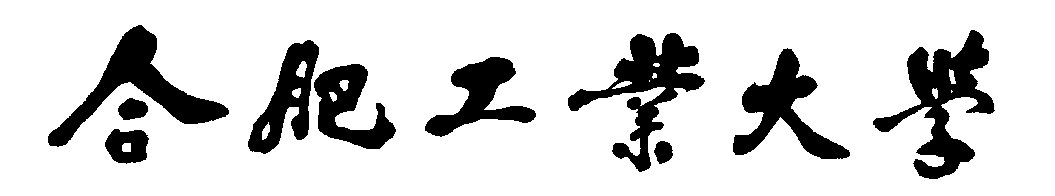
****

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

说 明

实验报告是关于实验教学内容、过程及效果的记录和总结，因此，应注意以下事项和要求：

1．每个实验单元在50页的篇幅内完成一份报告。“实验单元”指按照实验指导书规定的实验内容。若篇幅不够，可另附纸。

2、各实验的**预习部分的内容是进入实验室做实验的必要条件**，请按要求做好预习。

3．实验报告要求：**书写工整规范，语言表达清楚，数据和程序真实。理论联系实际，认真分析实验中出现的问题与现象，总结经验。**

4．参加实验的每位同学应独立完成实验报告的撰写，其中程序或相关的设计图纸也可以采用打印等方式粘贴到报告中。严禁抄袭或拷贝，否则，一经查实，按作弊论取，并取消理论课考试资格。

5．实验报告作为评定实验成绩的依据。

目录

[**实验一 Hadoop平台上部署WordCount程序** 4](#_Toc8324461)

[**一、** **实验目的和要求** 4](#_Toc8324462)

[**二、** **实验任务** 4](#_Toc8324463)

[**三、** **实验准备方案** 4](#_Toc8324464)

[**四、** **实验用仪器、设备** 4](#_Toc8324465)

[**五、** **实验内容与步骤（过程及数据记录）** 4](#_Toc8324466)

[**六、** **感想、体会、建议** 9](#_Toc8324467)

[**七、** **实验成绩** 9](#_Toc8324468)

[**实验二 统计某电商网站买家收藏商品数量** 10](#_Toc8324469)

[**一、** **实验目的和要求** 10](#_Toc8324470)

[**二、** **实验任务** 10](#_Toc8324471)

[**三、** **实验准备方案** 11](#_Toc8324472)

[**四、** **实验用仪器、设备** 11](#_Toc8324473)

[**五、** **实验内容与步骤（过程及数据记录）：** 11](#_Toc8324474)

[**六、** **感想、体会、建议** 14](#_Toc8324475)

[**七、** **实验成绩** 15](#_Toc8324476)

[**实验三 评论数据采集处理与可视化** 16](#_Toc8324477)

[**一、** **实验目的和要求** 16](#_Toc8324478)

[**二、** **实验任务** 16](#_Toc8324479)

[**三、** **实验准备方案** 16](#_Toc8324480)

[**四、** **实验用仪器、设备** 16](#_Toc8324481)

[**五、** **实验内容与步骤（过程及数据记录）** 16](#_Toc8324482)

[**1.** **数据采集** 16](#_Toc8324483)

[**2.** **数据清洗搭建解析框架** 20](#_Toc8324484)

[**3.** **分词项目搭建** 24](#_Toc8324485)

[**4.** **可视化** 26](#_Toc8324486)

[**六、** **感想、体会、建议** 38](#_Toc8324487)

[**七、** **实验成绩** 38](#_Toc8324488)

**实验序号及名称：实验 一 Hadoop平台上部署WordCount程序**

实验时间∶ 2019年 4月 2 日

1. **实验目的和要求**
2. 学会使用基本的命令行编译运行mapreduce程序。
3. 对mapreduce代码编写有基本的熟悉，对mapreduce执行过程有更深层次的了解。
4. 在实验过程中学会自己查找解决问题的方法，培养处理独自处理问题的能力。
5. **实验任务**
6. 找一个文本文件
7. 使用mapreduce统计每一个词出现的次数
8. **实验准备方案**

包括以下内容：（硬件类实验：实验原理、实验线路、设计方案等）

（软件类实验：所采用的系统、组件、工具、核心方法、框架或流程图、程序清单等）

实验采用的系统：centos 6.5、hadoop2.5.2

工具：VMware WorkStation

1. **实验用仪器、设备**

实验仪器：计算机一台

1. **实验内容与步骤（过程及数据记录）**
2. 将我们将 Hadoop 的 classhpath 信息添加到 CLASSPATH 变量中，在~/.bashrc中添加最后两行命令。

|  |
| --- |
| >>vi ~/.bashrc |

编辑输入：

|  |
| --- |
| export HADOOP\_HOME=~/hadoop-2.5.2/  export CLASSPATH=$($HADOOP\_HOME/bin/hadoop classpath):$CLASSPATH |

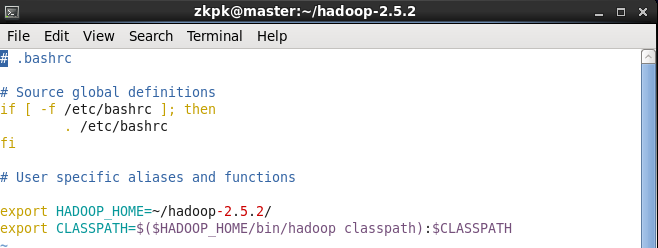


图1 配置CLASSPATH变量

1. 编译java文件

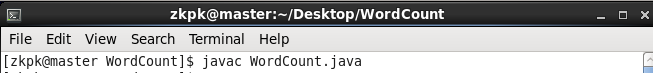


图2 编译java文件

1. 将class文件打包成jar包

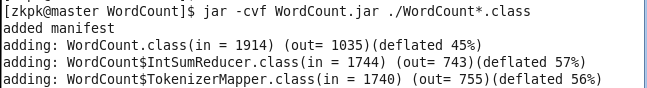


图3 将class文件打包

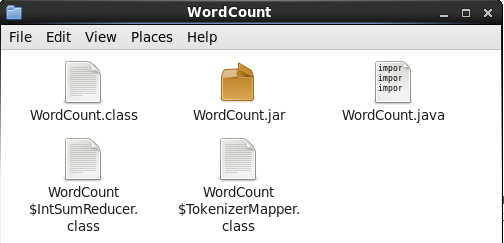


图4 生成的jar包

1. 新建input1文件夹：



图5 新建文件夹存储数据

1. 上传待统计文本到输入文件夹



图6 上传文件到HDFS

1. 运行WordCount.java编译生成的jar包。

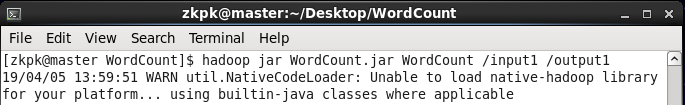


图7 使用Hadoop运行jar包

1. 查看统计结果（此处为了方便直接在eclipse中访问HDFS）



图7 待统计文件夹

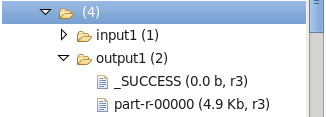


图8 生成结果文件

1. 查看WordCount统计结果

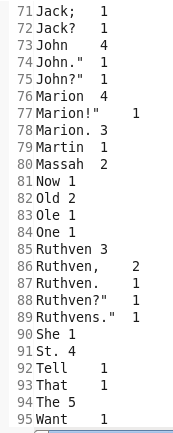


图9 词频统计结果。

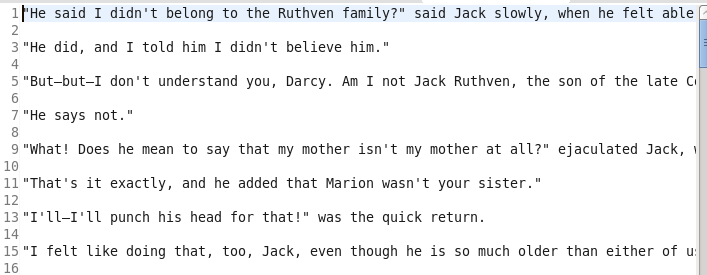


图10 待统计原文

在图10中是随意找的一篇英文小说，可以看到，在图9中由于使用的是空格作为分隔符，因此有些单词是连着标点符号的，这显然不是我们想要的，因此对其进一步进行处理。处理之后的结果如图11所示。可以看到多余的标点符号等都已经得到了有效的处理，得到的词频统计结果更加的准确，没有导致因为标点符号的不同导致算作两个词的情况发生。

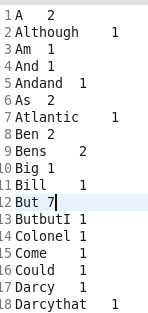


图11 进一步处理的统计结果

**9．部分代码解释**

**Map函数：**

private static final IntWritable one = new IntWritable(1);

private Text word = new Text();

public TokenizerMapper(){}

public void map(Object key, Text value, Mapper<Object, Text, Text,

IntWritable>.Context context) throws IOException, InterruptedException {

/\* 设置输入key、value对以及输出key、value对的格式 \*/

StringTokenizer itr = new StringTokenizer(value.toString());

/\* 以空格为标识符进行分词 \*/

while(itr.hasMoreTokens()){

String stri2=itr.nextToken().replaceAll("\\pP","");

/\* 对标点符号进行处理，使用正则去掉标点符号 \*/

this.word.set(stri2);

context.write(this.word, one);

