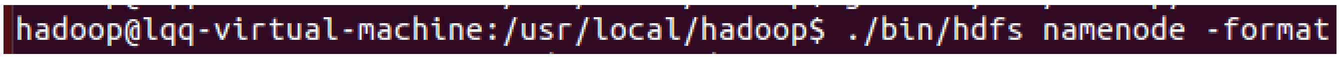
```
    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
        <name>hadoop.tmp.dir</name>
        <value>file:/usr/local/hadoop/tmp</value>
        <description>Abase for other temporary directories.</description>
    </property>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

对hdfs-site.xml进行修改

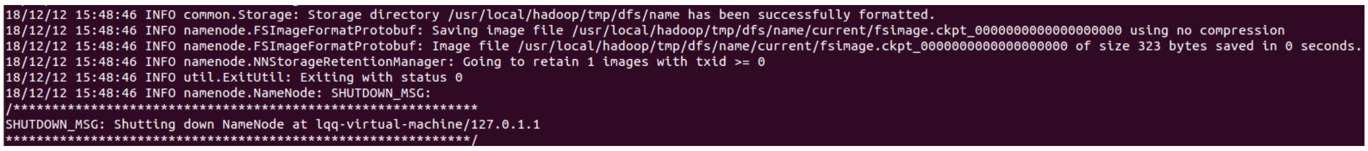

hdfs-site.xml (/usr/local/hadoop/etc/hadoop) - gedit
打开
保存
撤消
hdfs-site.xml
adoop/etc/hadoop/hdfs-site.xml"...
XML
制表符宽度：8
行 32，列 1
插入

```
  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>file:/usr/local/hadoop/tmp/dfs/name</value>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>file:/usr/local/hadoop/tmp/dfs/data</value>
    </property>
</configuration>
```

修改完成后对NameNode进行格式化



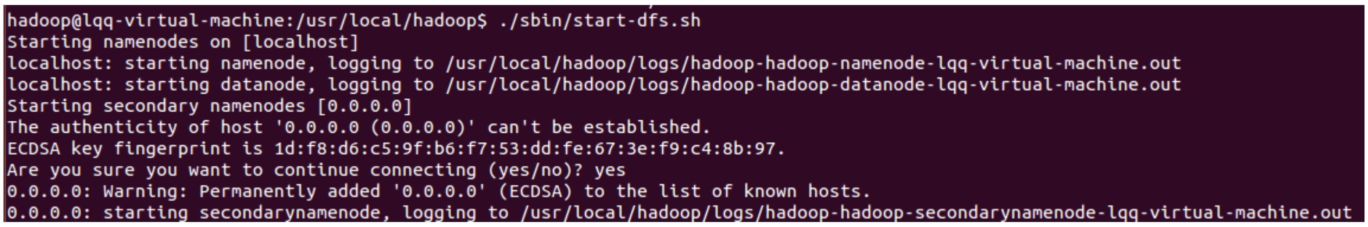
```
hadoop@lqq-virtual-machine:/usr/local/hadoop$ ./bin/hdfs namenode -format
```



```
18/12/12 15:48:46 INFO common.Storage: Storage directory /usr/local/hadoop/tmp/dfs/name has been successfully formatted.
18/12/12 15:48:46 INFO namenode.FSImageFormatProtobuf: Saving image file /usr/local/hadoop/tmp/dfs/name/current/fsimage.ckpt_0000000000000000000 using no compression
18/12/12 15:48:46 INFO namenode.FSImageFormatProtobuf: Image file /usr/local/hadoop/tmp/dfs/name/current/fsimage.ckpt_0000000000000000000 of size 323 bytes saved in 0 seconds.
18/12/12 15:48:46 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
18/12/12 15:48:46 INFO util.ExitUtil: Exiting with status 0
18/12/12 15:48:46 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at lqq-virtual-machine/127.0.1.1
************************************************************/
```

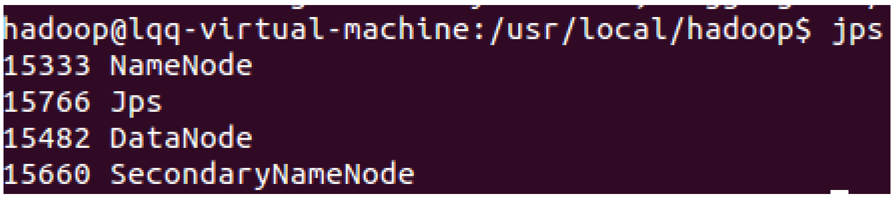
可以看到successfully formatted和Exitting with status 0的提示，说明配置成功

下面开启NameNode和DataNode守护进程



```
hadoop@lqq-virtual-machine:/usr/local/hadoop$ ./sbin/start-dfs.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hadoop-namenode-lqq-virtual-machine.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hadoop-datanode-lqq-virtual-machine.out
Starting secondary namenodes [0.0.0.0]
The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
ECDSA key fingerprint is 1d:f8:d6:c5:9f:b6:f7:53:dd:fe:67:3e:f9:c4:8b:97.
Are you sure you want to continue connecting (yes/no)? yes
0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts.
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hadoop-secondarynamenode-lqq-virtual-machine.out
```

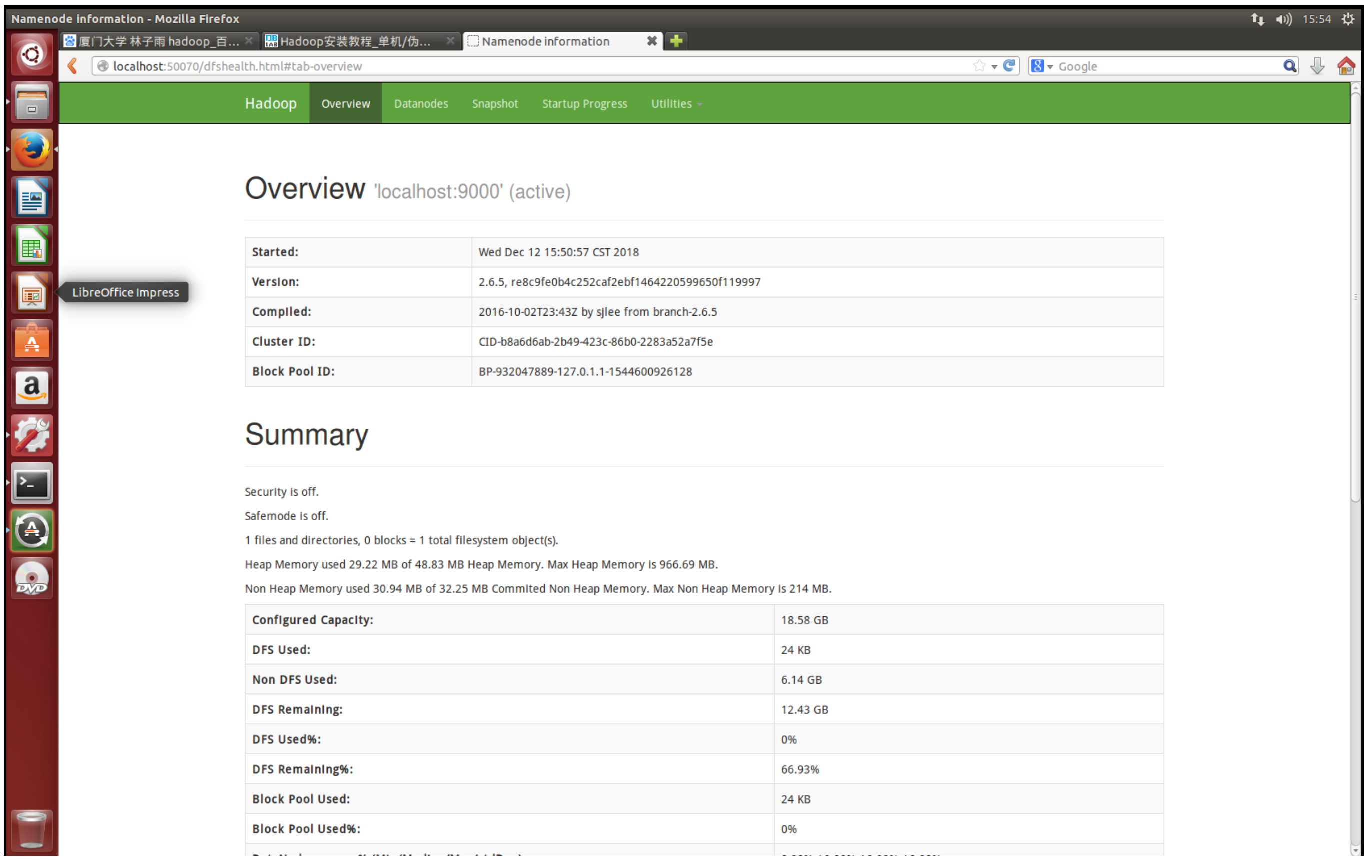
使用jps命令来判断是否启动成功



