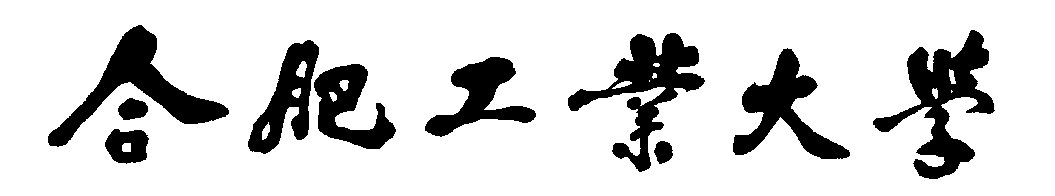
****

专业课程

（计算机与信息学院）

大数据处理技术实验报告

|  |
| --- |
| 计算机科学与技术16-3班 |
| 任恒2016212063 |
| 001班 |
| 吴共庆 |
| 吴共庆 |
| 科教楼D502 |

|  |
| --- |
| 专业班级 |
| 学生姓名及学号 |
| 课程教学班号 |
| 任课教师 |
| 实验指导教师 |
| 实验地点 |

2018～2019学年第 二 学期

目录

[综合设计-大数据处理技术2018-2019第二学期 **错误!未定义书签。**](#_Toc8335007)

[**1. Hadoop分布式环境搭建与Hive安装** **错误!未定义书签。**](#_Toc8335008)

1) [安装Hadoop **错误!未定义书签。**](#_Toc8335009)

2) [安装Hive **错误!未定义书签。**](#_Toc8335010)

[**2.** **Hive数据清洗 (30分)** **错误!未定义书签。**](#_Toc8335011)

1） [创建raw\_test数据库 **错误!未定义书签。**](#_Toc8335012)

2) [创建shop1，shop2表 **错误!未定义书签。**](#_Toc8335013)

3) [创建shop1\_1表 **错误!未定义书签。**](#_Toc8335014)

4) [创建shop1\_2表 **错误!未定义书签。**](#_Toc8335015)

5) [创建shop2\_1表 **错误!未定义书签。**](#_Toc8335016)

[**3. 数据分析(20分)** **错误!未定义书签。**](#_Toc8335017)

1. [统计评论量最多的5部电影 **错误!未定义书签。**](#_Toc8335018)
2. [统计2013年以后电影最多的5个地区 **错误!未定义书签。**](#_Toc8335019)
3. [电影总体热度系数 **错误!未定义书签。**](#_Toc8335020)

[**4. 数据可视化** **错误!未定义书签。**](#_Toc8335021)

1) [折线图 **错误!未定义书签。**](#_Toc8335022)

2) [雷达图 **错误!未定义书签。**](#_Toc8335023)

[**5． 总结** 24](#_Toc8342811)

综合设计-大数据处理技术2018-2019第二学期

根据麦肯锡全球研究所的定义，大数据是一种规模大到在获取、存储、管理、分析方面大大超出了传统数据库软件工具能力范围的数据集合，具有海量的数据规模、快速的数据流转、多样的数据类型和价值密度低四大特征。而当前人们所说的人工智能，是指研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术以及应用系统的一门新的技术科学，是由人工制造出来的系统所表现出来的智能。但由于传统人工智能受制于计算能力，并没能完成大规模的并行计算和并行处理，人工智能系统的能力较差。

随着移动互联网的爆发，数据量呈现出指数级的增长，大数据的积累为人工智能提供了基础支撑。同时受益于计算机技术在数据采集、存储、计算等环节的突破，人工智能已从简单的算法+数据库发展演化到了机器学习+深度理解的状态。大数据+人工智能帮助了各行各业的企业从原本毫无价值的海量数据中挖掘出用户的需求，使数据能够从量变到质变，真正产生价值。

本综合设计将分为大数据平台搭建、数据预处理、数据分析、数据可视化以及综合题等五部分内容，请按照要求完成。

1. **Hadoop分布式环境搭建与Hive安装. （30分）**

请先按照如下规划表对三台虚拟机进行配置

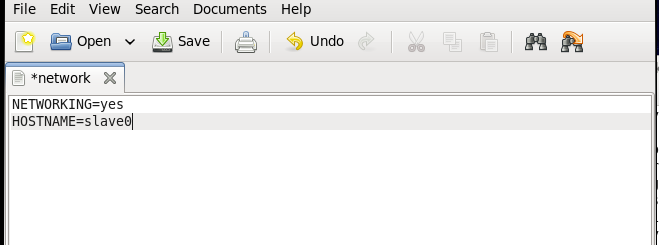
|  |  |  |
| --- | --- | --- |
| 主机名 | IP | 备注 |
| master | 192.168.10.2 | 主节点 |
| slave1 | 192.168.10.3 | 从节点 |
| slave2 | 192.168.10.4 | 从节点 |

安装包均放在/opt/hadoop-package/文件夹下，要求如下：

1. .安装Hadoop (20分)
2. 配置hdfs-site.xml，内容为文件的副本数1。并将配置页面截图保存到实验报告任务1-1-1中。（4分）

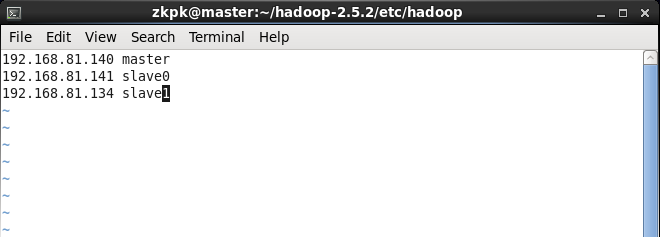
更改主机名：将hostname改为master、salve0、slave1.

|  |
| --- |
| >>gedit /etc/sysconfig/network |

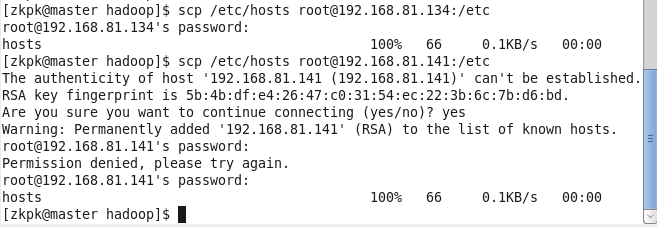


在主节点中配置hosts文件：

|  |
| --- |
| >>sudo vim /etc/hosts |



将主节点上hosts文件复制到两个slave节点上：



三个节点刷新hostname：（如master）

|  |
| --- |
| >>hostname master |

配置密钥：

每个节点下分别执行：

|  |
| --- |
| >>ssh-keygen rsa |



将从节点密钥拷贝到主节点（以slave1为例）：

|  |
| --- |
| >>scp ~/.ssh/id\_rsa.pub root@master:~/.ssh/id\_rsa\_slave1.pub |

主节点上用scp 命令将 authorized\_keys 文件拷贝到子节点上的 “/root/.ssh”目录下

|  |
| --- |
| >>scp authorized\_keys [root@slave0:/root/.ssh/](mailto:root@slave0:/root/.ssh/)  >>scp authorized\_keys [root@slave1:/root/.ssh/](mailto:root@slave1:/root/.ssh/) |

hdfs-site.xml文件配置：

|  |
| --- |
| gedit ~/Hadoop-2.5.2/etc/Hadoop/hdfs-site.xml |

Slave文件配置：

|  |
| --- |
| >>gedit ~/hadoop-2.5.2/etc/hadoop/slave  输入：  slave0  slave1 |

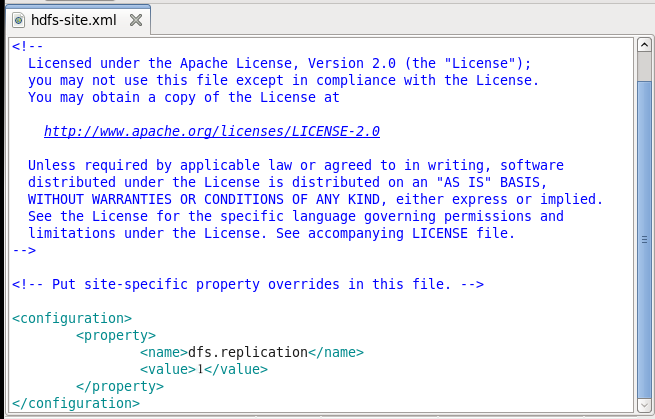
拷贝主节点上的”/etc/profile”文件到子节点:

|  |
| --- |
| >>scp /etc/profile root@slave0:/etc/  >>scp /etc/profile root@slave1:/etc/ |