/\* 发送key-Value对 \*/

}

**Redeuce函数：**

private IntWritable result = new IntWritable();

public IntSumReducer(){}

public void reduce(Text key, Iterable<IntWritable> values,

Reducer<Text, IntWritable,Text,IntWritable>.Context context)

throws IOException,

InterruptedException{

int sum = 0;

IntWritable val;

for(Iterator itr = values.iterator(); itr.hasNext(); sum += val.get()){

val = (IntWritable)itr.next();

}

/\* 对相同key的value列表进行出现次数的相加 \*/

this.result.set(sum);

context.write(key, this.result);

/\* 发送key、value对,分别为单词以及出现的次数 \*/

1. **感想、体会、建议**

通过本次实验，我对mapreduce代码的编写有了基本的掌握，虽然在上学期已经在云计算课程中编写过相关的代码，现在都相当于是重新熟悉一下mapreduce代码编写的过程。此外，通过mapreduce代码的编写可以在一定程度上加深对map以及reduce过程的了解与掌握，通过本次实验，为下面对mapreduce的进一步应用做了准备。

1. **实验成绩**

**指导教师签名：**

年 月 日

**实验序号及名称：实验 二 统计某电商网站买家收藏商品数量**

实验时间∶ 2019 年 4 月 15 日

1. **实验目的和要求**
2. 编写mapreduce代码统计每位用户收藏商品的数量。
3. 掌握eclipse如何运行mapreduce程序。
4. 进一步巩固对mapreduce的了解
5. **实验任务**

现有某电商网站用户对商品的收藏数据，记录了用户收藏的商品id以及收藏日期，名为buyer\_favorite1。buyer\_favorite1包含：买家id，商品id，收藏日期这三个字段，数据以“\t”分割，样本数据及格式如下：

1. 买家id   商品id    收藏日期
2. 10181   1000481   2010-04-04 16:54:31
3. 20001   1001597   2010-04-07 15:07:52
4. 20001   1001560   2010-04-07 15:08:27
5. 20042   1001368   2010-04-08 08:20:30
6. 20067   1002061   2010-04-08 16:45:33
7. 20056   1003289   2010-04-12 10:50:55
8. 20056   1003290   2010-04-12 11:57:35
9. 20056   1003292   2010-04-12 12:05:29
10. 20054   1002420   2010-04-14 15:24:12
11. 20055   1001679   2010-04-14 19:46:04
12. 20054   1010675   2010-04-14 15:23:53
13. 20054   1002429   2010-04-14 17:52:45
14. 20076   1002427   2010-04-14 19:35:39
15. 20054   1003326   2010-04-20 12:54:44
16. 20056   1002420   2010-04-15 11:24:49
17. 20064   1002422   2010-04-15 11:35:54
18. 20056   1003066   2010-04-15 11:43:01
19. 20056   1003055   2010-04-15 11:43:06
20. 20056   1010183   2010-04-15 11:45:24
21. 20056   1002422   2010-04-15 11:45:49
22. 20056   1003100   2010-04-15 11:45:54
23. 20056   1003094   2010-04-15 11:45:57
24. 20056   1003064   2010-04-15 11:46:04
25. 20056   1010178   2010-04-15 16:15:20
26. 20076   1003101   2010-04-15 16:37:27
27. 20076   1003103   2010-04-15 16:37:05
28. 20076   1003100   2010-04-15 16:37:18
29. 20076   1003066   2010-04-15 16:37:31
30. 20054   1003103   2010-04-15 16:40:14
31. 20054   1003100   2010-04-15 16:40:16

要求编写MapReduce程序，统计每个买家收藏商品数量

1. **实验准备方案**

包括以下内容：（硬件类实验：实验原理、实验线路、设计方案等）

（软件类实验：所采用的系统、组件、工具、核心方法、框架或流程图、程序清单等）

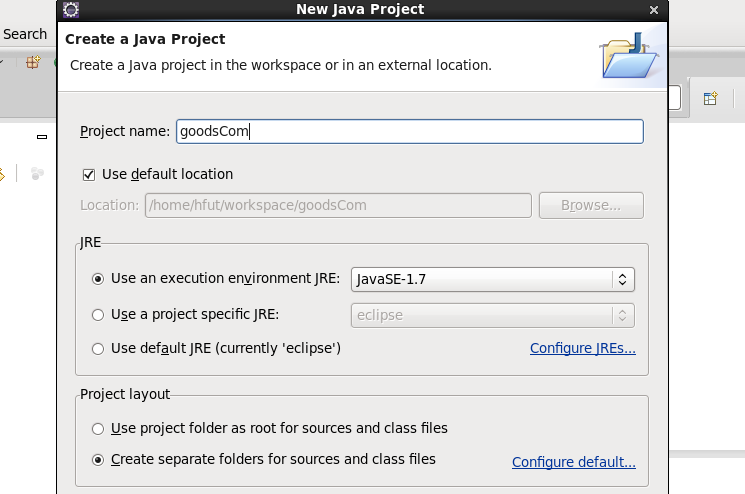
实验采用的系统：centos 6.5、hadoop2.5.2

工具：VMware WorkStation

1. **实验用仪器、设备**

实验仪器：计算机一台

1. **实验内容与步骤（过程及数据记录）：**
2. 新建工程



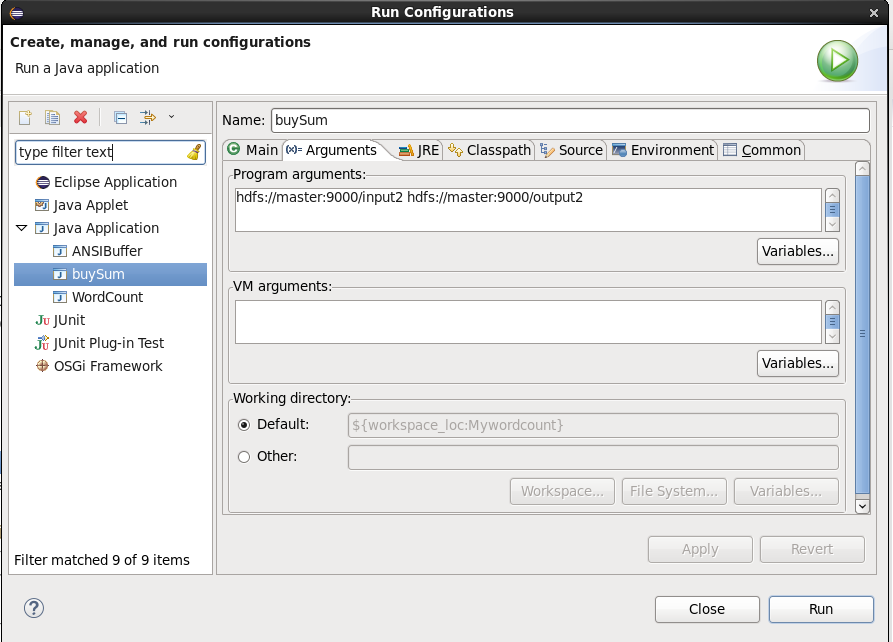
图一

1. 待处理数据结构

表一 原始数据结构

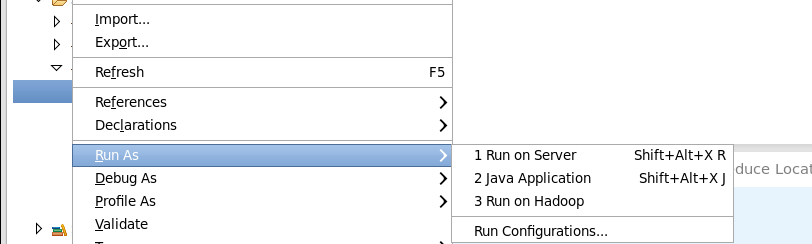
|  |  |  |
| --- | --- | --- |
| 买家ID | 商品ID | 收藏时间 |
| 10181 | 1000481 | 2010-04-04 16:54:31 |
| 20001 | 1001597 | 2010-04-07 15:07:52 |
| 20001 | 1001560 | 2010-04-07 15:08:27 |
| 20042 | 1001368 | 2010-04-08 08:20:30 |
| 20067 | 1002061 | 2010-04-08 16:45:33 |
| 20056 | 1003289 | 2010-04-12 10:50:55 |
| 20056 | 1003290 | 2010-04-12 11:57:35 |
| 20056 | 1003292 | 2010-04-12 12:05:29 |

1. 设置输入输出路径

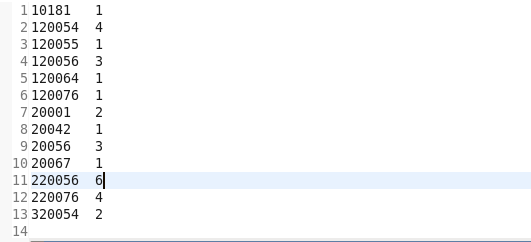


图二 设置输入输出路径

1. Run->run on Hadoop:



1. 实验结果



图三 统计结果

1. 处理数据流程

在map函数中将每一个key、value对的value进行分割，分隔符为’\t’,然后提取提取分割后的列表中的第一项（即用户ID），然后发送一个key、value对。在reduce函数中将相同key值的value相加，即可得到用户收藏的商品总数。

1. 部分代码

Map**函数**

String[] words = value.toString().split("\t");

/\* 使用’\t’作为分隔符。得到切分列表 \*/

word.set(words[0]);

context.write(word, one);

/\* 发送一个key，value对，表示用户收藏了一项商品。 \*/

Reduce函数

int sum = 0;

for (IntWritable val : values) {

sum += val.get();

}

/\* 对于key相同，即同一个用户，统计他收藏的商品数量 \*/

result.set(sum);