```
hadoop@lqq-virtual-machine:/usr/local/hadoop$ jps
15333 NameNode
15766 Jps
15482 DataNode
15660 SecondaryNameNode
```

可以看到，出现了NameNode, DataNode和SecondaryNameNode三个进程，说明启动成功

在50070端口还可以查看节点信息


Namenode information - Mozilla Firefox
15:54
厦门大学 林子雨 hadoop_百...
Hadoop安装教程_单机/伪...
Namenode information
localhost:50070/dfshealth.html#tab-overview
Google
Hadoop
Overview
Datanodes
Snapshot
Startup Progress
Utilities
LibreOffice Impress

Overview 'localhost:9000' (active)

Started:	Wed Dec 12 15:50:57 CST 2018
Version:	2.6.5, re8c9fe0b4c252caf2ebf1464220599650f119997
Compiled:	2016-10-02T23:43Z by sjlee from branch-2.6.5
Cluster ID:	CID-b8a6d6ab-2b49-423c-86b0-2283a52a7f5e
Block Pool ID:	BP-932047889-127.0.1.1-1544600926128

Summary

Security is off.

Safemode is off.

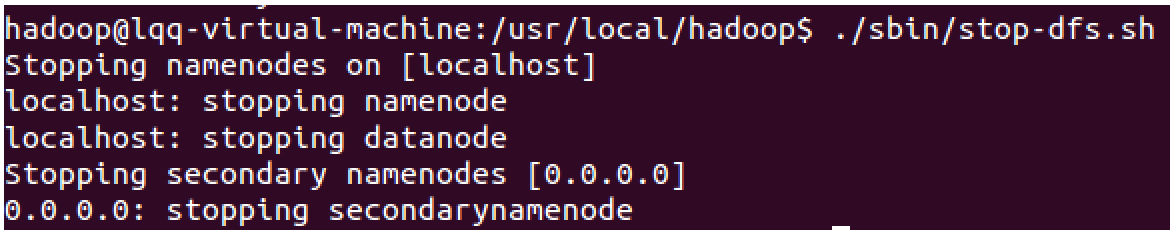
1 files and directories, 0 blocks = 1 total filesystem object(s).

Heap Memory used 29.22 MB of 48.83 MB Heap Memory. Max Heap Memory is 966.69 MB.

Non Heap Memory used 30.94 MB of 32.25 MB Commited Non Heap Memory. Max Non Heap Memory is 214 MB.

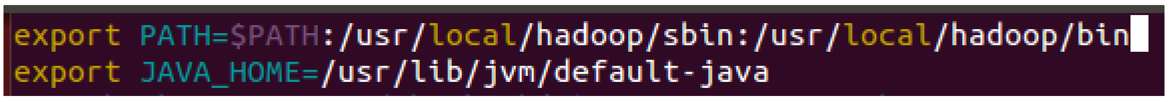
Configured Capacity:	18.58 GB
DFS Used:	24 KB
Non DFS Used:	6.14 GB
DFS Remaining:	12.43 GB
DFS Used%:	0%
DFS Remaining%:	66.93%
Block Pool Used:	24 KB
Block Pool Used%:	0%

关闭Hadoop的命令



```
hadoop@lqq-virtual-machine:/usr/local/hadoop$ ./sbin/stop-dfs.sh
Stopping namenodes on [localhost]
localhost: stopping namenode
localhost: stopping datanode
Stopping secondary namenodes [0.0.0.0]
0.0.0.0: stopping secondarynamenode
```

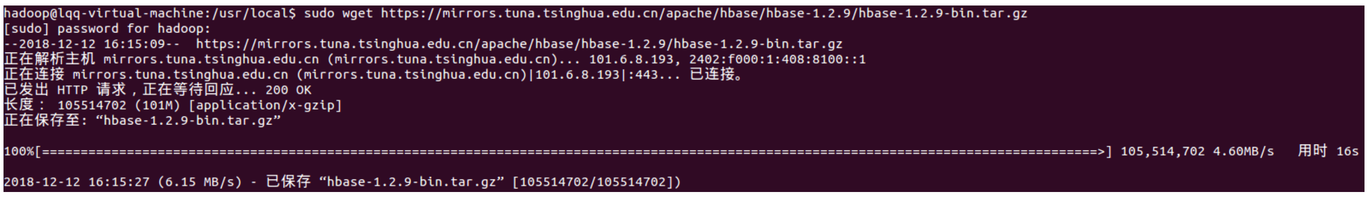
将Hadoop加入环境变量，这样在任意目录中都可以使用hdfs命令



```
export PATH=$PATH:/usr/local/hadoop/sbin:/usr/local/hadoop/bin
export JAVA_HOME=/usr/lib/jvm/default-java
```

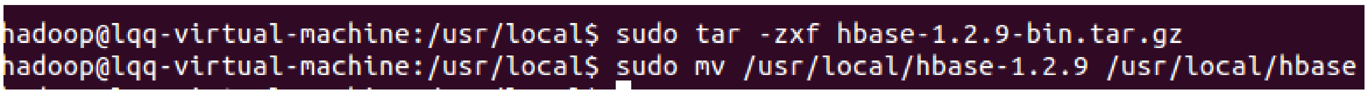
三、HBase的安装与伪分布式配置

下载HBase-1.2.9



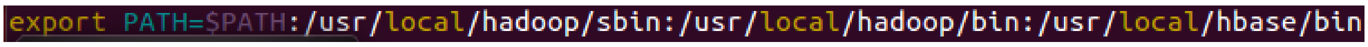
```
hadoop@lqq-virtual-machine:/usr/local$ sudo wget https://mirrors.tuna.tsinghua.edu.cn/apache/hbase/hbase-1.2.9/hbase-1.2.9-bin.tar.gz
[sudo] password for hadoop:
--2018-12-12 16:15:09--  https://mirrors.tuna.tsinghua.edu.cn/apache/hbase/hbase-1.2.9/hbase-1.2.9-bin.tar.gz
正在解析主机 mirrors.tuna.tsinghua.edu.cn (mirrors.tuna.tsinghua.edu.cn)... 101.6.8.193, 2402:f000:1:408:8100::1
正在连接 mirrors.tuna.tsinghua.edu.cn (mirrors.tuna.tsinghua.edu.cn)|101.6.8.193|:443... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度： 105514702 (101M) [application/x-gzip]
正在保存至: “hbase-1.2.9-bin.tar.gz”

100%[===================================================================================================================>] 105,514,702 4.60MB/s   用时 16s

2018-12-12 16:15:27 (6.15 MB/s) - 已保存 “hbase-1.2.9-bin.tar.gz” [105514702/105514702])
```

解压并修改文件名

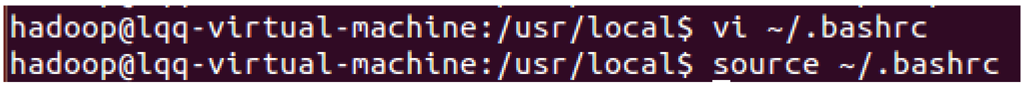