并分别在子节点上使 “/etc/profile”文件生效：

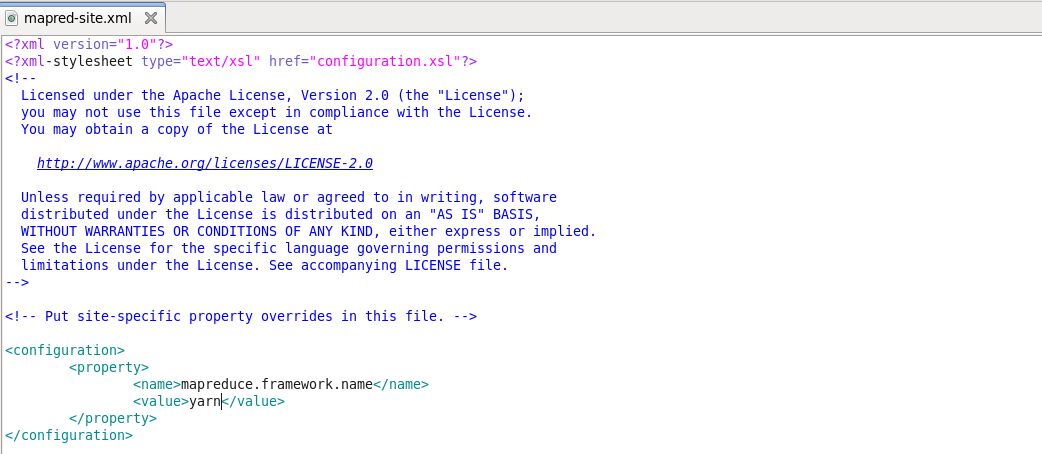
|  |
| --- |
| >>source /etc/profile |

**其他文件配置参考教程（此处主要是将两个节点扩展到三个节点）**

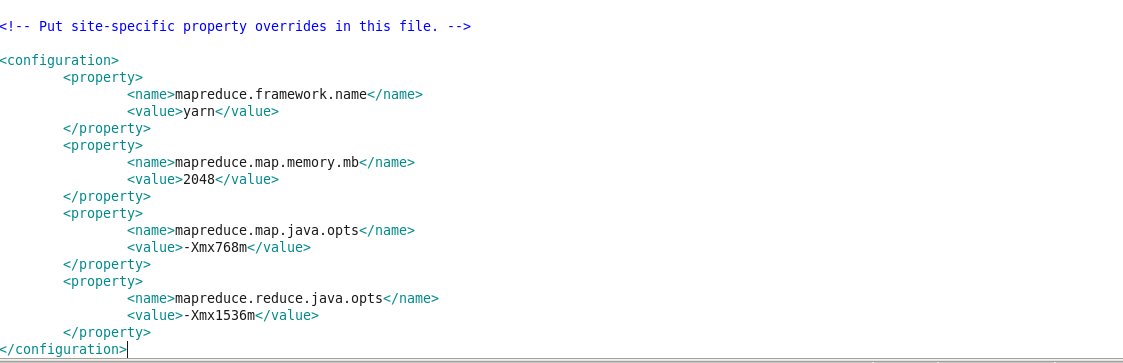


1. 配置mapred-site.xml，指定mapreduce的环境为yarn。并将配置页面截图保存到实验报告任务1-1-2中。（4分）

|  |
| --- |
| >>gedit ~/Hadoop-2.5.2/etc/hadoop/mapred-site.xml |



1. 设置mapreduce.map.java.opts的值为-Xmx768m，设置mapreduce.reduce.memory.mb的值为2048MB，设置mapreduce.reduce.java.opts的值为 –Xmx1536m。并将配置页面截图并保存到实验报告任务1-1-3中。（6分）

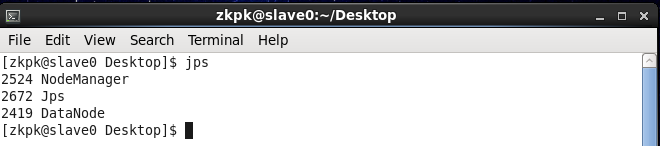


1. 启动集群。在三个节点上分别执行jps命令查看Hadoop进程并将查看到的进程结果截图保存到实验报告任务1-1-4中。(6分)

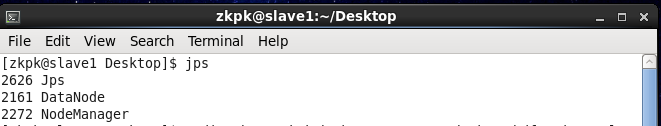
主节点查看启动的情况：



Slave0查看启动的情况：



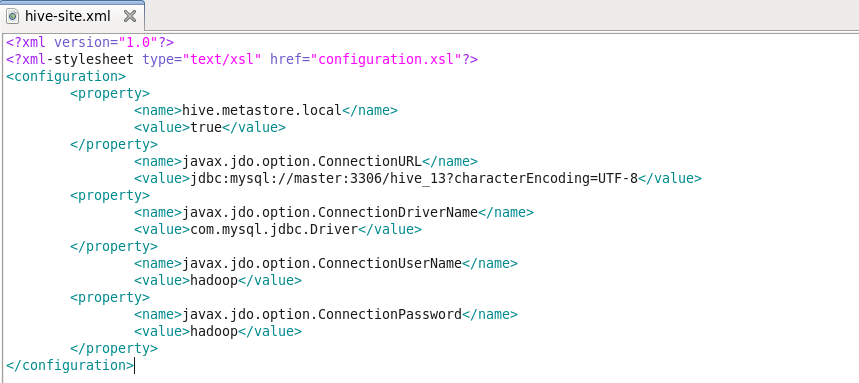
Slave1查看启动的情况：

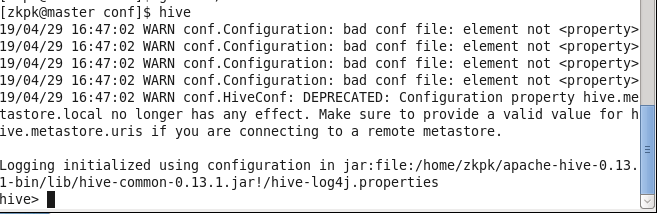


1. .安装Hive（mysql已经安装好）（10分）
2. 拷贝hive-env.sh.template并命名为hive-env.sh，配置hive-env.sh，内容为Hadoop安装目录。并将配置页面截图保存到实验报告任务1-2-1中。（2分）配置hive-env.sh:



2) 新建hive.site.xml（可参考如下截图中的hive-site.xml模板，注意：星号内容需自行补充）添加mysql连接信息。并将配置页面截图保存到实验报告任务1-2-2中。（2分）



3) 启动hive，并将查看到的结果截图保存到实验报告任务1-2-3中。（6分）

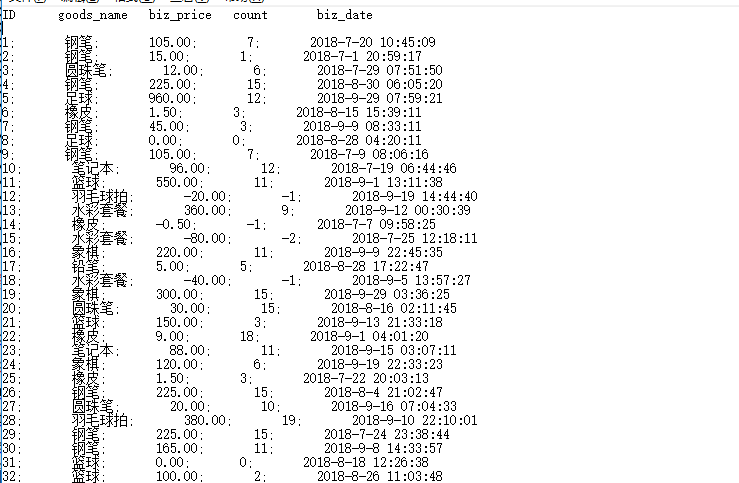
**2. Hive数据清洗 (30分)**

（在Hive交互环境下完成操作）

“安徽省大数据学院”跟学校进行校企合作，提供了一些商品交易的数据源给学校学生进行数据清洗，想请你帮他们完成数据的清洗。

“数据源”目录下有shop\_1.txt和shop\_2.txt数据源，里面是一些商品的交易记录，数据格式如下:

其中，数据源属性从左往右以此是id(商品记录id)，goods\_name(商品名称)， biz\_price(商品交易金额)，count(商品交易数量)，biz\_date(商品交易日期)



**要求如下：**

1) 创建raw\_test数据库，进入raw\_test数据库。并将该截图保存到实验报告任务2-1-1中。(5分)

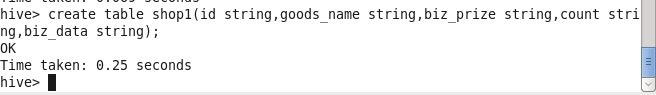
创建raw\_test数据库：



2) 基于raw\_test数据库，创建shop1，shop2表，将数据源shop\_1.txt和shop\_2.txt分别导入对应的表中。分别查询两个表中的前十条数据并将这两个截图保存到实验报告任务2-1-2中。(10分)

创建shop1表：

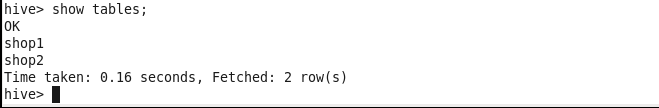
|  |
| --- |
| <<create table shop1(id string,goods\_name string,biz\_prize string,count string,biz\_data string) ; |



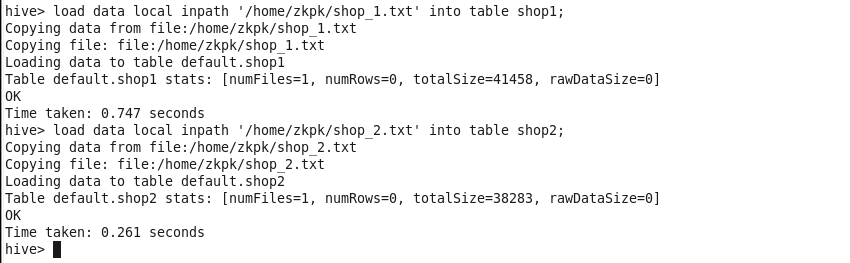
创建shop2表：

|  |
| --- |
| <<create table shop2(id string,goods\_name string,biz\_prize string,count string,biz\_data string) ; |