设置将要发出的value值。

context.write(key, result);

/\* 发送一个key、value对，key和value

分别为用户ID以及用户收藏的商品数量 \*/

主函数：

job.setJarByClass(buSum.class);

/\* 通过class名字获取到class所在的jar包 \*/

job.setMapperClass(TokenizerMapper.class);

job.setCombinerClass(IntSumReducer.class);

job.setReducerClass(IntSumReducer.class);

/\* 设置map、reduce和conbiner的所在类 \*/

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

/\* 设置输入输出参数格式 \*/

FileInputFormat.addInputPath(job, new Path(otherArgs[0]));

FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));

/\* 设置输入输出路径 \*/

1. **感想、体会、建议**

通过本次实验是在上一次WordCount的基础上进一步的掌握mapreduce编程，通过自己动手写代码可以对mapreduce的过程有更深层次的认识。同时也通过eclipse更方便的运行mapreduce程序。此外为了能够使用eclipse也需要续费一定的时间，在此并未详细给出安装过程，通过前面的实验教程可以相对容易的配置成功，避免了很多坑，但是在以往自己配置的过程中还是出现过很多问题，当时耗费了挺久的时间，归因于网上的部分教程很不详细以及自己对linux系统的不熟悉。但是在本次的过程就相对的方便了很多，减少了很多麻烦，过程比较清晰、详细，在此感谢一下老师花费时间给出的详细教程以及上课时的演示。

1. **实验成绩**

**指导教师签名：**

年 月 日

**实验序号及名称：实验 三**  **评论数据采集处理与可视化**

实验时间∶ 2019 年 4 月 28 日

1. **实验目的和要求**
2. 爬取京东上的1000条评论
3. 将爬取的信息格式化
4. 将评论存储到mysql中
5. 使用jieba等分词工具分词
6. 制作词云进行可视化
7. **实验任务**

爬取京东或淘宝某一商品的评论1000条，使用jieba分词等分词工具分词，统计词频（使用MapReduce或HBase或Hive），并以词云、统计表的方式可视化呈现。

1. **实验准备方案**

包括以下内容：（硬件类实验：实验原理、实验线路、设计方案等）

（软件类实验：所采用的系统、组件、工具、核心方法、框架或流程图、程序清单等）

实验采用的系统：centos 6.5、hadoop2.5.2

工具：VMware WorkStation

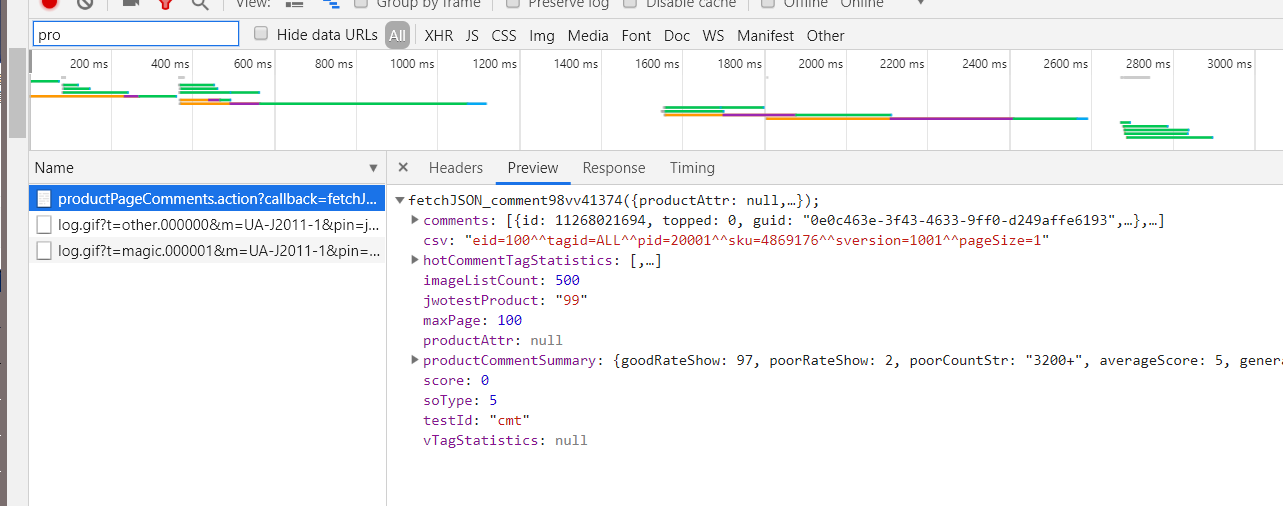
其他工具：jieba分词工具，echart，Tomcat server等

1. **实验用仪器、设备**

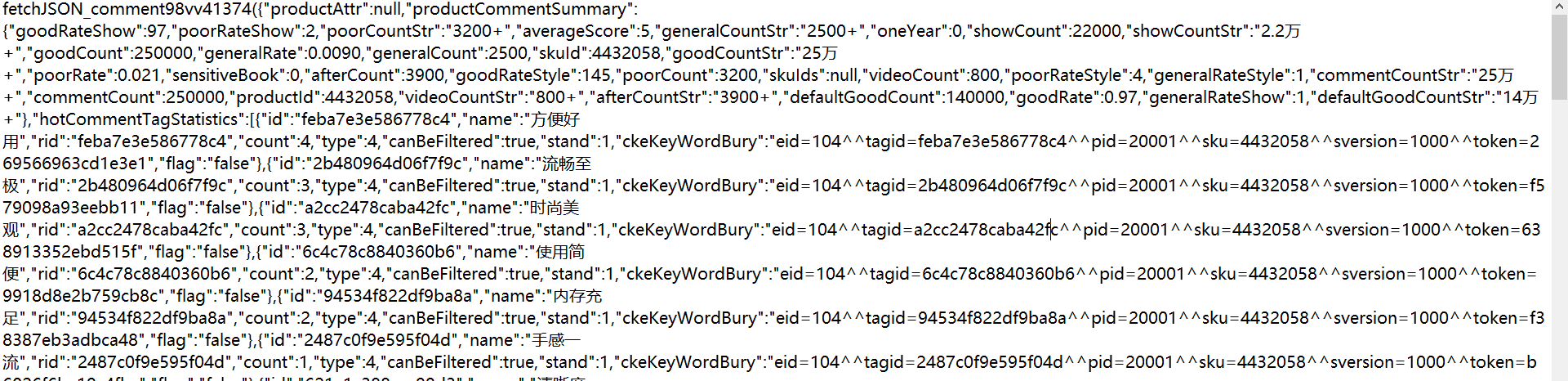
笔记本一台

1. **实验内容与步骤（过程及数据记录）**
2. **数据采集**

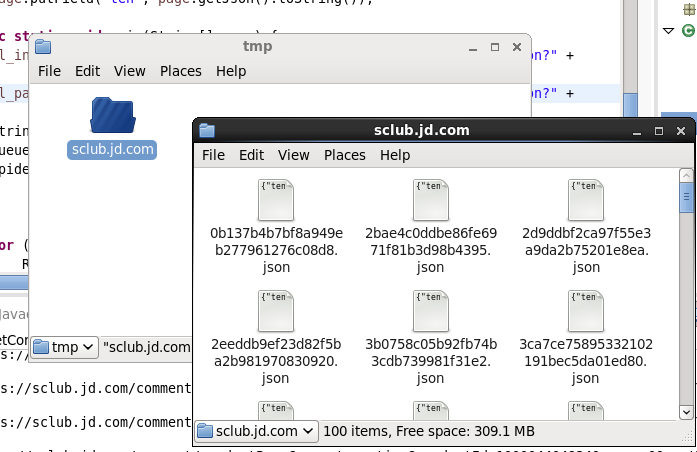
1.1 进入网址<https://item.jd.com/4432058.html>，查看浏览器请求的数据文件，在network->filter中输入pro选择productPageComments...文件，可以看到文件内容如下图所示：



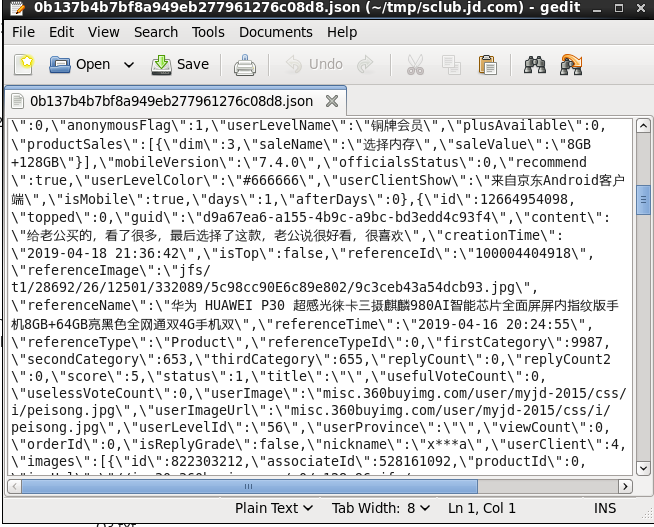
1.2 选中文件，右键复制call back在浏览器中打开，查看返回的Json形式的评论信息：