```
hadoop@lqq-virtual-machine:/usr/local$ sudo tar -zxf hbase-1.2.9-bin.tar.gz
hadoop@lqq-virtual-machine:/usr/local$ sudo mv /usr/local/hbase-1.2.9 /usr/local/hbase
```

添加环境变量并使其生效

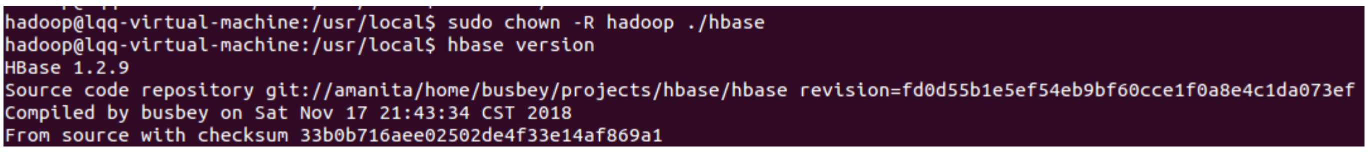


```
export PATH=$PATH:/usr/local/hadoop/sbin:/usr/local/hadoop/bin:/usr/local/hbase/bin
```



```
hadoop@lqq-virtual-machine:/usr/local$ vi ~/.bashrc
hadoop@lqq-virtual-machine:/usr/local$ source ~/.bashrc
```

添加HBase权限并查看HBase版本以检验是否安装成功

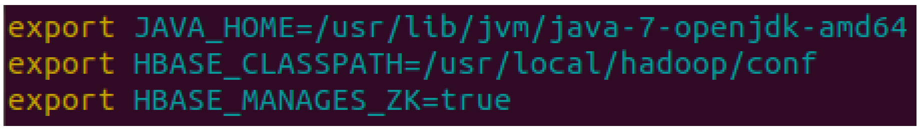


```
hadoop@lqq-virtual-machine:/usr/local$ sudo chown -R hadoop ./hbase
hadoop@lqq-virtual-machine:/usr/local$ hbase version
HBase 1.2.9
Source code repository git://amanita/home/busbey/projects/hbase/hbase revision=fd0d55b1e5ef54eb9bf60cce1f0a8e4c1da073ef
Compiled by busbey on Sat Nov 17 21:43:34 CST 2018
From source with checksum 33b0b716aee02502de4f33e14af869a1
```

可以看到，显示HBase的版本，安装成功

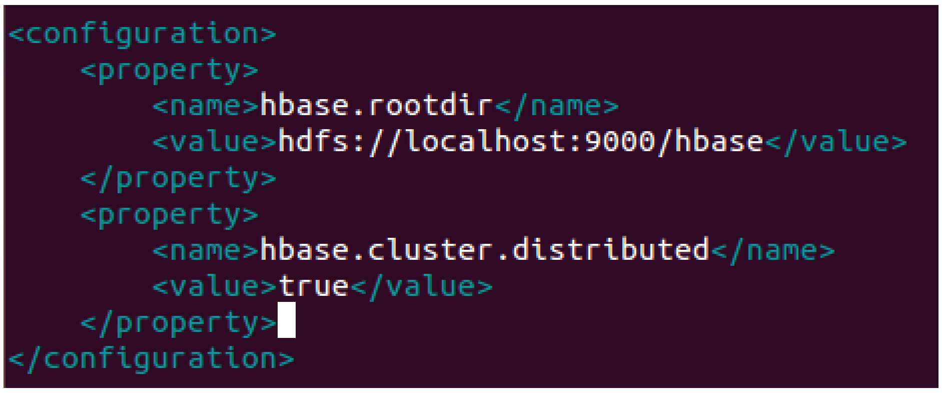
接下来进行伪分布式的配置

首先在hbase-env.sh中进行JAVA_HOME, HBASE_CLASSPATH和HBASE_MANAGES_ZK这些环境变量的配置



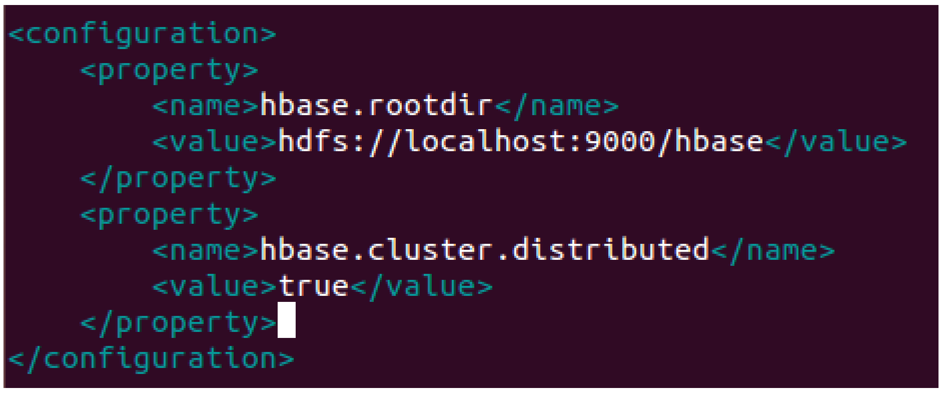
```
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
export HBASE_CLASSPATH=/usr/local/hadoop/conf
export HBASE_MANAGES_ZK=true
```

接下来修改hbase-site.xml



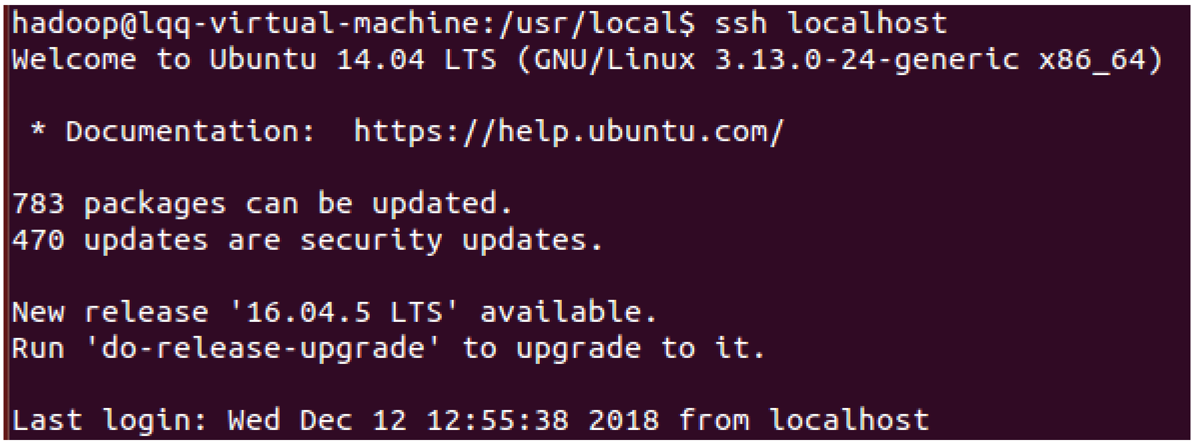
```
<configuration>
    <property>
        <name>hbase.rootdir</name>
        <value>hdfs://localhost:9000/hbase</value>
    </property>
    <property>
        <name>hbase.cluster.distributed</name>
        <value>true</value>
    </property>
</configuration>
```

修改hbase.rootdir，指定HBase数据在HDFS上的存储路径；将属性hbase.cluter.distributed设置为true。假设当前Hadoop集群运行在伪分布式模式下，在本机上运行，且NameNode运行在9000端口。其中hbase.rootdir指定HBase的存储目录，hbase.cluster.distributed设置集群处于分布式模式



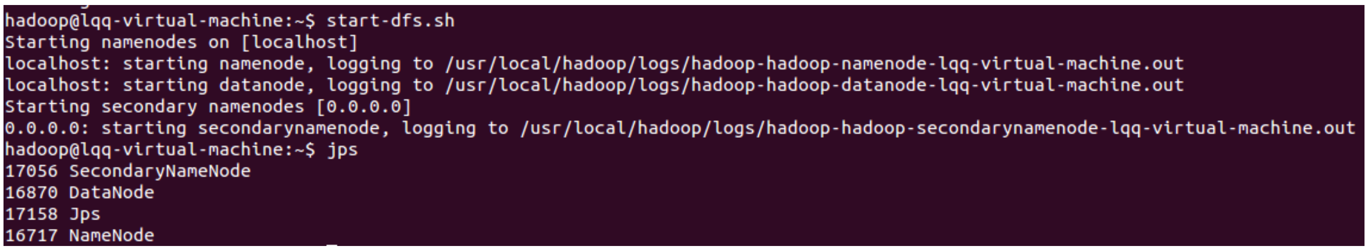
```
<configuration>
    <property>
        <name>hbase.rootdir</name>
        <value>hdfs://localhost:9000/hbase</value>
    </property>
    <property>
        <name>hbase.cluster.distributed</name>
        <value>true</value>
    </property>
</configuration>
```

下面运行HBase进行测试。首先登录ssh



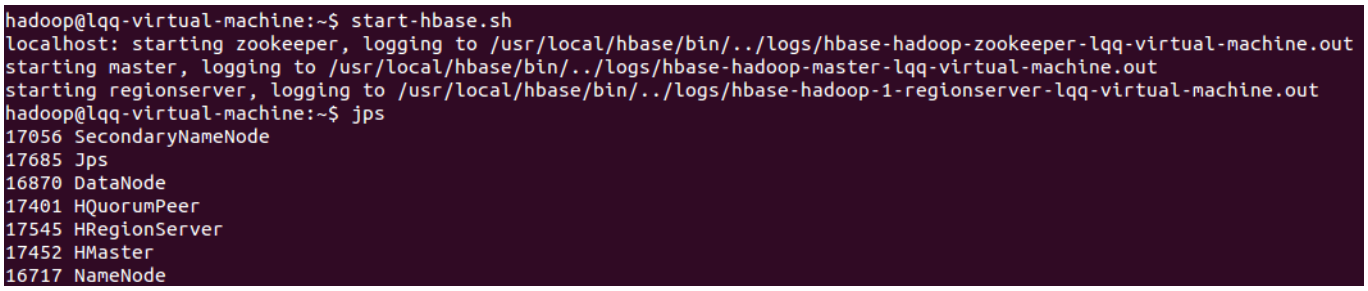
```
hadoop@lqq-virtual-machine:/usr/local$ ssh localhost
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