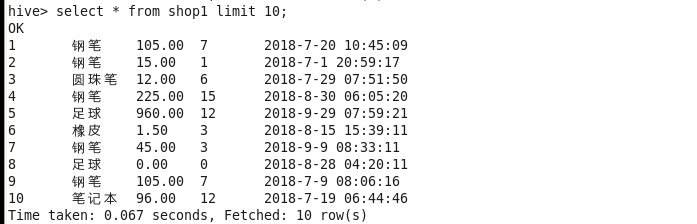




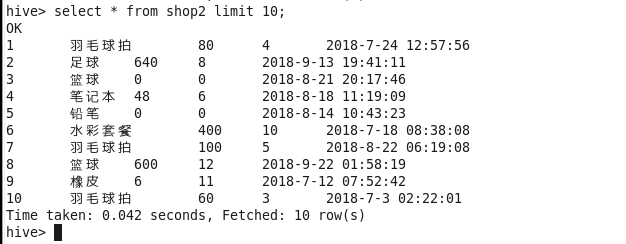
上传数据到shop1和shop2表中：



Shop1中前十条数据：

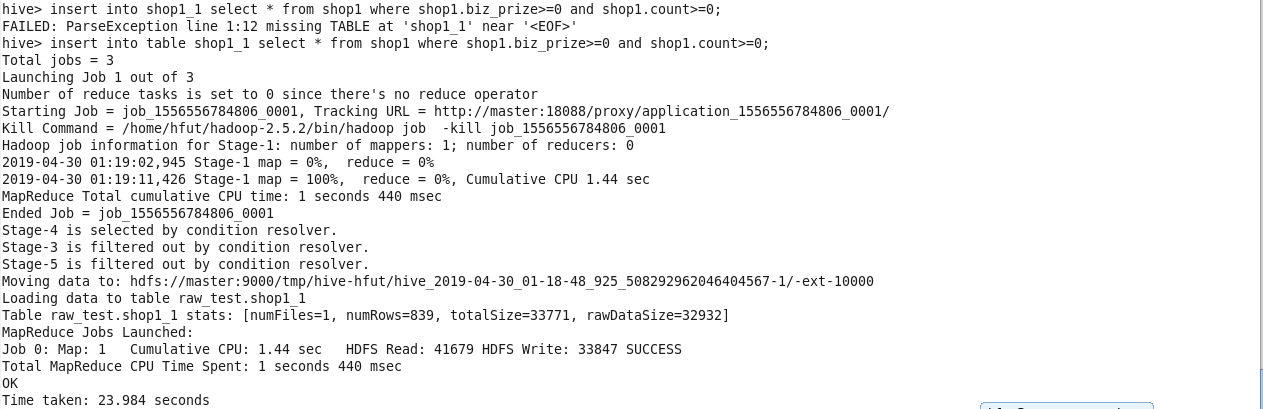


Shop2中前十条数据：



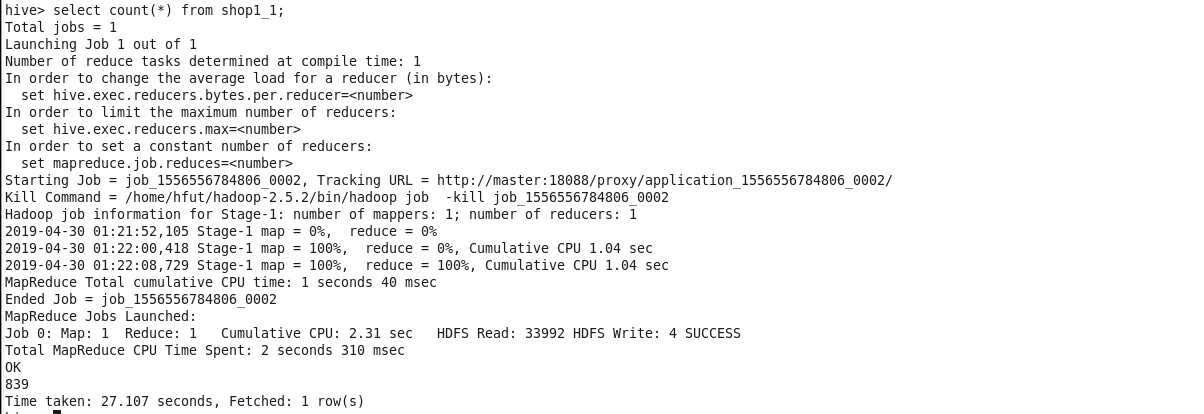
3) 创建shop1\_1表，过滤掉shop1表中交易金额为负数或者交易数量为负数的记录，并将结果保存到shop1\_1表中。查询shop1\_1表中的记录总数并将结果截图保存到实验报告任务2-1-3中。（5分）

|  |
| --- |
| <<create table shop1\_1(id string,goods\_name string,biz\_prize string,count string,biz\_data string) row format delimited fields terminated by '\t' stored as textfile; |



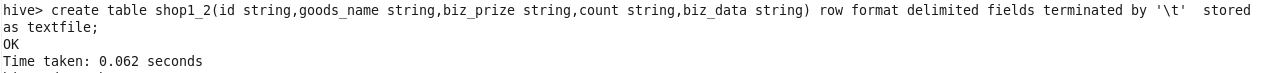
查询shop1\_1表中的记录总数

|  |
| --- |
| <<select count(\*) from shop1\_1; |

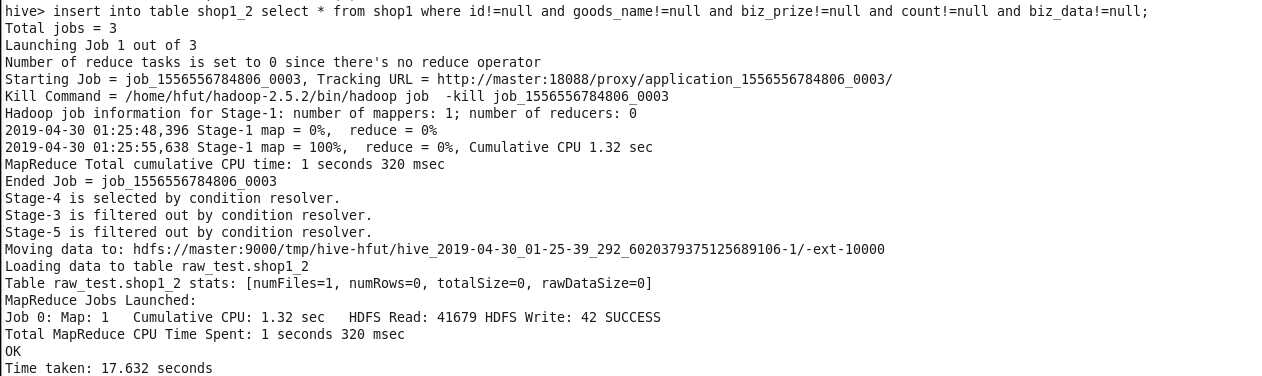


4) 创建shop1\_2表，过滤掉shop1表中存在字段为空的记录，并将结果保存到shop1\_2表中。查询shop1\_2表中的记录总数并将结果截图保存到实验报告任务2-1-4中。（5分）

|  |
| --- |
| <<create table shop1\_2(id string,goods\_name string,biz\_prize string,count string,biz\_data string) row format delimited fields terminated by '\t' stored as textfile; |

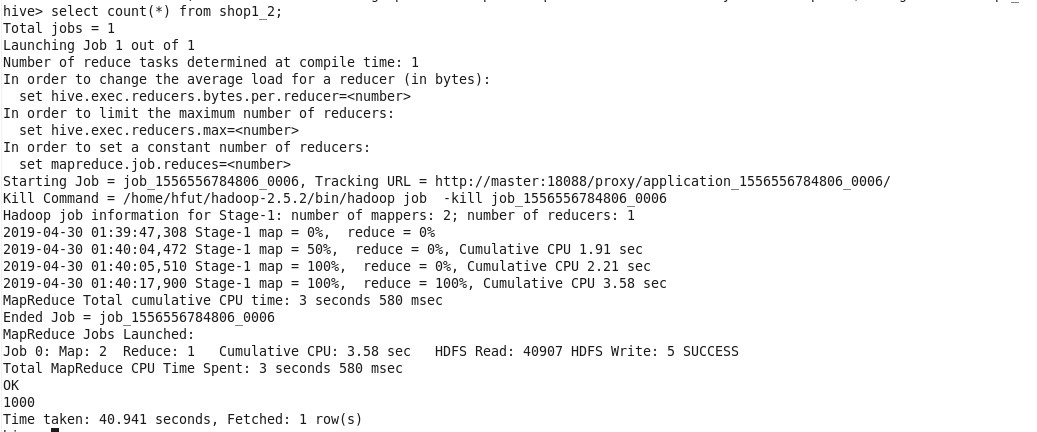


|  |
| --- |
| <<insert into table shop1\_2 select \* from shop1 where id!=null and goods\_name!=null and biz\_prize!=null and count!=null and biz\_data!=null; |