1.3 新建java工程，以java project的形式运行爬虫代码，查看使用java爬虫下载的数据是否成功、正确：



从上图可以看到确实是下载了100个文件，随即打开一个文件查看内容：

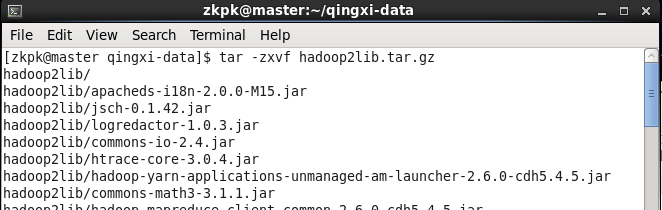


1.4 本次评论所使用的爬虫代码，更改了代理等设置：

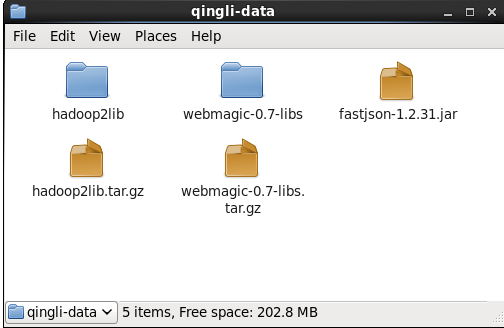
|  |
| --- |
| package my.webmagic;  import us.codecraft.webmagic.Page;  import us.codecraft.webmagic.Request;  import us.codecraft.webmagic.Site;  import us.codecraft.webmagic.Spider;  import us.codecraft.webmagic.pipeline.ConsolePipeline;  import us.codecraft.webmagic.pipeline.JsonFilePipeline;  import us.codecraft.webmagic.processor.PageProcessor;  import us.codecraft.webmagic.scheduler.QueueScheduler;  public class GetComments implements PageProcessor {  // 对爬取站点的一些属性进行设置，例如：设置域名，设置代理等；  private Site site = Site.*me*()  .setDomain("sclub.jd.com")  .setSleepTime(2000)  .addCookie("TrackID", "15yCkQ1-EBrUjdqR00DQ5QscyENaUom43JpYSvjMJpq6anA50TrPtspNLl6U8XaLsrVJcwNmEdABTf0HNRnM0w2bDvqpTE7RZbUMrf46d7Y0")  .addCookie("\_\_jda", "122270672.15289023667021425393993.1528902367.1528938475.1528969012.4")  .addCookie("\_\_jdb", "122270672.7.15289023667021425393993|4.1528969012")  .addCookie("\_\_jdc ", "122270672")  .addCookie("\_\_jdu", "15289023667021425393993")  .addCookie("\_\_jdv", "122270672|cps.youmai.com|t\_1000049399\_39857496|tuiguang|aed1c4304f694e599245a430daeddb2f|1528905344545")  .addHeader("User-Agent", "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.101 Safari/537.3")  .addHeader("Accept", "\*/\*")  .addHeader("Accept-Encoding", "gzip, deflate, br")  .addHeader("Accept-Language", "en-US,en;q=0.8")  .addHeader("Connection", "keep-alive")  .addHeader("Referer", "https://www.jd.com/")  .setUserAgent("Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.101 Safari/537.3");    public Site getSite() {  return site;  }  public void process(Page page) {  page.putField("ten", page.getJson().toString());  }  public static void main(String[] args) {  String url\_init = "https://sclub.jd.com/comment/productPageComments.action?" + "productId=100004404934&score=0&sortType=5&page=0&pageSize=10&isShadowSku=0&fold=1";  String url\_pattern = "https://sclub.jd.com/comment/productPageComments.action?" + "productId=100004404934&score=0&sortType=5&pageSize=10&isShadowSku=0&fold=1&page=";  String output = "/home/zkpk/tmp/";  QueueScheduler scheduler = new QueueScheduler();  Spider spider = Spider.*create*(new GetComments()).addUrl(url\_init)  .setScheduler(scheduler)  .addPipeline(new JsonFilePipeline(output))  .addPipeline(new ConsolePipeline());  for (int i = 1; i < 100; i++) {  Request request = new Request();  request.setUrl(url\_pattern + i);  scheduler.push(request, spider);  }  spider.thread(50).run();  }  } |

1. **数据清洗搭建解析框架**

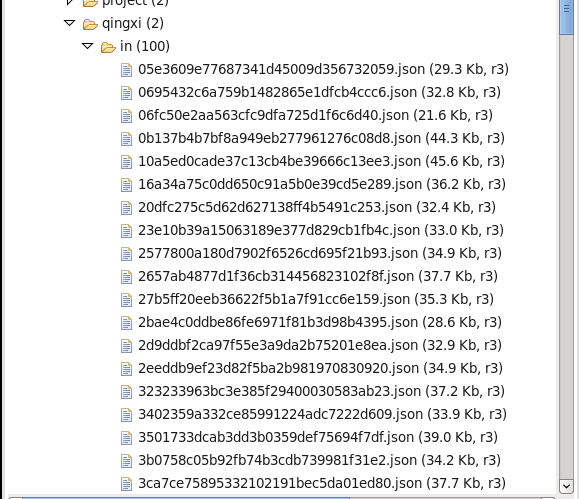
2.1 准备hadoop2lib.tar.gz和fastjson-1.2.31.jar软件包并解压到当前文件夹：



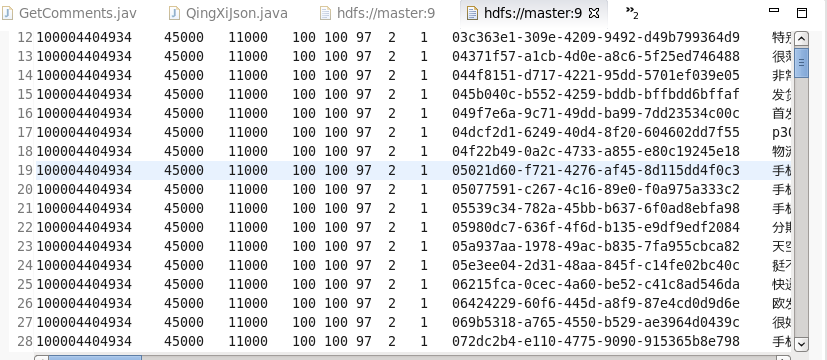
2.2 解压以及复制结果如下：



2.2 将所有评论上传到HDFS，并查看是否上传成功，此处为在eclipse中查看：



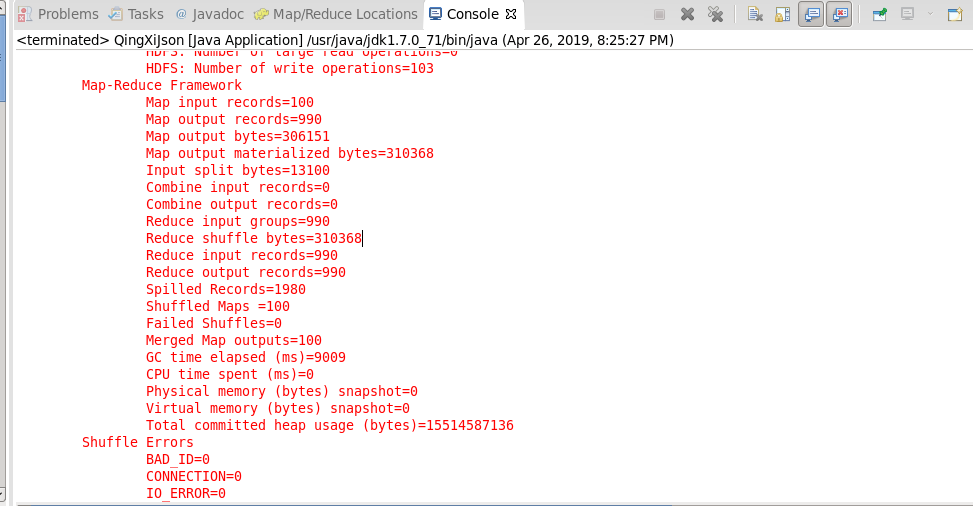
2.3 对HDFS中的评论进行清洗，将实用信息格式化并查看清洗结果：



2.4 将清洗结果从hdfs上下载到本地：



2.5 解决控制台不输出运行信息的问题，将~/hadoop-2.5.2/etc/hadoop/log4j.properties文件，拷贝到qingxi项目的src目录下。查看到输出结果，通过log日志排错，问题成功解决。