783 packages can be updated.
470 updates are security updates.

New release '16.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Dec 12 12:55:38 2018 from localhost
```

然后启动Hadoop



```
hadoop@lqq-virtual-machine:~$ start-dfs.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hadoop-namenode-lqq-virtual-machine.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hadoop-datanode-lqq-virtual-machine.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hadoop-secondarynamenode-lqq-virtual-machine.out
hadoop@lqq-virtual-machine:~$ jps
17056 SecondaryNameNode
16870 DataNode
17158 Jps
16717 NameNode
```

可以看到Hadoop启动成功。再启动HBase



```
hadoop@lqq-virtual-machine:~$ start-hbase.sh
localhost: starting zookeeper, logging to /usr/local/hbase/bin/../logs/hbase-hadoop-zookeeper-lqq-virtual-machine.out
starting master, logging to /usr/local/hbase/bin/../logs/hbase-hadoop-master-lqq-virtual-machine.out
starting regionserver, logging to /usr/local/hbase/bin/../logs/hbase-hadoop-1-regionserver-lqq-virtual-machine.out
hadoop@lqq-virtual-machine:~$ jps
17056 SecondaryNameNode
17685 Jps
16870 DataNode
17401 HQuorumPeer
17545 HRegionServer
17452 HMaster
16717 NameNode
```

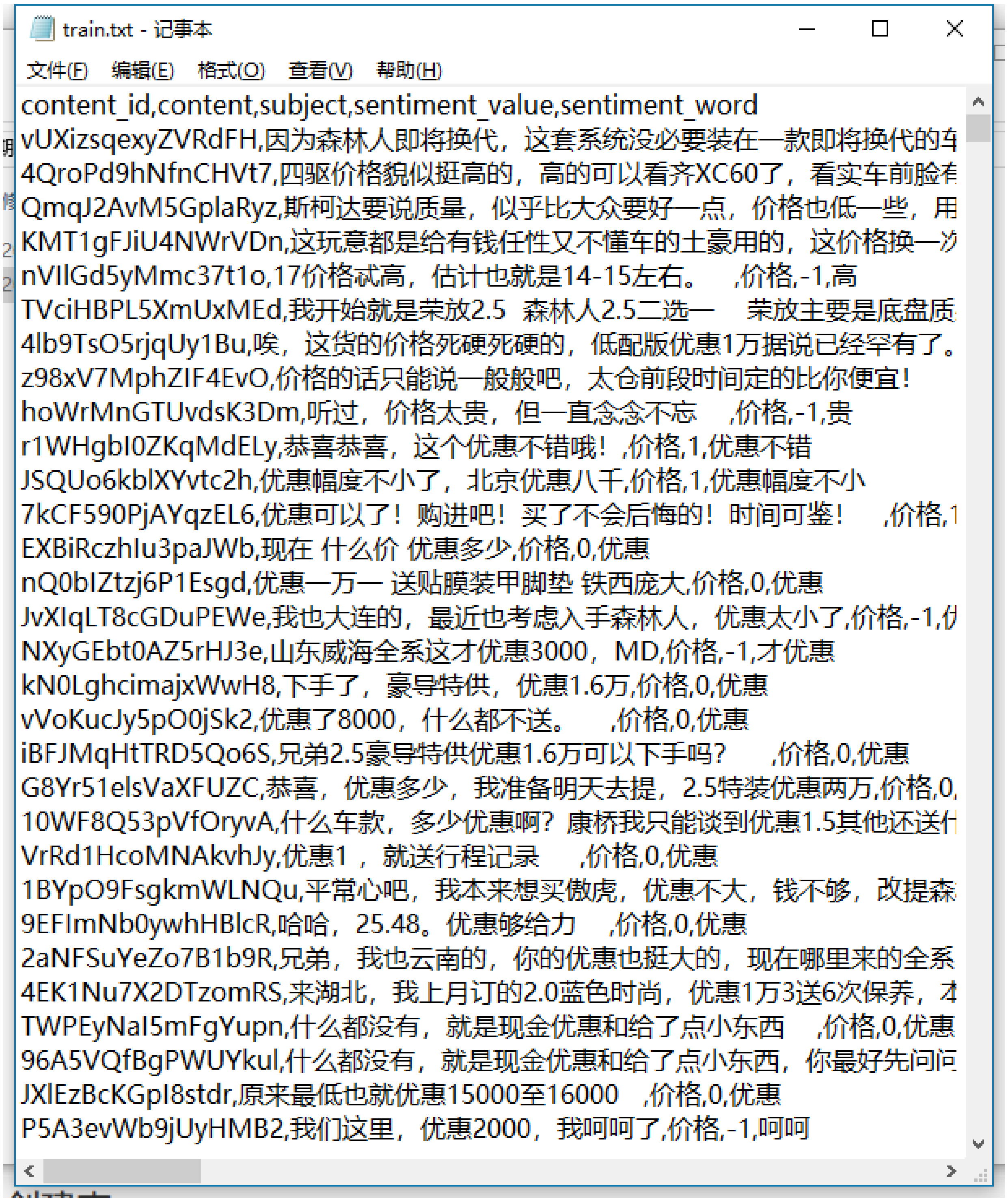
可以看到HBase也启动成功

至此，环境全部配置完成

使用Java API对HBase进行操作

一、数据集

使用的是CCF汽车评论的训练数据，有将近1w条。