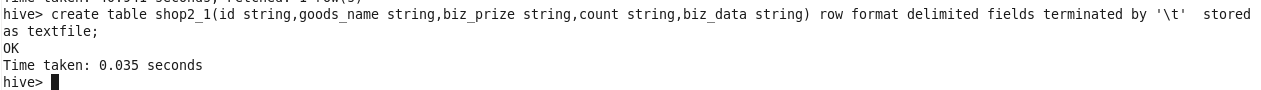
查询shop1\_2表中的记录总数

|  |
| --- |
| <<select count(\*) from shop1\_2; |

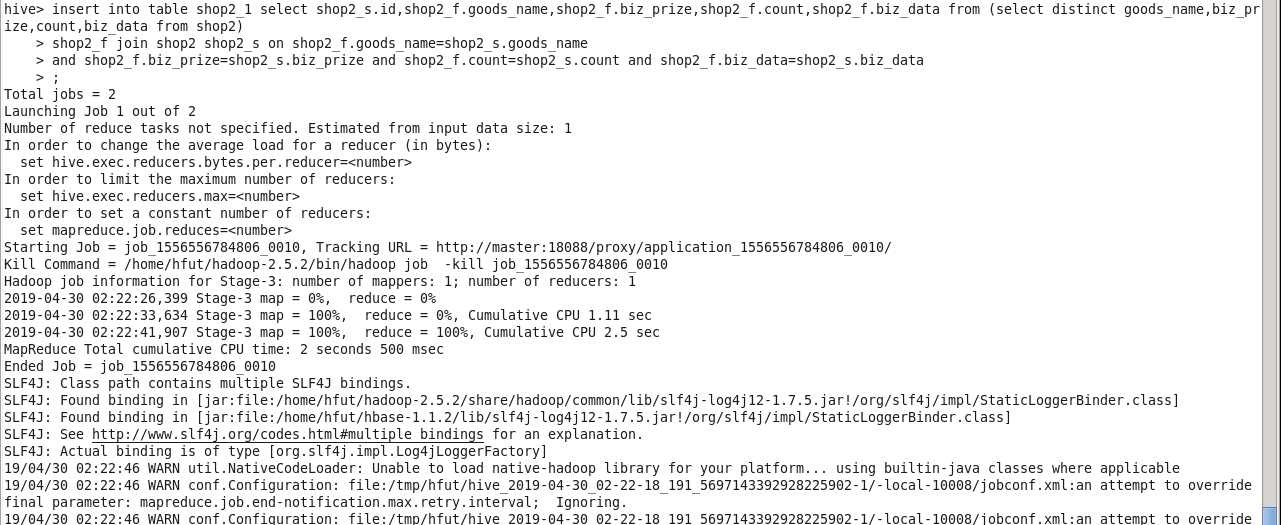


5) 创建shop2\_1表，过滤掉shop2表中重复的记录（除id字段外，其他字段都相同的记录被认为重复记录；如果存在2条及2条以上的重复记录，则只保留1条记录），并将结果保存到shop2\_1表中。查询shop2\_1表中的记录总数并将结果截图保存到实验报告任务2-1-5中。（5分）

|  |
| --- |
| <<create table shop2\_1(id string,goods\_name string,biz\_prize string,count string,biz\_data string) row format delimited fields terminated by '\t' stored as textfile; |

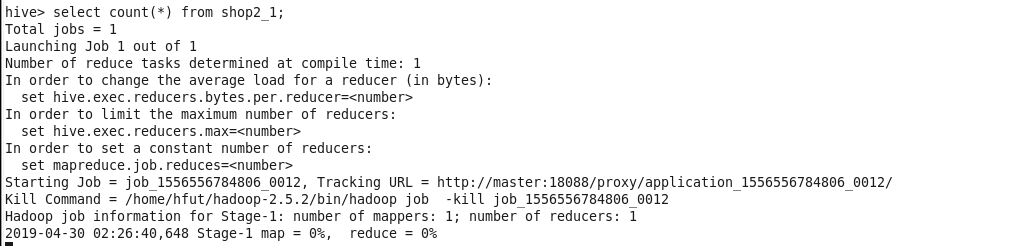


|  |
| --- |
| >> insert into table shop2\_1  >>select shop2\_s.id , shop2\_f.goods\_name , shop2\_f.biz\_prize,  >>shop2\_f.count, shop2\_f.biz\_data from  >>(select distinct goods\_name, biz\_prize,count, biz\_data from shop2)  >>shop2\_f join shop2 shop2\_s on shop2\_f.goods\_name=shop2\_s.goods\_name  >>and shop2\_f.biz\_prize=shop2\_s.biz\_prize and shop2\_f.count=shop2\_s.count and  >>shop2\_f.biz\_data=shop2\_s.biz\_data; |



查看shop2\_1总数：

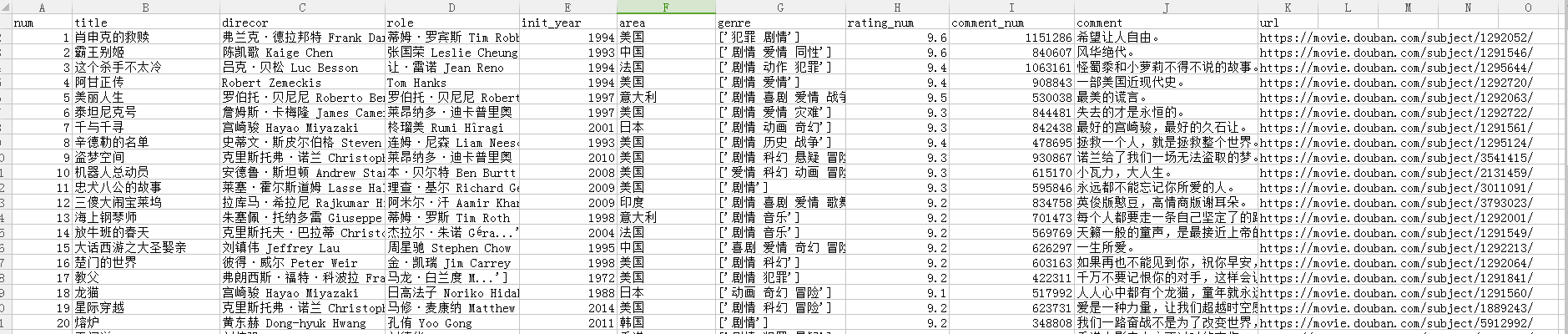
|  |
| --- |
| >>select count(\*) from shop2\_1; |



**3.数据分析(20分)**

“安徽省大数据学院”推出校园公众号以来受到广泛师生的关注，成为师生校园生活中必不可少的一部分；学校影视社为丰富师生课余生活开阔眼界和见识，为广大师生推送相关主题的优质影片信息。在请教了相关的老师以后，现在需要从影片的数据源进行影片的分析。在老师的推荐下影视社的成员找到你了，想请你帮他们完成影片数据的分析。

“数据源”目录下有top250\_f1.txt数据源，存放着部分影片信息。具体数据格式如下:



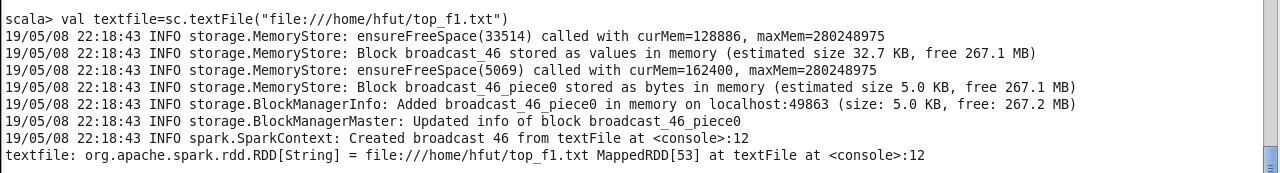
其中，数据源属性为:num(影片序号)，title(电影名)，direcor(导演)，role(主演)，init\_year(上映年份)，area(上映地区)，genre(电影类别)，rating\_num(评分)，comment\_num(评论数量) ，comment(评论)，url(链接)

**要求如下：**

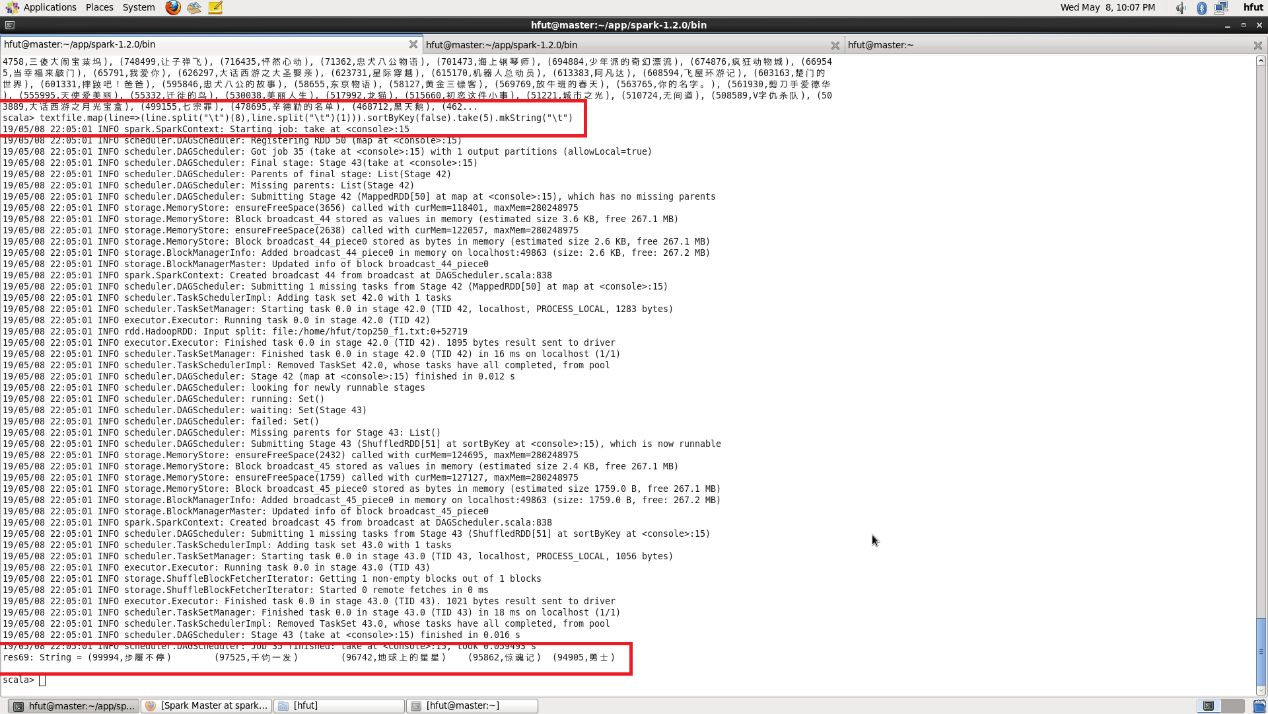
使用Spark平台完成以下题目，可使用spark-sql、spark-shell或者java/python调用spark库实现，截图时需保留spark-sql、spark-shell界面或者java/python代码以证实使用了spark。