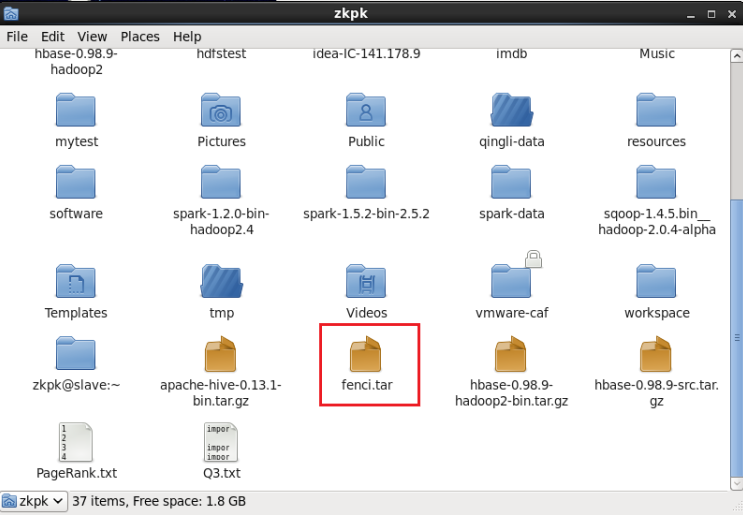


2.6 数据清洗代码

|  |
| --- |
| package my.mr;  import java.io.IOException;  import org.apache.hadoop.fs.Path;  import org.apache.hadoop.io.Text;  import org.apache.hadoop.mapreduce.Job;  import org.apache.hadoop.mapreduce.Mapper;  import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;  import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;  import com.alibaba.fastjson.JSONArray;  import com.alibaba.fastjson.JSONObject;  public class QingXiJson {  public static void main(String[] args) throws IOException, ClassNotFoundException, InterruptedException {  Job job = Job.getInstance();  job.setJobName("QingXiJson");  job.setJarByClass(QingXiJson.class);    job.setMapperClass(doMapper.class);  //job.setReducerClass(doReducer.class);  job.setOutputKeyClass(Text.class);  job.setOutputValueClass(Text.class);  Path in = new Path("hdfs://master:9000/qingxi/in");  Path out = new Path("hdfs://master:9000/qingxi/out/1");  FileInputFormat.addInputPath(job, in);  FileOutputFormat.setOutputPath(job, out);  System.exit(job.waitForCompletion(true) ? 0 : 1);  }  public static class doMapper extends Mapper<Object, Text, Text, Text> {  @Override  protected void map(Object key, Text value, Context context)  throws IOException, InterruptedException {  String initJsonString = value.toString();  JSONObject initJson = JSONObject.parseObject(initJsonString );  if (!initJsonString.contains("productCommentSummary") && !initJsonString.contains("comments")) {  return;  }    JSONObject myjson = initJson.getJSONObject("ten");    JSONObject productCommentSummary = myjson.getJSONObject("productCommentSummary");  String productId = productCommentSummary.get("productId").toString();  String commentCount = productCommentSummary.get("commentCount").toString();  String goodCount = productCommentSummary.get("goodCount").toString();  String generalCount = productCommentSummary.get("generalCount").toString();  String poorCount = productCommentSummary.get("poorCount").toString();  String goodRateShow = productCommentSummary.get("goodRateShow").toString();  String generalRateShow = productCommentSummary.get("generalRateShow").toString();  String poorRateShow = productCommentSummary.get("poorRateShow").toString();  /\* comments 包括十条评论 \*/  JSONArray comments = myjson.getJSONArray("comments");  for (int i = 0; i < comments.size(); i++) {  JSONObject comment = comments.getJSONObject(i);  String guid = comment.getString("guid");  String content = comment.getString("content").replace('\n', ' ');  String creationTime = comment.getString("creationTime");  String score = comment.getString("score");  String nickname = comment.getString("nickname");  String userLevelName = comment.getString("userLevelName");  String userClientShow = comment.getString("userClientShow");  String isMobile = comment.getString("isMobile");  String days = comment.getString("days");  StringBuilder sb = new StringBuilder();    sb.append(productId); sb.append("\t");  sb.append(commentCount); sb.append("\t");  sb.append(goodCount); sb.append("\t");  sb.append(generalCount); sb.append("\t");  sb.append( poorCount ); sb.append("\t");  sb.append( goodRateShow ); sb.append("\t");  sb.append( generalRateShow ); sb.append("\t");  sb.append( poorRateShow ); sb.append("\t");  sb.append( guid ); sb.append("\t");  sb.append( content ); sb.append("\t");  sb.append( creationTime ); sb.append("\t");  sb.append( score ); sb.append("\t");  sb.append( nickname ); sb.append("\t");  sb.append( userLevelName ); sb.append("\t");  sb.append( userClientShow ); sb.append("\t");  sb.append( isMobile ); sb.append("\t");  sb.append( days );  String result = sb.toString();  context.write(new Text(result), new Text(""));  }  }  }  } |

1. **分词项目搭建**

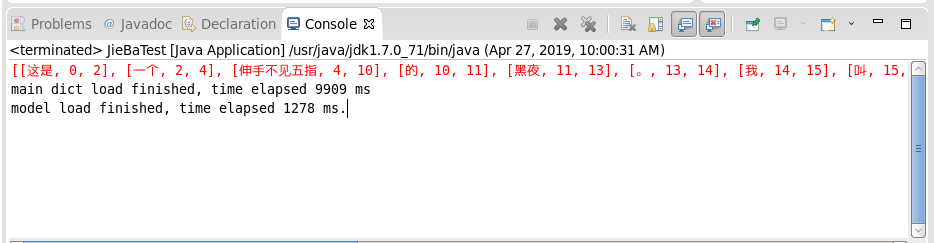
3.1 获取分词项目fenci.tar：



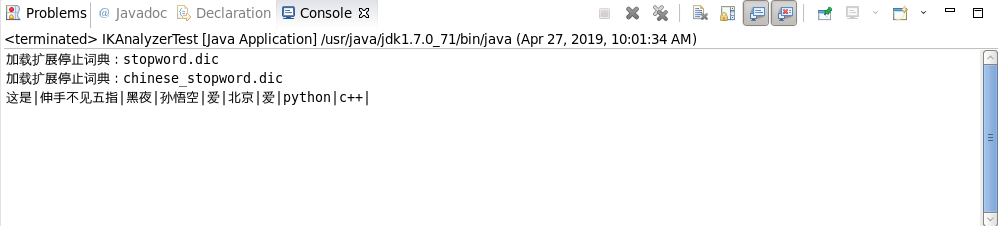
3.2 将fenci.tar导入到当前工程当中：



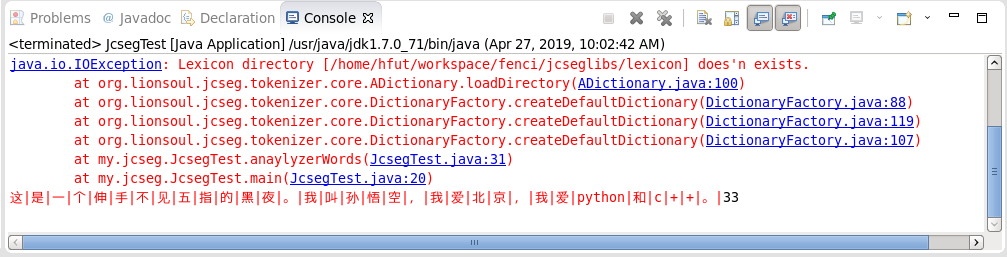
3.3 选择其中的jieBaTest.java文件并运行，查看结巴分词效果：



3.4 选择其中的IKAnalyzerTest.java文件并运行，查看IKAnalyzer分析效果：

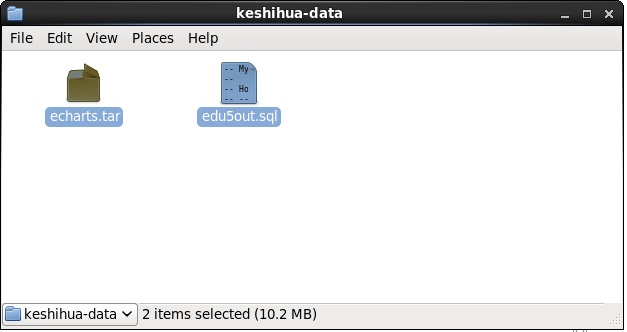


3.5 选择JcsegTest.java问价并运行，查看JcsegTest分词方法：



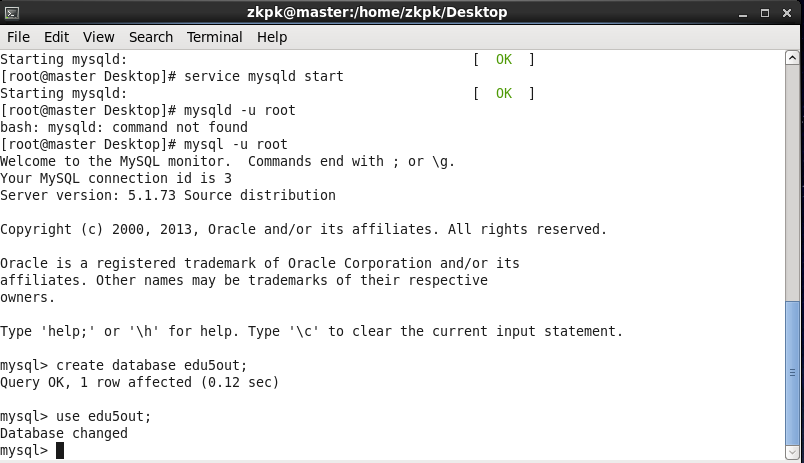
从以上结果中可以清楚的看到使用jieba和IKAnalyzer的分词效果要明显的优于第三个的分词效果，是否可以将人名、地名提取出来在一定程度上可以反应分词工具的好坏。

3.6 加载可视化文件，将echart.tar加载到eclipse中：



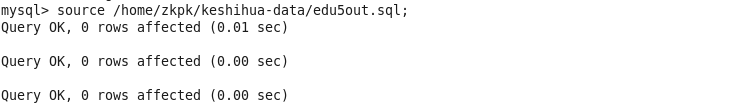
* 1. 使用service mysqld start命令打开mysql并创建新的数据库：

注意：此处使用的root登录。



1. **可视化**

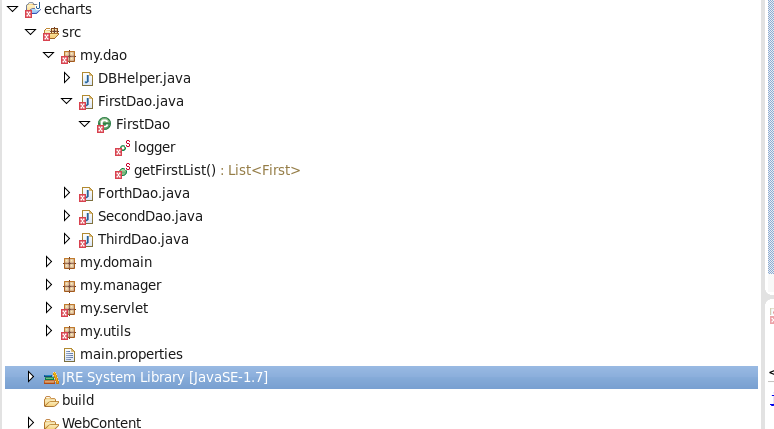
4.1 将/home/zkpk/keshihua-data目录下的edu5out.sql脚本导入到mysql的edu5out库中：