```
content_id,content,subject,sentiment_value,sentiment_word
vUXizsqexyZVRdFH,因为森林人即将换代，这套系统没必要装在一款即将换代的车
4QroPd9hNfnCHVt7,四驱价格貌似挺高的，高的可以看齐XC60了，看实车前脸有
QmqJ2AvM5GplaRyz,斯柯达要说质量，似乎比大众要好一点，价格也低一些，用
KMT1gFJiU4NWrVDn,这玩意都是给有钱任性又不懂车的土豪用的，这价格换一次
nVIlGd5yMmc37t1o,17价格忒高，估计也就是14-15左右。    ,价格,-1,高
TVciHBPL5XmUxMEd,我开始就是荣放2.5  森林人2.5二选一    荣放主要是底盘质
4lb9TsO5rjqUy1Bu,唉，这货的价格死硬死硬的，低配版优惠1万据说已经罕有了。
z98xV7MphZIF4EvO,价格的话只能说一般般吧，太仓前段时间定的比你便宜！
hoWrMnGTUvdsK3Dm,听过，价格太贵，但一直念念不忘    ,价格,-1,贵
r1WHgbI0ZKqMdELy,恭喜恭喜，这个优惠不错哦！,价格,1,优惠不错
JSQUo6kblXYvtc2h,优惠幅度不小了，北京优惠八千,价格,1,优惠幅度不小
7kCF590PjAYqzEL6,优惠可以了！购进吧！买了不会后悔的！时间可鉴！    ,价格,1
EXBiRczhlu3paJWb,现在 什么价 优惠多少,价格,0,优惠
nQ0bIZtzj6P1Esgd,优惠一万一 送贴膜装甲脚垫 铁西庞大,价格,0,优惠
JvXIqLT8cGDuPEWe,我也大连的，最近也考虑入手森林人，优惠太小了,价格,-1,优
NXyGEbt0AZ5rHJ3e,山东威海全系这才优惠3000，MD,价格,-1,才优惠
kN0LghcimajxWwH8,下手了，豪导特供，优惠1.6万,价格,0,优惠
vVoKucJy5pO0jSk2,优惠了8000，什么都不送。    ,价格,0,优惠
iBFJMqHtTRD5Qo6S,兄弟2.5豪导特供优惠1.6万可以下手吗？    ,价格,0,优惠
G8Yr51elsVaXFUZC,恭喜，优惠多少，我准备明天去提，2.5特装优惠两万,价格,0,
10WF8Q53pVfOryvA,什么车款，多少优惠啊？康桥我只能谈到优惠1.5其他还送什
VrRd1HcoMNAkvhJy,优惠1 ，就送行程记录    ,价格,0,优惠
1BYpO9FsgkmWLNQu,平常心吧，我本来想买傲虎，优惠不大，钱不够，改提森
9EFImNb0ywhHBlcR,哈哈，25.48。优惠够给力    ,价格,0,优惠
2aNFSuYeZo7B1b9R,兄弟，我也云南的，你的优惠也挺大的，现在哪里来的全系
4EK1Nu7X2DTzomRS,来湖北，我上月订的2.0蓝色时尚，优惠1万3送6次保养，本
TWPEyNal5mFgYupn,什么都没有，就是现金优惠和给了点小东西    ,价格,0,优惠
96A5VQfBgPWUYkul,什么都没有，就是现金优惠和给了点小东西，你最好先问问
JXlEzBcKGpl8stdr,原来最低也就优惠15000至16000    ,价格,0,优惠
P5A3evWb9jUyHMB2,我们这里，优惠2000，我呵呵了,价格,-1,呵呵
```

二、程序结构

```
- HBaseTest/src
  - HBaseUtility.java  // 基本操作实现
  - Test.java  // 测试用类
```

三、基本操作实现

- 建立连接

```java
// create connection
public static void init() {
    configuration = HBaseConfiguration.create();
    configuration.set("hbase.rootdir", "hdfs://localhost:9000/hbase");
    try {
        connection = ConnectionFactory.createConnection(configuration);
        admin = connection.getAdmin();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

- 关闭连接

```java
// close connection
public static void close() {
    try {
        if (admin != null) {
            admin.close();
        }
        if (null != connection) {
            connection.close();
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

- 按照表名和各列族名创建新表

```java
public static void createTable(String myTableName, String[] colFamily) throws IOException {
    init();
    TableName tableName = TableName.valueOf(myTableName);

    if (admin.tableExists(tableName)) {  // table has existed
        System.out.println("Table is exists!");
    } else {
        HTableDescriptor hTableDescriptor = new HTableDescriptor(tableName);
        for (String str:colFamily) {
            HColumnDescriptor hColumnDescriptor = new HColumnDescriptor(str);
            hTableDescriptor.addFamily(hColumnDescriptor);
        }
        admin.createTable(hTableDescriptor);
        System.out.println("Create table success");
    }
    close();
}
```

- 根据表名删除表

```
public static void deleteTable(String tableName) throws IOException {
    init();
    TableName tn = TableName.valueOf(tableName);
    if (admin.tableExists(tn)) {
        admin.disableTable(tn);
        admin.deleteTable(tn);
    }
    close();
}
```

- 遍历已经存在的表

```
public static void listTables() throws IOException {
    init();
    HTableDescriptor hTableDescriptors[] = admin.listTables();
    for (HTableDescriptor hTableDescriptor:hTableDescriptors) {  // for every table in the list
        System.out.println(hTableDescriptor.getNameAsString());
    }
    close();
}
```

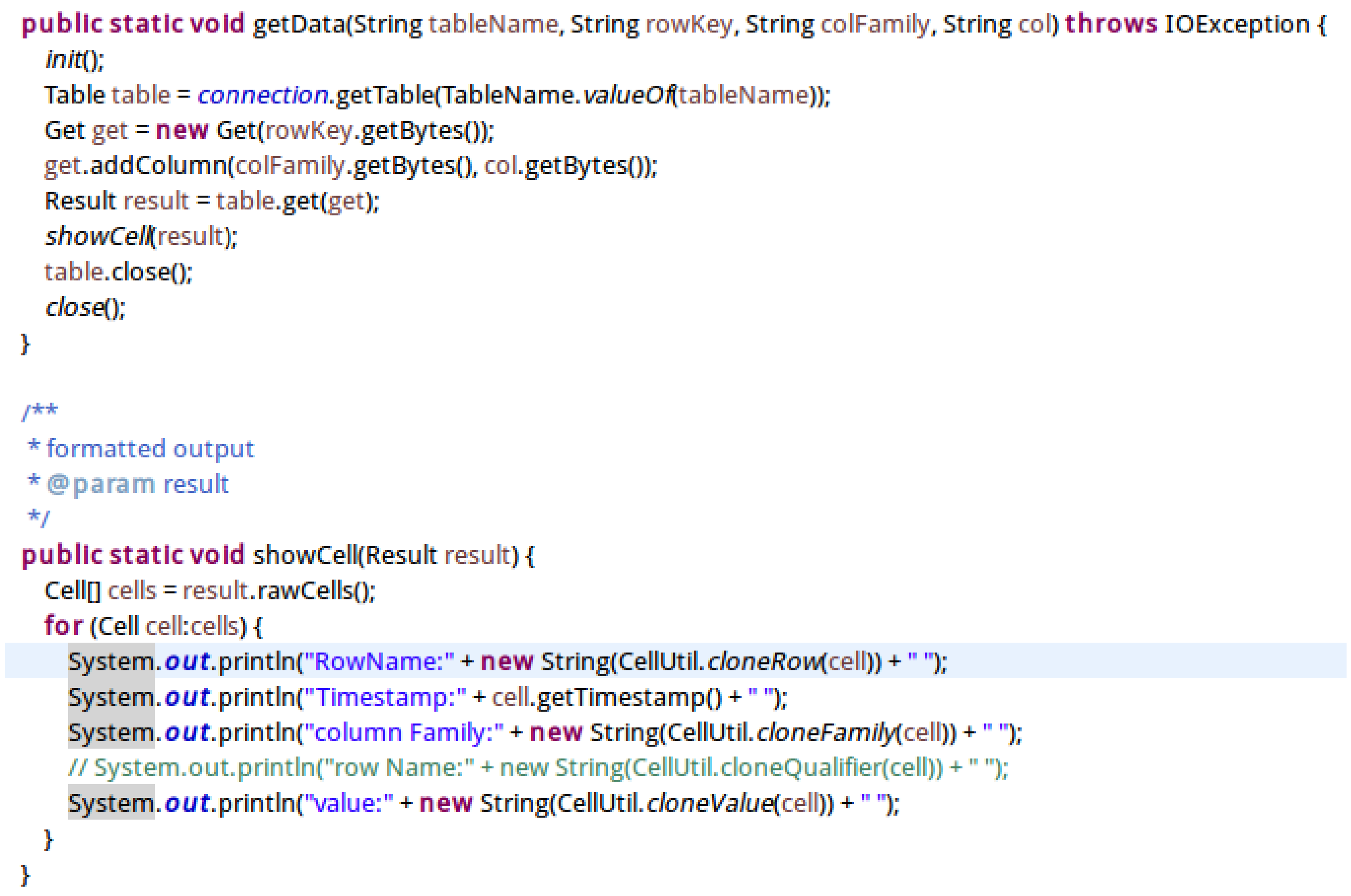
- 插入数据。这里的rowKey不作为表中的列族出现。插入的时候需要给定列的名字，也就是说不是一行一行插入，而是插入指定的rowKey行colFamily列col子列，相当于一格一格插入。子列col可以是不存在的