1) 统计评论量最多的5部电影，并将结果截图保存到实验报告任务3-1-1中。(5分)

|  |
| --- |
| >>val textfile = sc.textFile([file:///home/hfut/top\_f1.txt](file:///F:\home\hfut\top_f1.txt))  //读入文件成为RDD |



|  |
| --- |
| >>textfile.map(line=>(line.split(“\t”)(8),line.split(“\t”)(1))).sortByKey(false).take(5).mkString(“\t”)  //生成（电影名，评论数）的key，value对按照value值倒序排序，取最大的5  //部输出 |

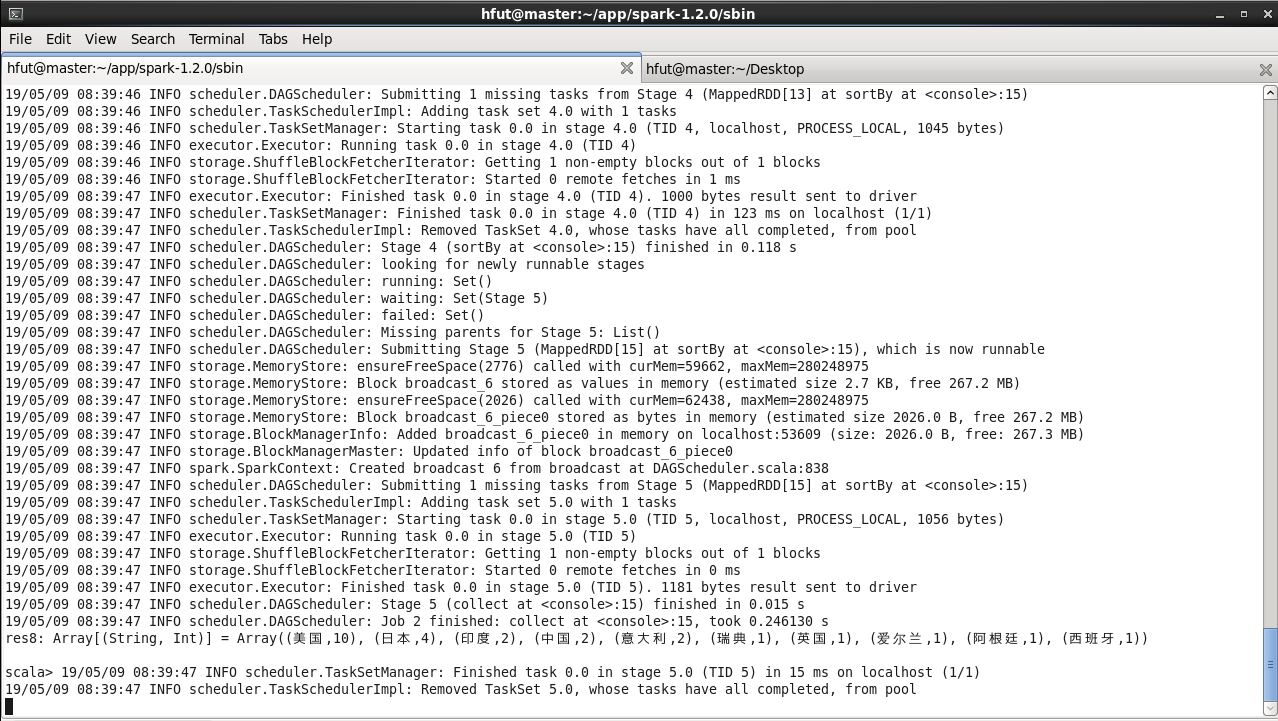
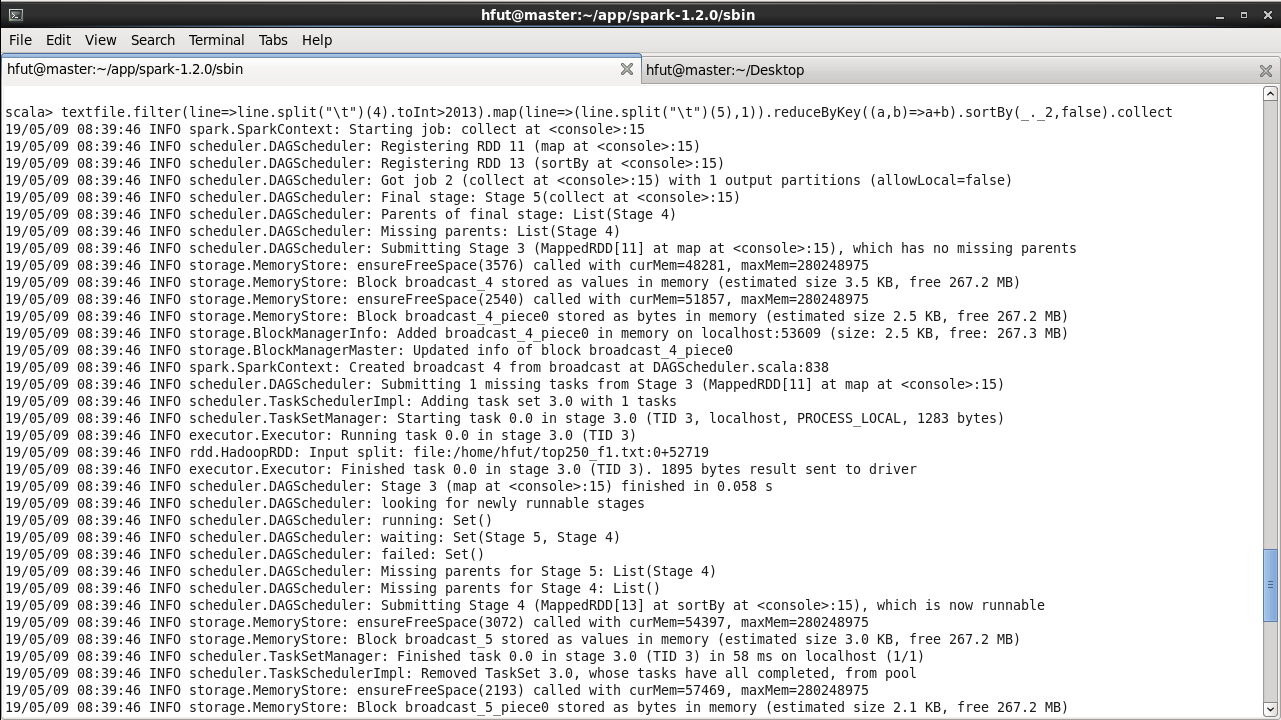
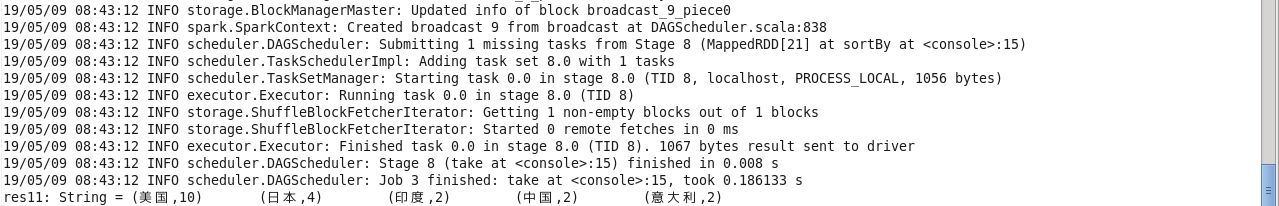


从上图可以看到结果为：

1. 《步履不停》 99994条
2. 《千钧一发》 97525条
3. 《地球上的星星》 96742条
4. 《惊魂记》 95862条
5. 《勇士》 94905条

2) 统计2013年以后电影最多的5个地区，并将结果截图保存到实验报告任务3-1-2中。(5分)

|  |
| --- |
| >> val textfile = sc.textFile([file:///home/hfut/top\_f1.txt](file:///F:\home\hfut\top_f1.txt))  //读入文件成为RDD  //file指示从本地读取，类似的有HDFS表示从HDFS读取文件  >>textfile.filter(line=>line.split("\t")(4).toInt>2013).map(line=>(line.split("\t")(5),1)).reduceByKey((a,b)=>a+b).sortBy(\_.\_2,false).take(5).mkString(“\t”)  //使用filter筛除2013年即以前的电影，使用map获得（地区，1）的key，//value对然后reduceByKey对相同key的key，value相加，按照value大小倒//序排列，取最大的五个输出 |