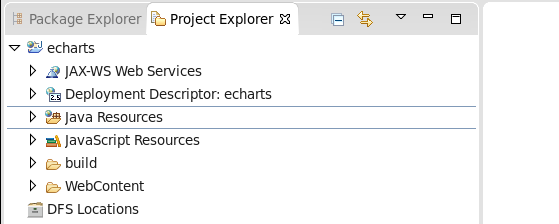
4.2 查看当前数据库（edu\_5out）中存在有哪些表（show tables）：



4.3 导入echart包后的项目情况：



4.4 修改jre system Library，重新配置之后的工程错误得到改正：



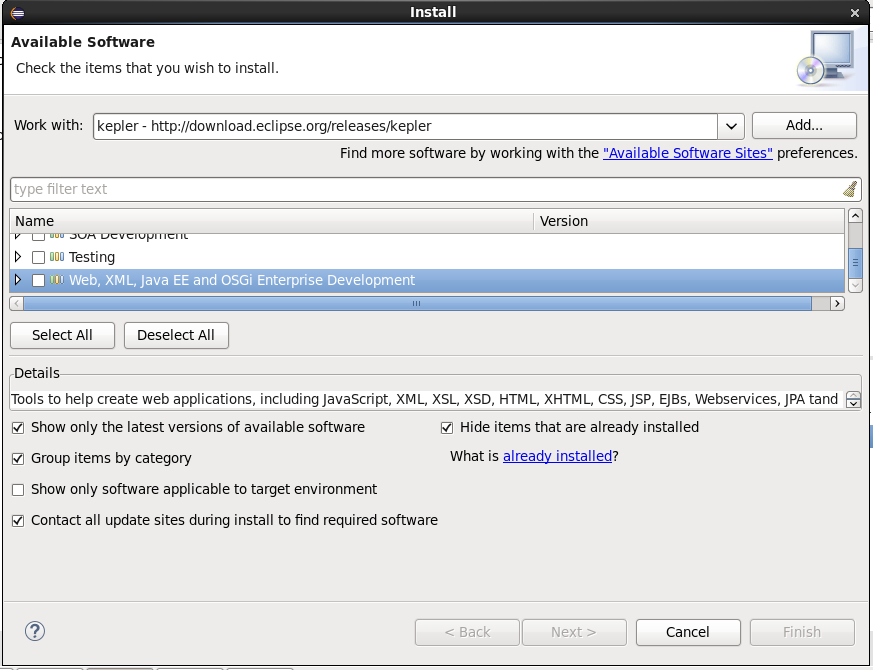
4.5 配置server（Tomcat）

4.5.1解决找不到server选项的问题：

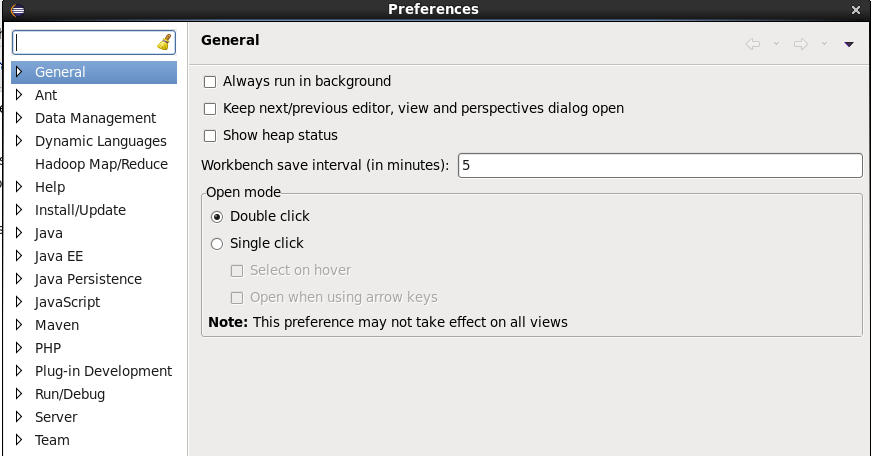
在work with中输入：

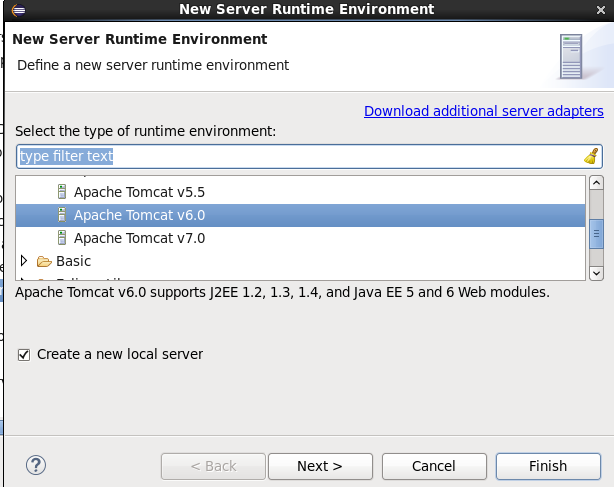
|  |
| --- |
| kepler - http://download.eclipse.org/releases/kepler |

找到Web,XML, Java EE and OSGi Enterprise Development选项打钩，然后接下来所有选择next

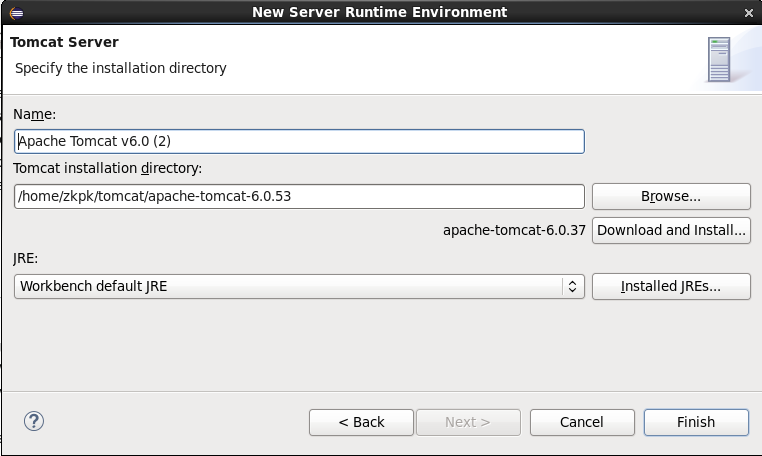


4.5.2 在window->Preference中选择server：

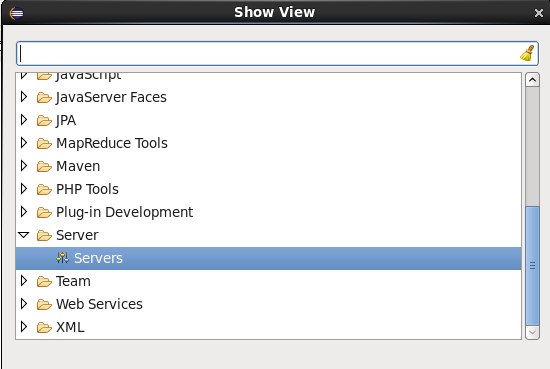




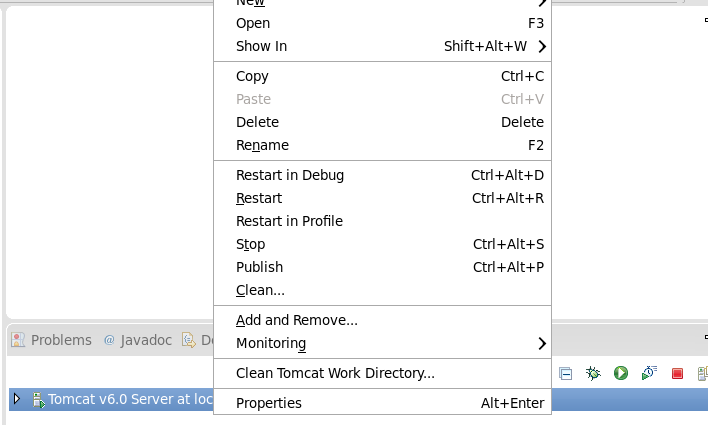
4.5.3 选择tomcat的安装路径：



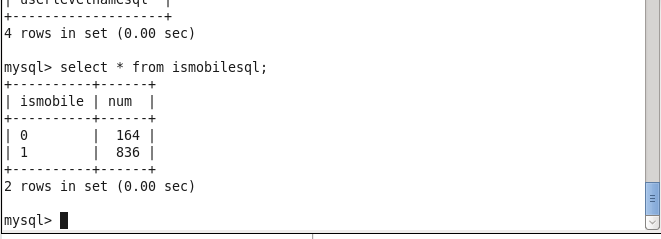
4.5.4 使server选项在下方任务栏显示，选中show view，在server中选中server选项，将其加入快捷方式：