```
public static void insertRow(String tableName, String rowKey, String colFamily, String col, String val) throws IOException {
    init();
    Table table = connection.getTable(TableName.valueOf(tableName));
    Put put = new Put(rowKey.getBytes());
    put.addColumn(colFamily.getBytes(), col.getBytes(), val.getBytes());
    table.put(put);
    table.close();
    close();
}
```

- 删除数据。这里我设置的是删除指定的rowKey行colFamily列的数据，也是一格一格的删除

```
public static void deleteRow(String tableName, String rowKey, String colFamily, String col) throws IOException {
    init();
    Table table = connection.getTable(TableName.valueOf(tableName));
    Delete delete = new Delete(rowKey.getBytes());
    // delete all data from specified column family
    delete.addFamily(colFamily.getBytes());
    table.delete(delete);
    table.close();
    close();
}
```

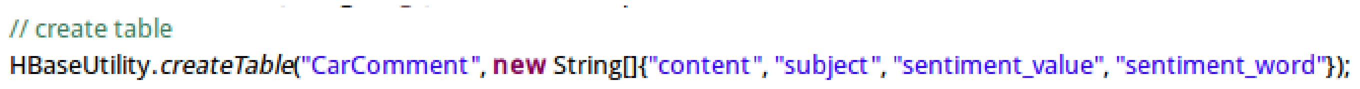
- 查找数据。这里同样是一格一格查找，showCell函数进行格式化输出



```java
public static void getData(String tableName, String rowKey, String colFamily, String col) throws IOException {
    init();
    Table table = connection.getTable(TableName.valueOf(tableName));
    Get get = new Get(rowKey.getBytes());
    get.addColumn(colFamily.getBytes(), col.getBytes());
    Result result = table.get(get);
    showCell(result);
    table.close();
    close();
}