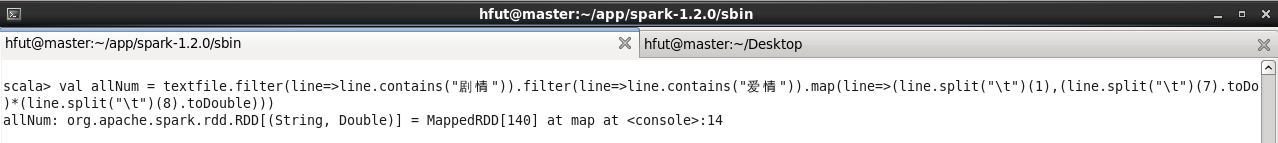
最终结果：

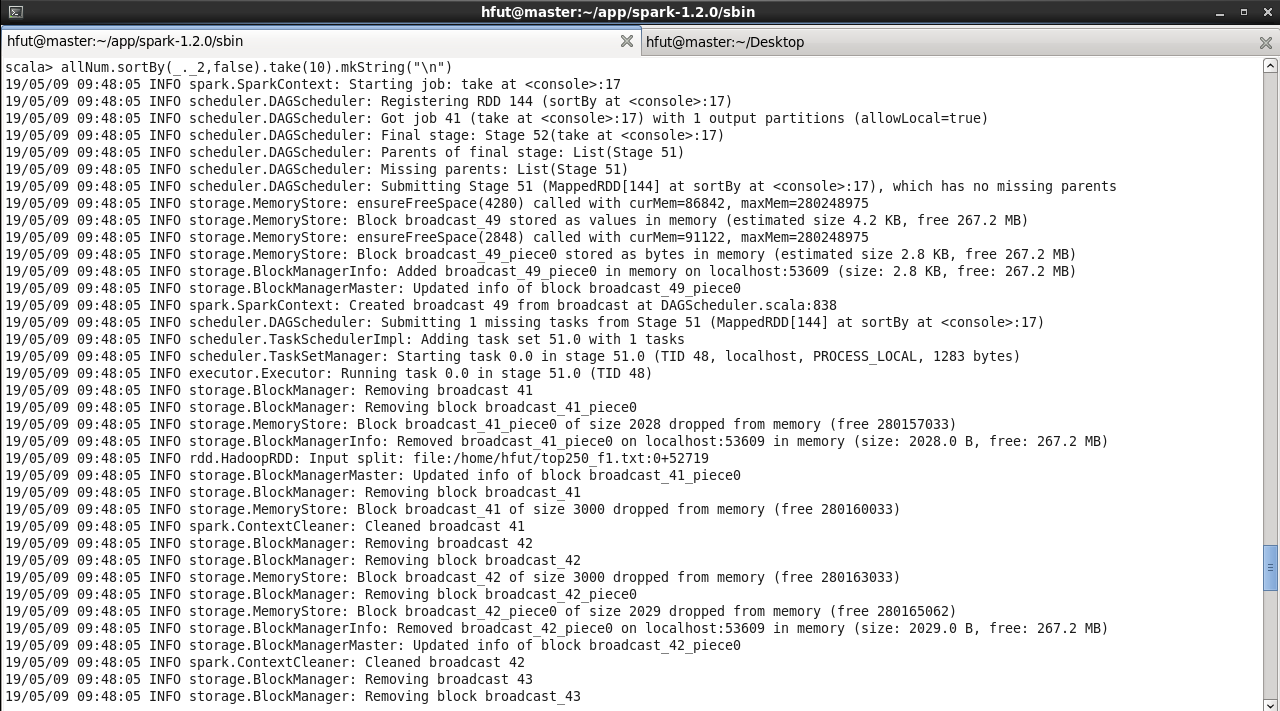
|  |
| --- |
| 1. 美国 10部 2. 日本 4部 3. 印度 2部 4. 中国 2部 5. 意大利 2部 |

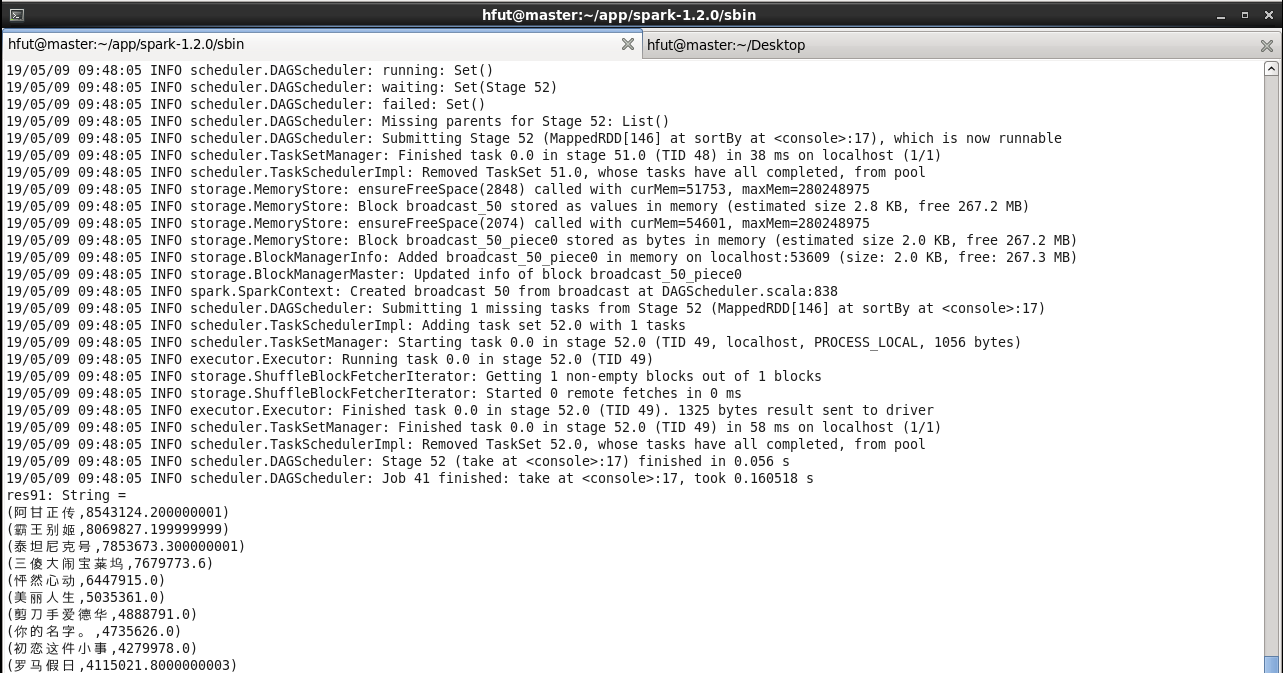
3) 电影排行榜一般是按分数来排列的，但如果相同分又怎样区分谁更好看，或者说某一部电影只有少部分人看但评分都是高分，那它不就一样可以登上排行榜前列。所以现在就创建一个总体热度系数（影片评论数量乘以评分），再用这个系数对影片作一个排序，然后找出电影类型中同时包含“剧情”和“爱情”的，系数最大的前十部，并将结果截图保存到实验报告任务3-1-3中。(10分)

使用的命令如下：

|  |
| --- |
| >> val textfile = sc.textFile([file:///home/hfut/top\_f1.txt](file:///F:\home\hfut\top_f1.txt))  //读入文件  >> val allNum = textfile.filter(line=>line.contains("剧情"))  .filter(line=>line.contains("爱情"))  .map(line=>(line.split("\t")(1),(line.split("\t")(7).toDouble)  \*(line.split("\t")(8).toDouble)))  //每一行一次为筛除不包含“爱情”，“剧情”的行，使用“ ”切分获得电影名、//以及评论数和评论进行相乘  >> allNum.sortBy(\_.\_2,false).take(10).mkString("\t")  //使用value值进行倒序排序取最大的10个 |







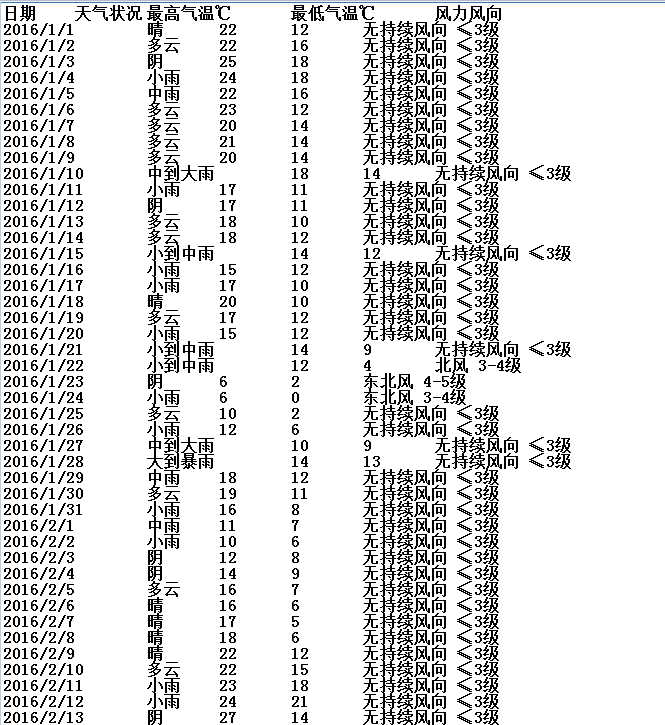
结果：

|  |  |
| --- | --- |
| 电影名称 | 总体热度系数 |
| 阿甘正传 | 8543124.200000001 |
| 霸王别姬 | 8069827.199999999 |
| 泰坦尼克号 | 7853673.300000001 |
| 三傻大闹宝莱坞 | 7679773.6 |
| 怦然心动 | 6447915.0 |
| 美丽人生 | 5035361.0 |
| 剪刀手爱德华 | 4888791.0 |
| 你的名字 | 4735626.0 |
| 初恋这件小事 | 4279978.0 |
| 罗马假日 | 4115021.8000000003 |

**4.数据可视化(20分)**

“安徽省大数据学院”跟学校进行校企合作，提供了一些气象的数据源给学校学生进行数据的可视化，想请你帮他们完成数据的可视化。

“数据源”目录下有weather.txt数据源，存放部分气象信息，如下图所示:

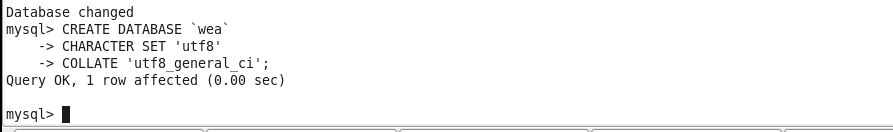


**要求如下：**

1. 通过折线图展示2018年9月份天每天最高温度变化图和最低温度变化图 。并将2018年9月份每天最高温度和最低温度查询结果和折线图展示结果截图保存到实验报告任务4-1-1中。(10分)

使用的sql语句如下：依次为创建数据库，指定编码；

|  |
| --- |
| Mysql>>CREATE DATABASE `wea`  ->CHARACTER SET 'utf8'  ->COLLATE 'utf8\_general\_ci'; |



指定当前使用的数据库：

|  |
| --- |
| mysql>>use wea; |

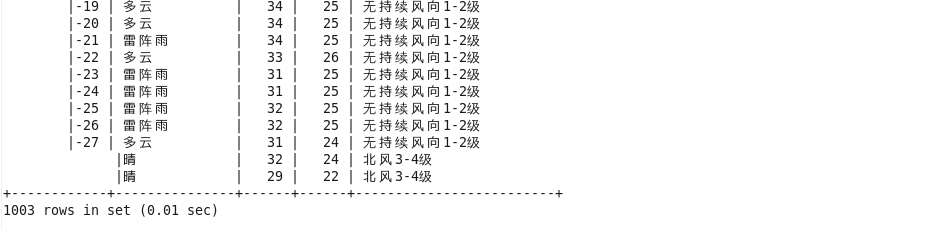
创建weather表：

|  |
| --- |
| CREATE TABLE `weather` (  `data` date DEFAULT NULL,  `cloud` varchar(11) DEFAULT NULL,  `high` int(3) DEFAULT NULL,  `low` int(3) DEFAULT NULL,  `wind` varchar(20) DEFAULT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=utf8; |

从本地加载数据到weather表中：

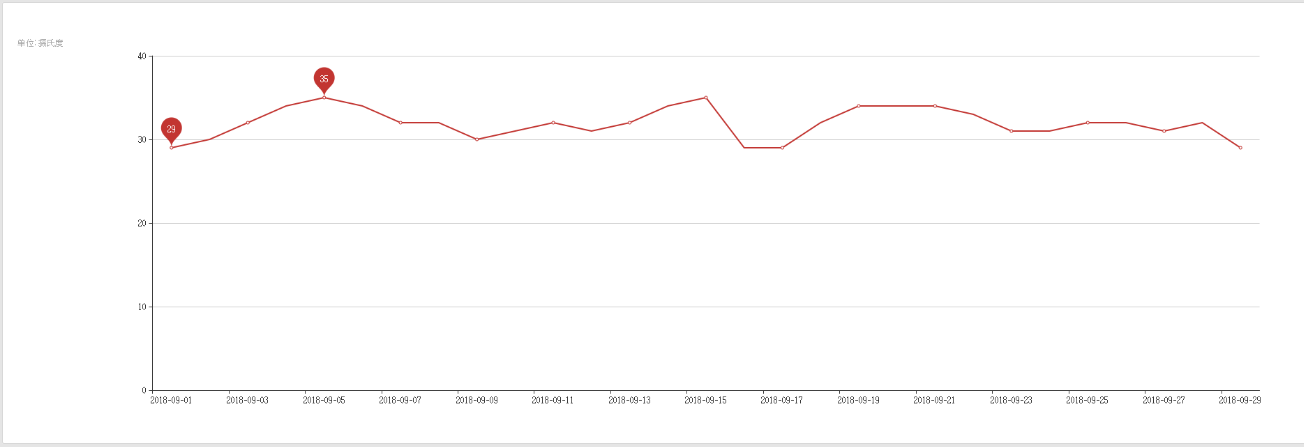
|  |
| --- |
| Mysql>>load data local infile "/home/hfut/weather.txt" into table weather fields terminated by '\t'; |

使用查询语句查询表的结构（篇幅有限仅展示部分）：

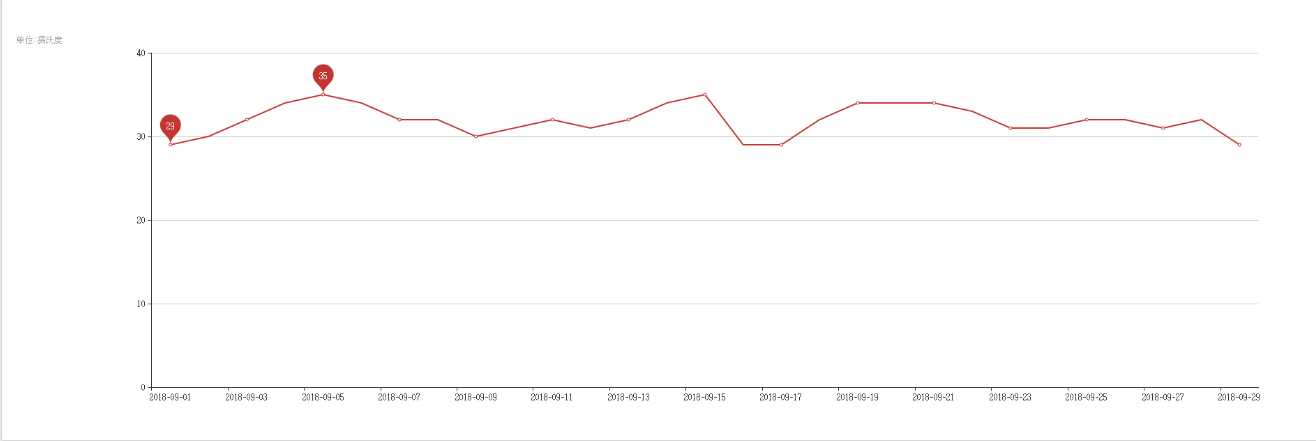


|  |
| --- |
| String sql = "select \* from weather where data between \'2018-9-1\' and \'2018-9-30\' order by data ascs"; |

最高气温折线图：



最低气温折线图：



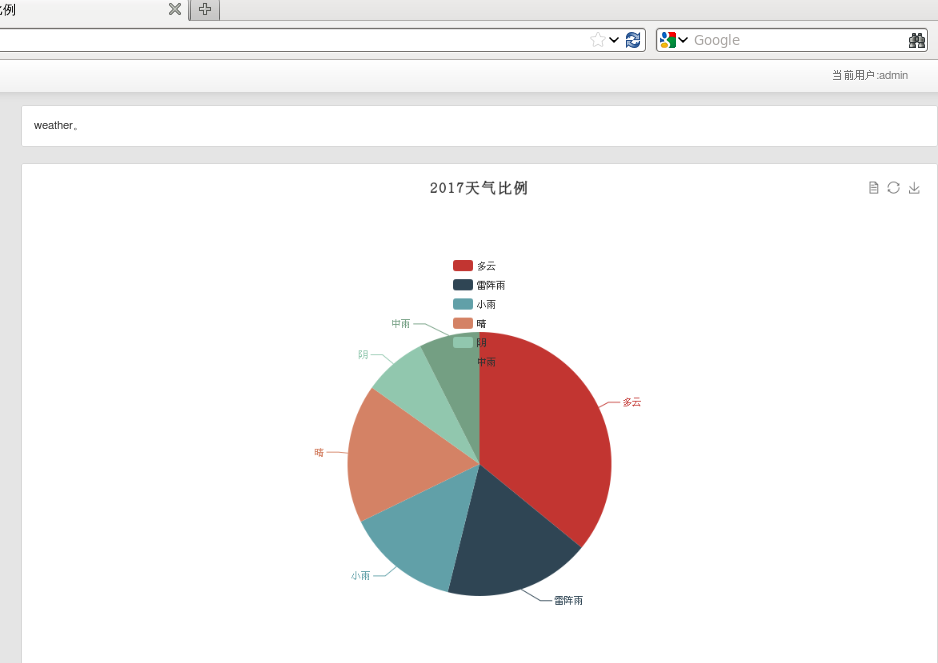
|  |
| --- |
| >> select data,high,low from weather where data between '2018-9-1' and '2018-9-30'; |

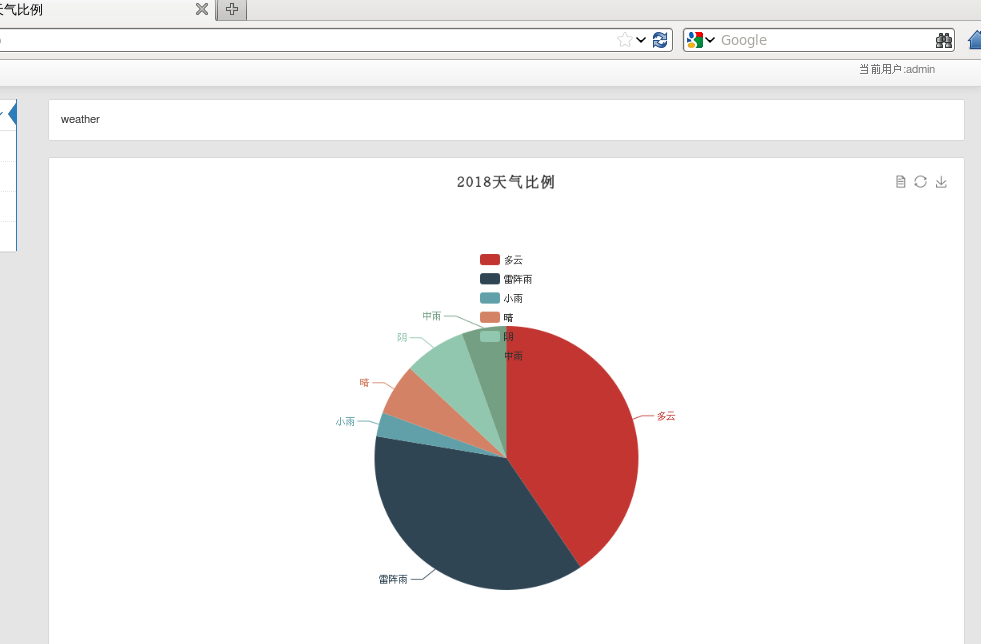


通过sql语句查询的结果可以看到折线图的结果与数据相符合。

2) 通过雷达图展示2017年和2018年天气概况信息(雷达图中所统计的6种天气概况分别为:多云，雷阵雨，小雨，晴，阴，中雨;这6种天气概况最大极限值为:200)。并将2017年和2018年天气概况信息查询结果和雷达图展示结果截图保存到实验报告任务4-1-2中。(10分)