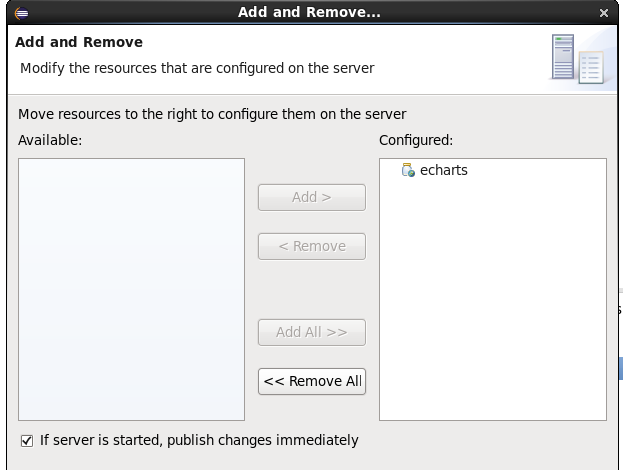
4.5.5 在下方任务栏中选中server，开启server（右键start）：



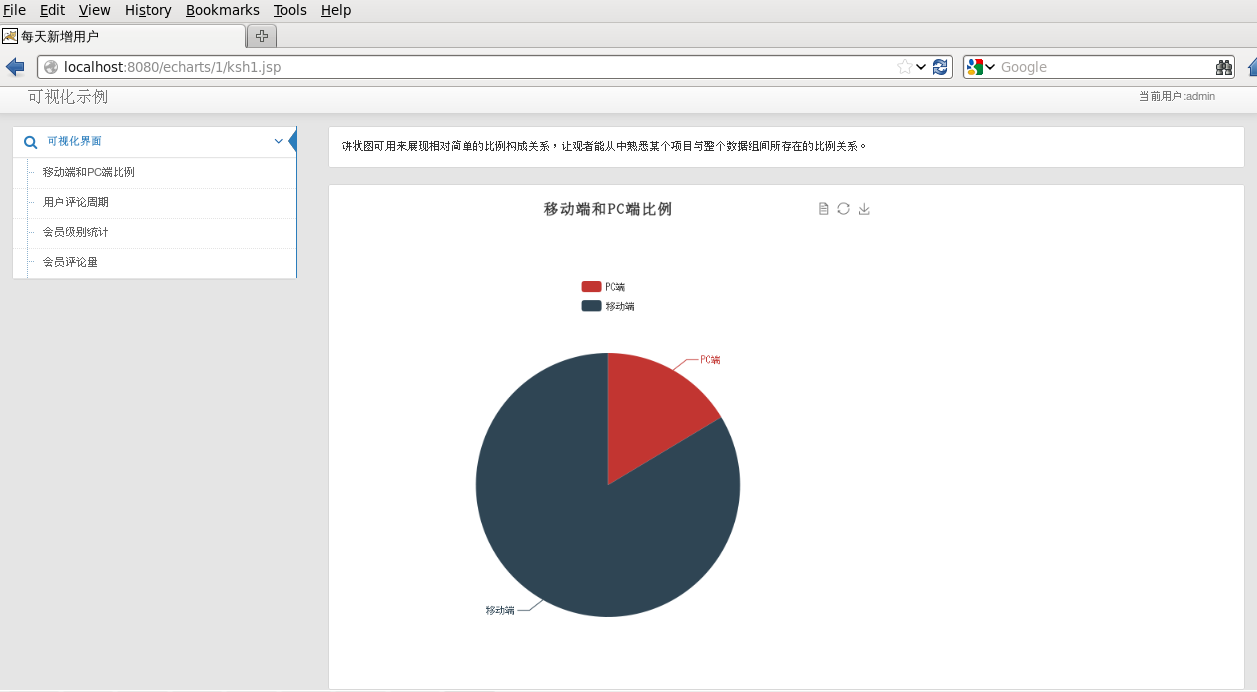
4.6 查看MySQL数据库中的ismobilesql表（select \* from ismobilesql;）其中1代表使用的是移动端，0代表使用的是PC端，可以看到两个数据分别为836和164：



4.7 在下方的信息栏中选中server，右键选中Add and move将echart加入右方configured（使该项目在服务器中运行）：

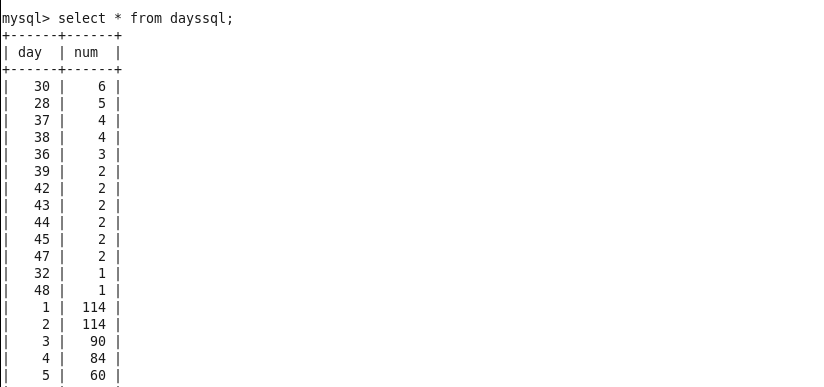


* 1. 在浏览器中输入：<http://localhost:8080/echarts/1/ksh1.jsp>查看效果，可以看到使用移动端和PC端的比例和在上方使用mysql查询到的数据是保持一致的，鼠标移动到饼图上面可以具体的看到每一部分的数量也是占比为836和164。

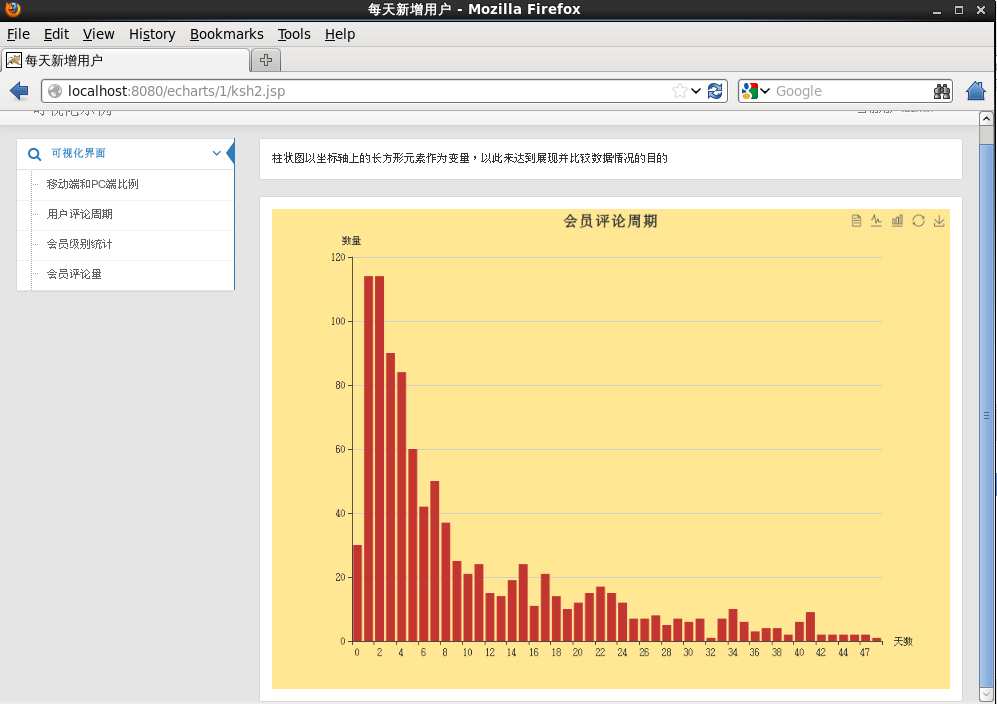


4.9 验证可视化比例是否正确，在mysql中输入下方语句，查看dayssql语句：

|  |  |
| --- | --- |
| select \* from dayssql; |  |

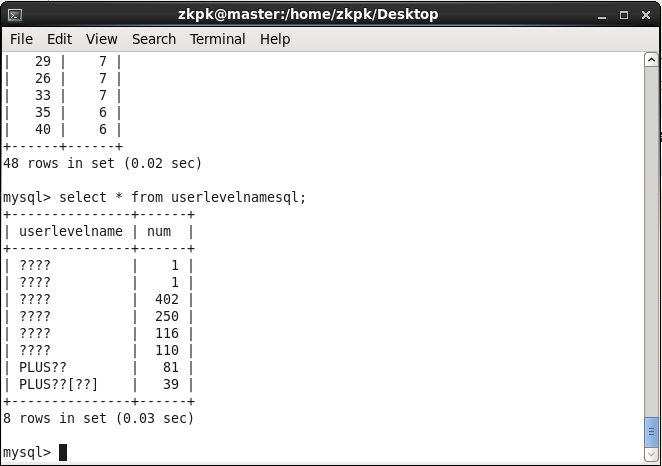


4.10 在浏览器中输入<http://localhost:8080/echarts/1/ksh2.jsp>查看每天的评论数量的直方图，显示效果如下图所示，可以看到和上面的使用mysql查出来的数据保持一致。