/**
 * formatted output
 * @param result
 */
public static void showCell(Result result) {
    Cell[] cells = result.rawCells();
    for (Cell cell:cells) {
        System.out.println("RowName:" + new String(CellUtil.cloneRow(cell)) + " ");
        System.out.println("Timestamp:" + cell.getTimestamp() + " ");
        System.out.println("column Family:" + new String(CellUtil.cloneFamily(cell)) + " ");
        // System.out.println("row Name:" + new String(CellUtil.cloneQualifier(cell)) + " ");
        System.out.println("value:" + new String(CellUtil.cloneValue(cell)) + " ");
    }
}
```

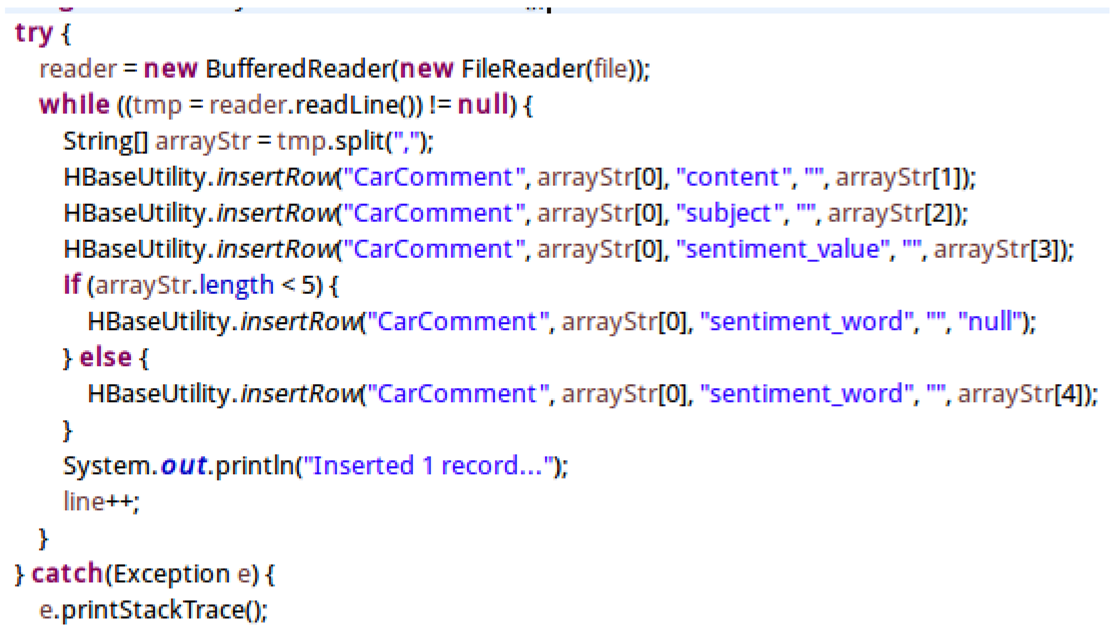
四、测试

首先创建CarComment表。这里原始数据的第一列id不作为colFamily



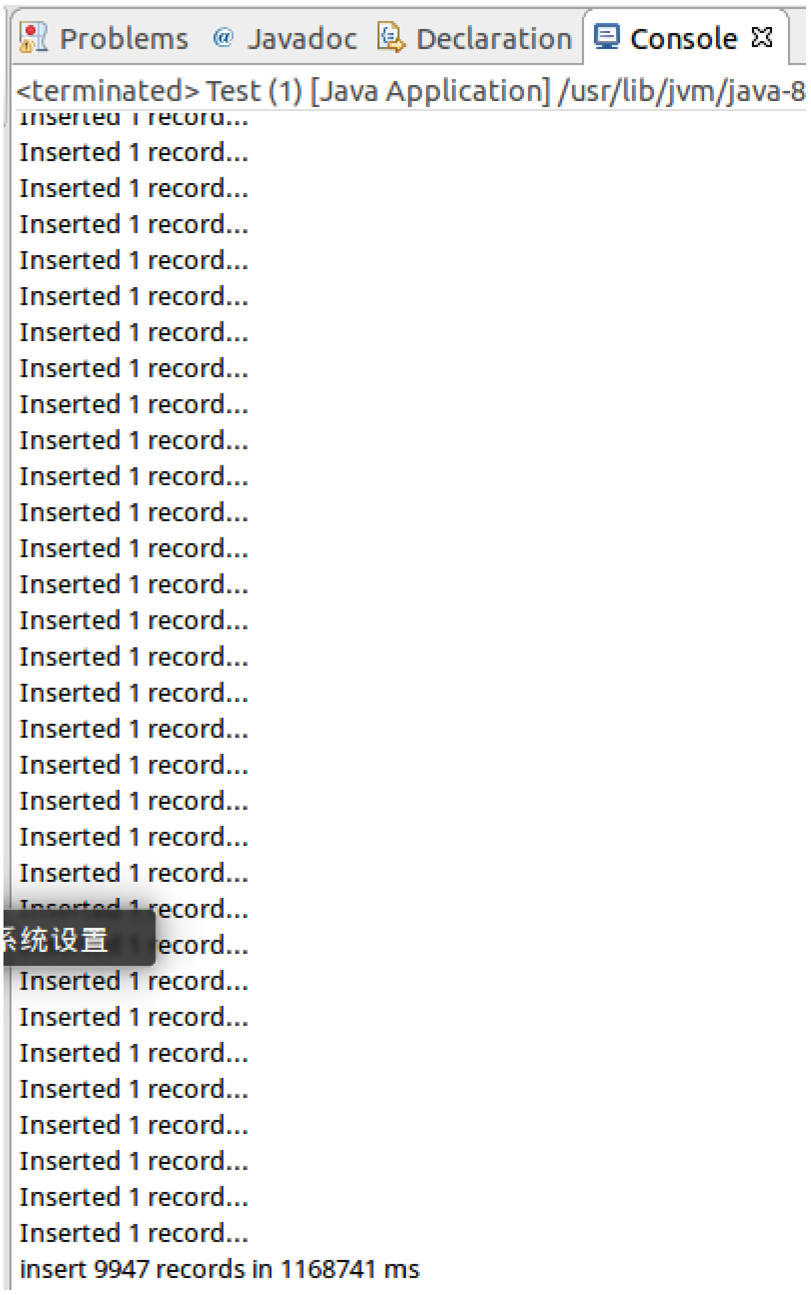
```java
// create table
HBaseUtility.createTable("CarComment", new String[]{"content", "subject", "sentiment_value", "sentiment_word"});
```

从本地txt文件读取数据并进行插入。将原始数据的第一列id作为rowKey



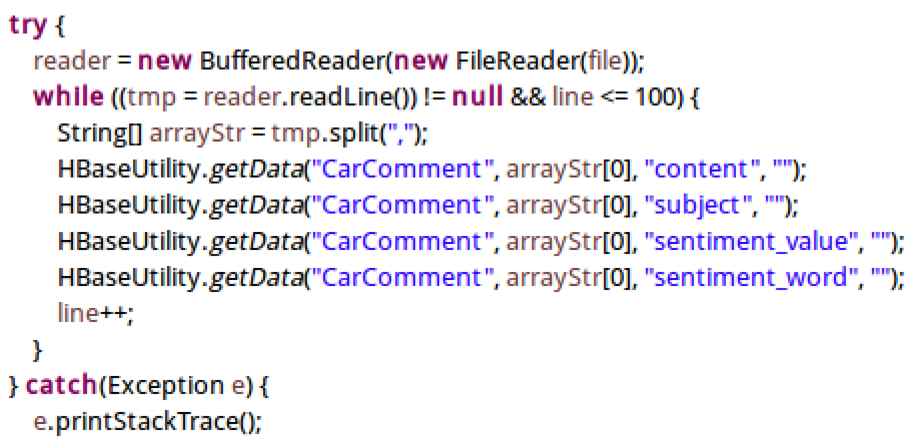
```java
try {
    reader = new BufferedReader(new FileReader(file));
    while ((tmp = reader.readLine()) != null) {
        String[] arrayStr = tmp.split(",");
        HBaseUtility.insertRow("CarComment", arrayStr[0], "content", "", arrayStr[1]);
        HBaseUtility.insertRow("CarComment", arrayStr[0], "subject", "", arrayStr[2]);
        HBaseUtility.insertRow("CarComment", arrayStr[0], "sentiment_value", "", arrayStr[3]);
        if (arrayStr.length < 5) {
            HBaseUtility.insertRow("CarComment", arrayStr[0], "sentiment_word", "", "null");
        } else {
            HBaseUtility.insertRow("CarComment", arrayStr[0], "sentiment_word", "", arrayStr[4]);
        }
        System.out.println("Inserted 1 record...");
        line++;
    }
} catch(Exception e) {
    e.printStackTrace();
```

插入完成


Problems @ Javadoc Declaration Console

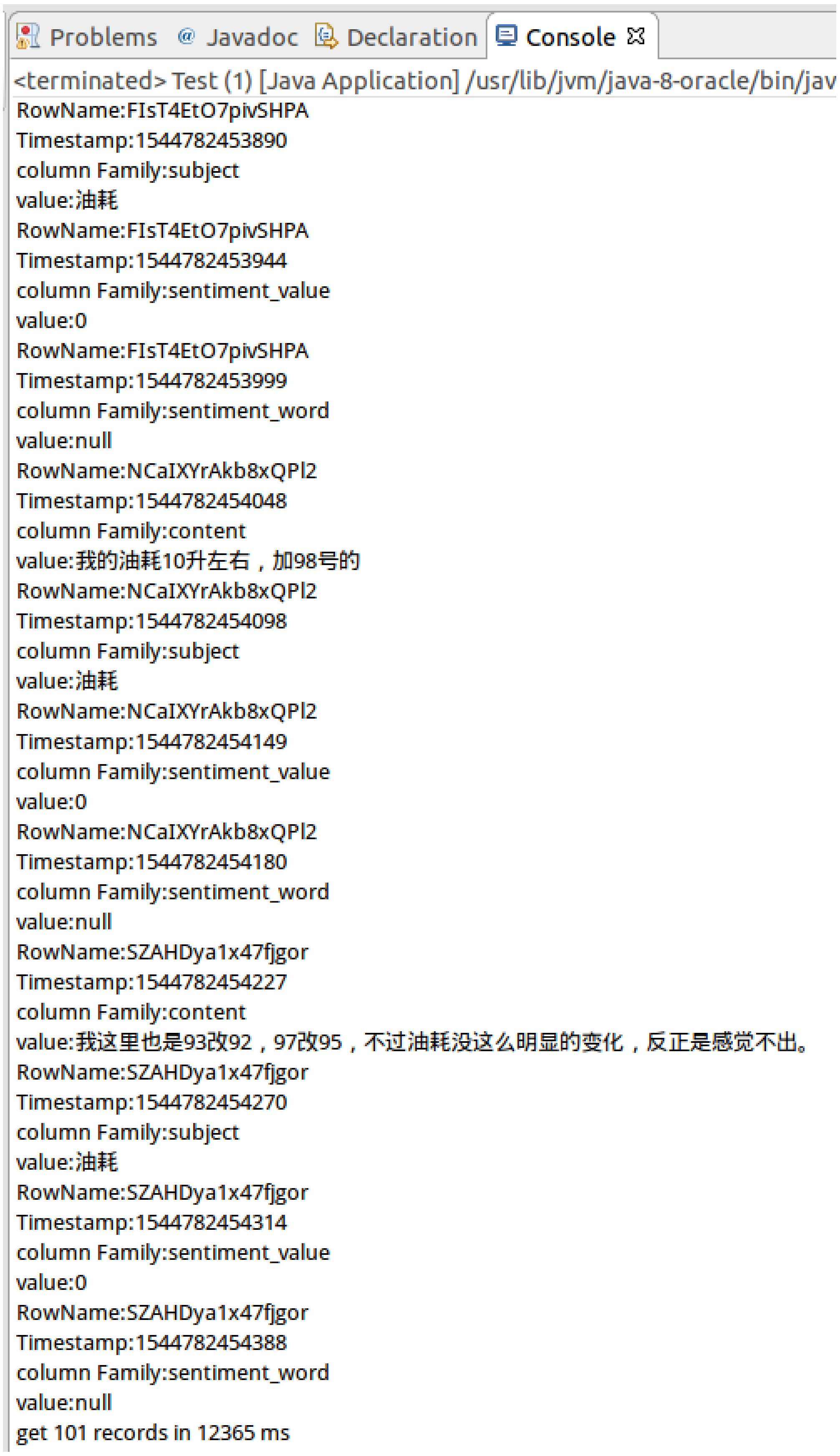
```
<terminated> Test (1) [Java Application] /usr/lib/jvm/java-8
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
Inserted 1 record...
insert 9947 records in 1168741 ms
```

这里取前101条数据进行展示