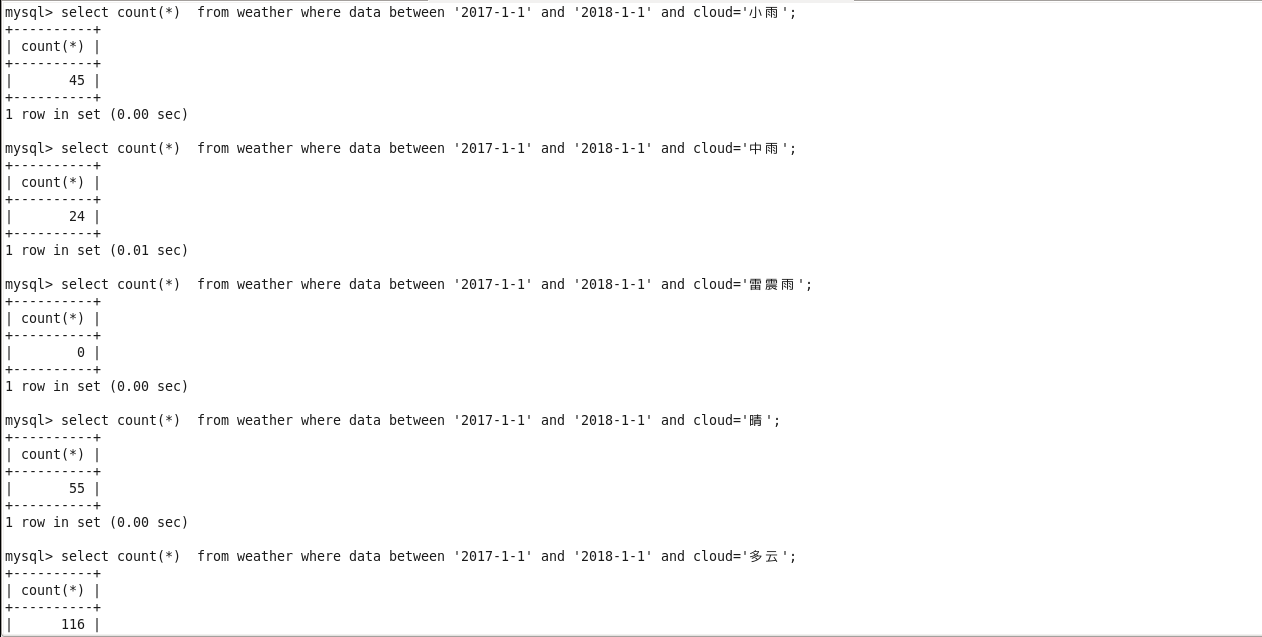
实验结果：

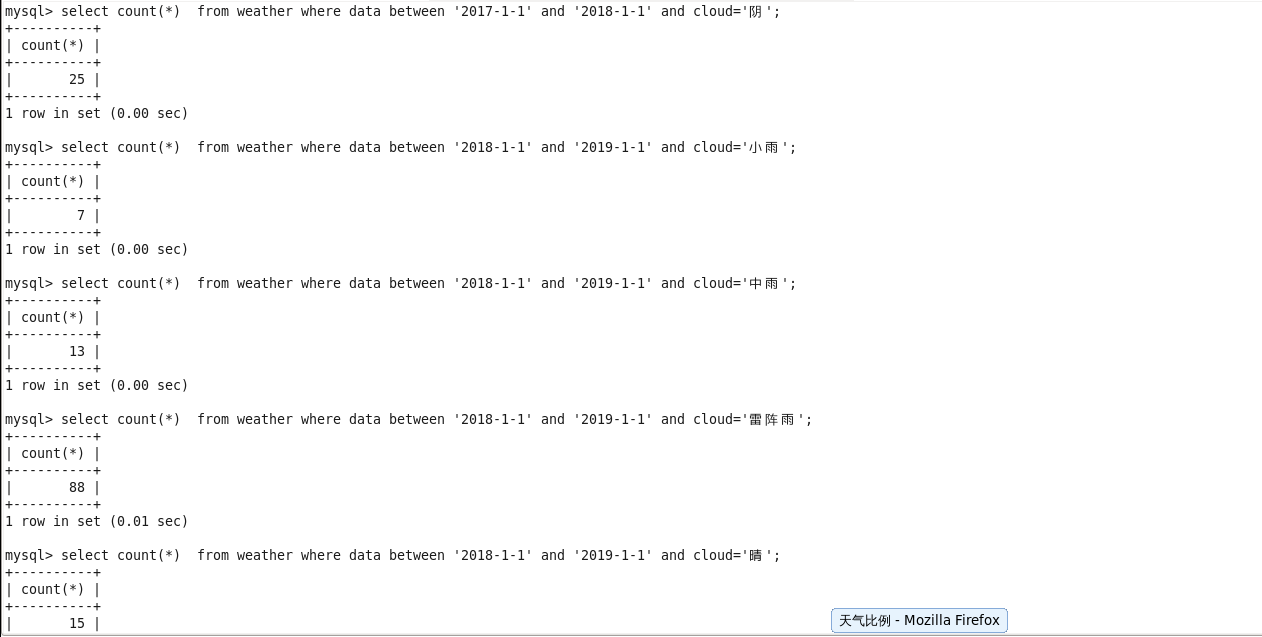


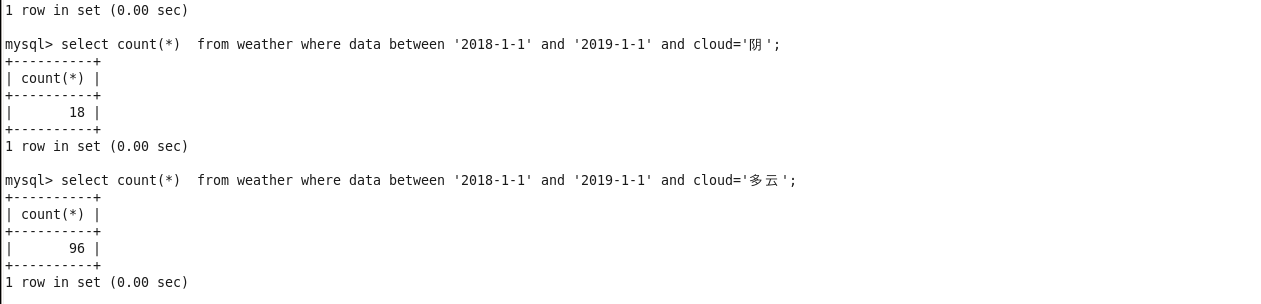


使用sql语句查询数据验证：

|  |
| --- |
| select count(\*) from weather where data between '2017-1-1' and '2018-1-1' and cloud='小雨';  select count(\*) from weather where data between '2017-1-1' and '2018-1-1' and cloud='中雨';  select count(\*) from weather where data between '2017-1-1' and '2018-1-1' and cloud='雷震雨';  select count(\*) from weather where data between '2017-1-1' and '2018-1-1' and cloud='晴';  select count(\*) from weather where data between '2017-1-1' and '2018-1-1' and cloud='多云';  select count(\*) from weather where data between '2017-1-1' and '2018-1-1' and cloud='阴';  select count(\*) from weather where data between '2018-1-1' and '2019-1-1' and cloud='小雨';  select count(\*) from weather where data between '2018-1-1' and '2019-1-1' and cloud='中雨';  select count(\*) from weather where data between '2018-1-1' and '2019-1-1' and cloud='雷阵雨';  select count(\*) from weather where data between '2018-1-1' and '2019-1-1' and cloud='晴';  select count(\*) from weather where data between '2018-1-1' and '2019-1-1' and cloud='阴';  select count(\*) from weather where data between '2018-1-1' and '2019-1-1' and cloud='多云'; |







通过上述数据，发现饼图的每一个部分和数据库的数据相一致，符合题目要求。

**5． 总结**

在本次课程综合设计中我收获颇多，包括从单slave扩展到2个slave的hadoop、hive上去，再有使用了更多、更复杂的hive的sql语句，基本上简单的操作都有所了解，还有就是又一次的使用了可视化工具，这是再课程实验的基础上的自己尝试，虽然过程中也出现了不少的bug，但是当第一个图出现的时候还是很开心的，最后还就是自己装了一遍spark，虽然过程有些坎坷，但是装完之后感觉也就没什么了，然后又用了不是很长的时间学习了下scala以及spark-shell下的spark编程，跑出来的第一条有意义的语句也同时意味着我对spark有了更深的认识与了解，虽然说这个了解还是比较基础的，但是一定会更加深入的去了解。自学spark可能在一定程度上也意味着本门课程的成功，因为怎么说也是自学了一门新技术。总的来说，这门课收获颇多，感受颇深，受益匪浅！