```java
try {
    reader = new BufferedReader(new FileReader(file));
    while ((tmp = reader.readLine()) != null && line <= 100) {
        String[] arrayStr = tmp.split(",");
        HBaseUtility.getData("CarComment", arrayStr[0], "content", "");
        HBaseUtility.getData("CarComment", arrayStr[0], "subject", "");
        HBaseUtility.getData("CarComment", arrayStr[0], "sentiment_value", "");
        HBaseUtility.getData("CarComment", arrayStr[0], "sentiment_word", "");
        line++;
    }
} catch(Exception e) {
    e.printStackTrace();
```

展示结果


Problems Javadoc Declaration Console

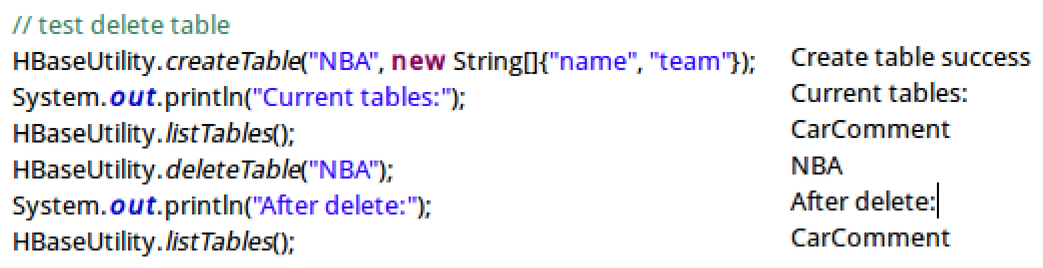
```
<terminated> Test (1) [Java Application] /usr/lib/jvm/java-8-oracle/bin/jav
RowName:FIsT4EtO7pivSHPA
Timestamp:1544782453890
column Family:subject
value:油耗
RowName:FIsT4EtO7pivSHPA
Timestamp:1544782453944
column Family:sentiment_value
value:0
RowName:FIsT4EtO7pivSHPA
Timestamp:1544782453999
column Family:sentiment_word
value:null
RowName:NCaIXYrAkb8xQPl2
Timestamp:1544782454048
column Family:content
value:我的油耗10升左右，加98号的
RowName:NCaIXYrAkb8xQPl2
Timestamp:1544782454098
column Family:subject
value:油耗
RowName:NCaIXYrAkb8xQPl2
Timestamp:1544782454149
column Family:sentiment_value
value:0
RowName:NCaIXYrAkb8xQPl2
Timestamp:1544782454180
column Family:sentiment_word
value:null
RowName:SZAHDya1x47fjgor
Timestamp:1544782454227
column Family:content
value:我这里也是93改92，97改95，不过油耗没这么明显的变化，反正是感觉不出。
RowName:SZAHDya1x47fjgor
Timestamp:1544782454270
column Family:subject
value:油耗
RowName:SZAHDya1x47fjgor
Timestamp:1544782454314
column Family:sentiment_value
value:0
RowName:SZAHDya1x47fjgor
Timestamp:1544782454388
column Family:sentiment_word
value:null
get 101 records in 12365 ms
```

删除第一行content列的数据

```java
// delete a data by row key
HBaseUtility.deleteRow("CarComment", "vUXizsqexyZVRdFH", "content", "");
// search it for test
HBaseUtility.getData("CarComment", "vUXizsqexyZVRdFH", "content", "");
```

下面的查询语句没有返回结果，说明删除成功

创建一个新表，并进行表的删除测试

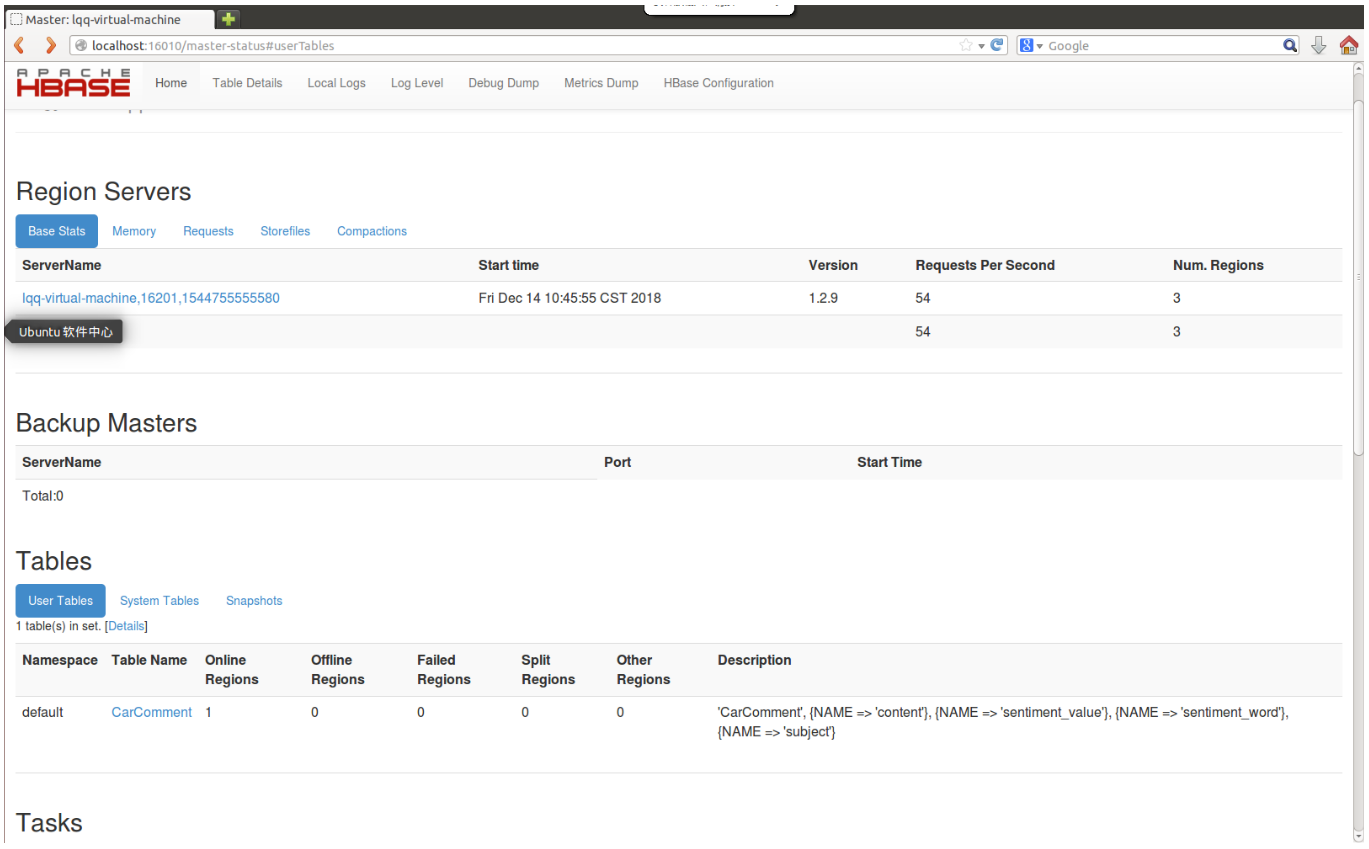


```
// test delete table
HBaseUtility.createTable("NBA", new String[]{"name", "team"});
System.out.println("Current tables:");
HBaseUtility.listTables();
HBaseUtility.deleteTable("NBA");
System.out.println("After delete:");
HBaseUtility.listTables();
```

```
Create table success
Current tables:
CarComment
NBA
After delete:
CarComment
```

可以看到，删除操作成功

可以通过16010端口查看HBase运行情况



Master: lqq-virtual-machine

localhost:16010/master-status#userTables

Region Servers

ServerName	Start time	Version	Requests Per Second	Num. Regions
lqq-virtual-machine,16201,1544755555580	Fri Dec 14 10:45:55 CST 2018	1.2.9	54	3
			54	3

Backup Masters

ServerName	Port	Start Time
Total:0		

Tables

1 table(s) in set. [Details]

Namespace	Table Name	Online Regions	Offline Regions	Failed Regions	Split Regions	Other Regions	Description
default	CarComment	1	0	0	0	0	'CarComment', {NAME => 'content'}, {NAME => 'sentiment_value'}, {NAME => 'sentiment_word'}, {NAME => 'subject'}

Tasks

总结

HBase是一种Hadoop数据库，基于行键（rowKey），列键（cloFamily）与时间戳建立索引，是一个可以随机访问的存储和检索数据的平台。HBase在一个服务器集群上运行，也可以相应地横向扩展。本次实验由于硬件条件（PC，虚拟机）的限制，进行了伪分布式的配置，并且对数据的基本操作进行了实现。之后出于好奇心，利用三台虚拟机部署Hadoop实现MapReduce，详见MapReduce.pdf

参考文献

1. 如何在Ubuntu 14.04中安装Java
   https://jingyan.baidu.com/article/1e5468f9642e4a484961b7c3.html
2. Hadoop安装教程
   http://dblab.xmu.edu.cn/blog/install-hadoop/

3. HBase的安装与运行
   http://dblab.xmu.edu.cn/blog/using_hbase/#more-149
4. HBase Java API详解
   https://www.cnblogs.com/tiantianbyconan/p/3557571.html