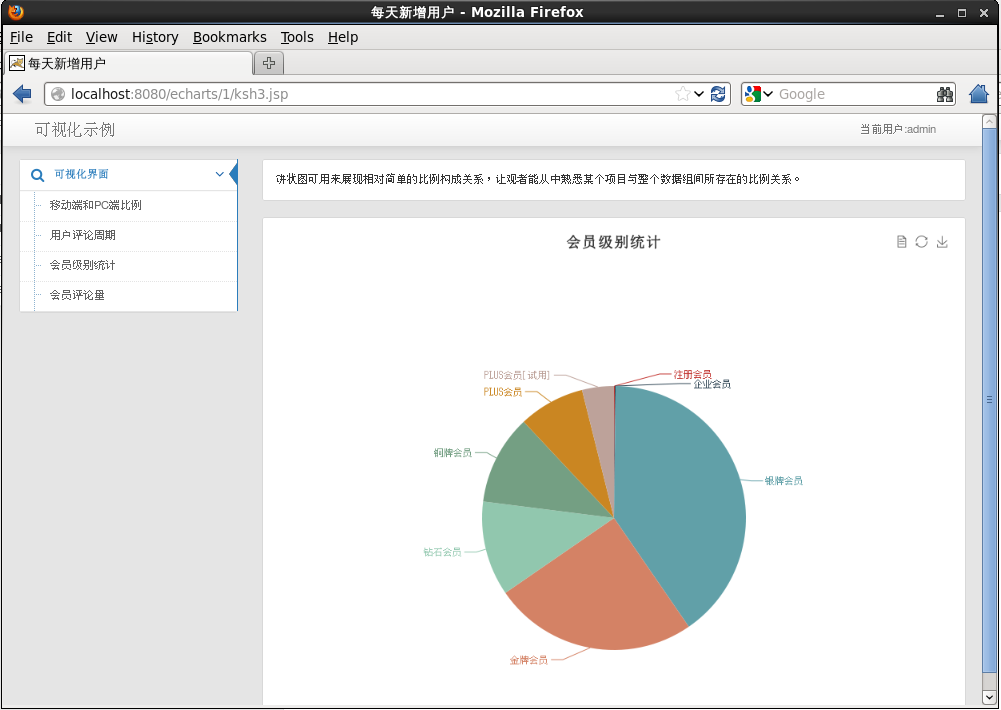


4.11 查看可视化结构是否正确，下图由于在创建数据库的时候编码错误因此出现乱码，在后面的过程中已经改正。

|  |
| --- |
| select \* from userlevelnamesql; |

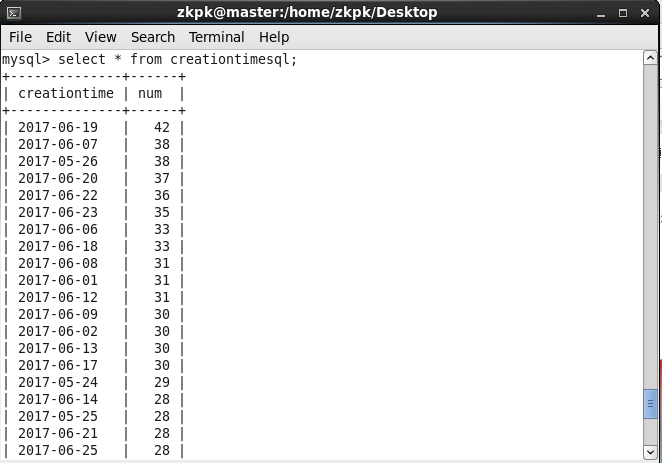


4.12 在浏览器中输入<http://localhost:8080/echarts/1/ksh3.jsp>查看各个等级会员所占的比例和上图一致。

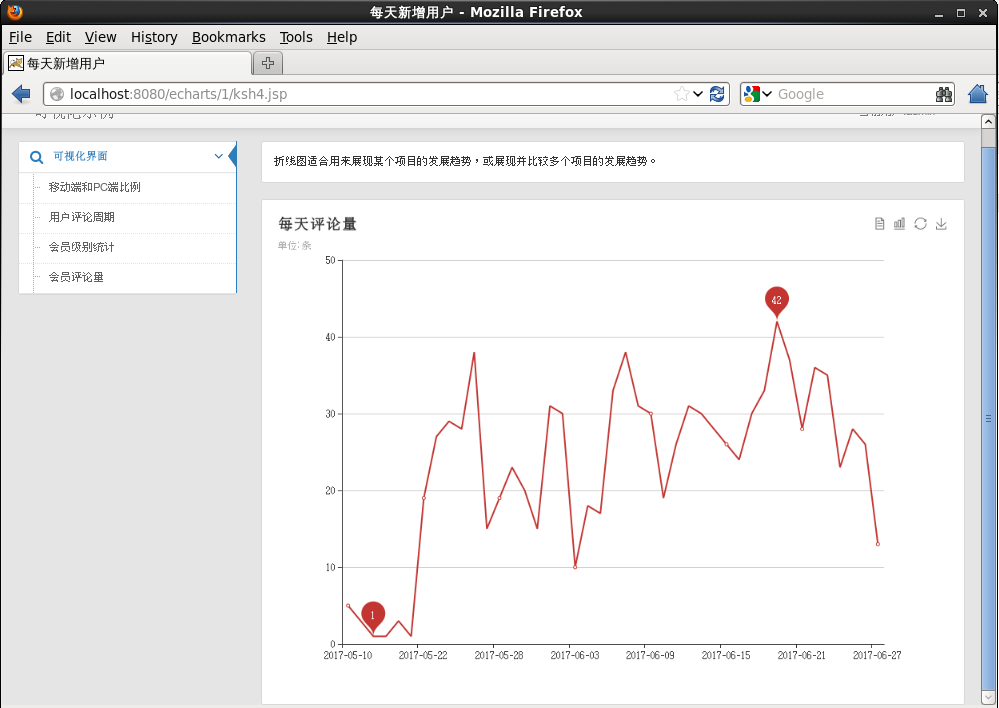


4.13 使用mysql查看creationtime中的数据：

|  |
| --- |
| select \* from creationtimesql; |

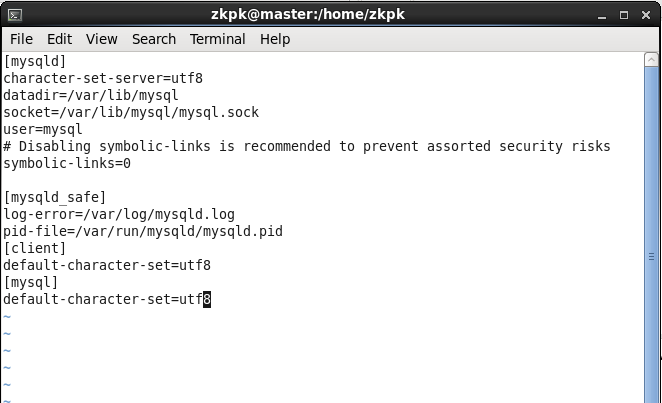


4.14 在浏览器中输入<http://localhost:8080/echarts/1/ksh4.jsp>查看每天的评论量。



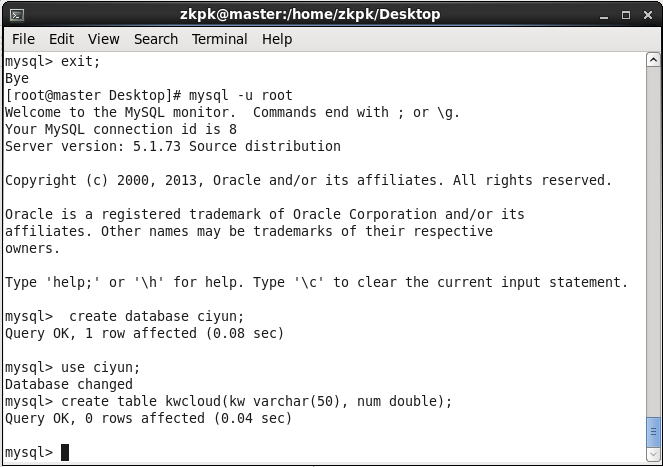
1. **制作词云**

5.1配置字符编码，将数据库中的编码改为UFTF-8。

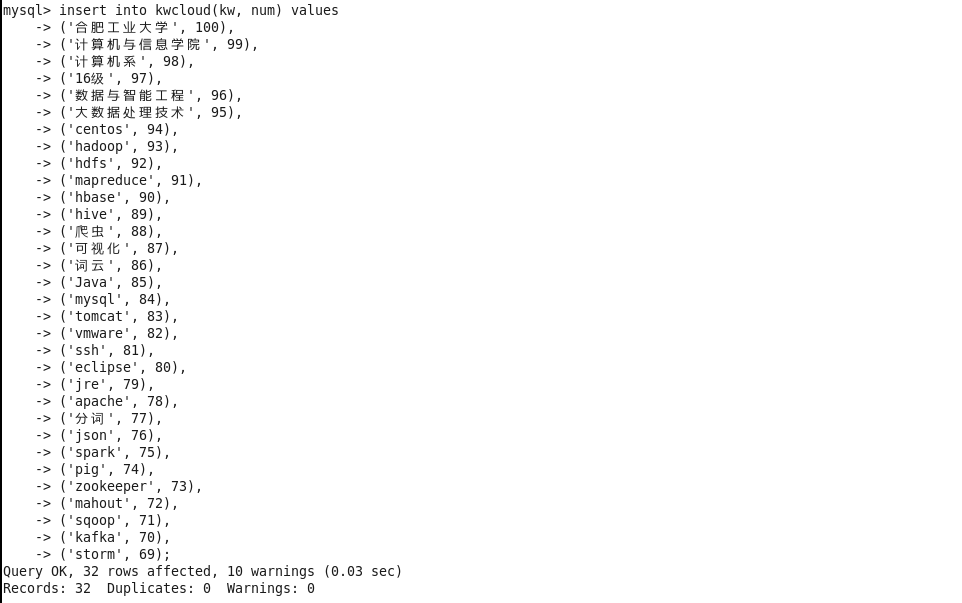


5.2重启mysql并建立数据库以及表：（注意在建立数据库的时候也要指定编码格式，不然在实验过程中仍然出错）。

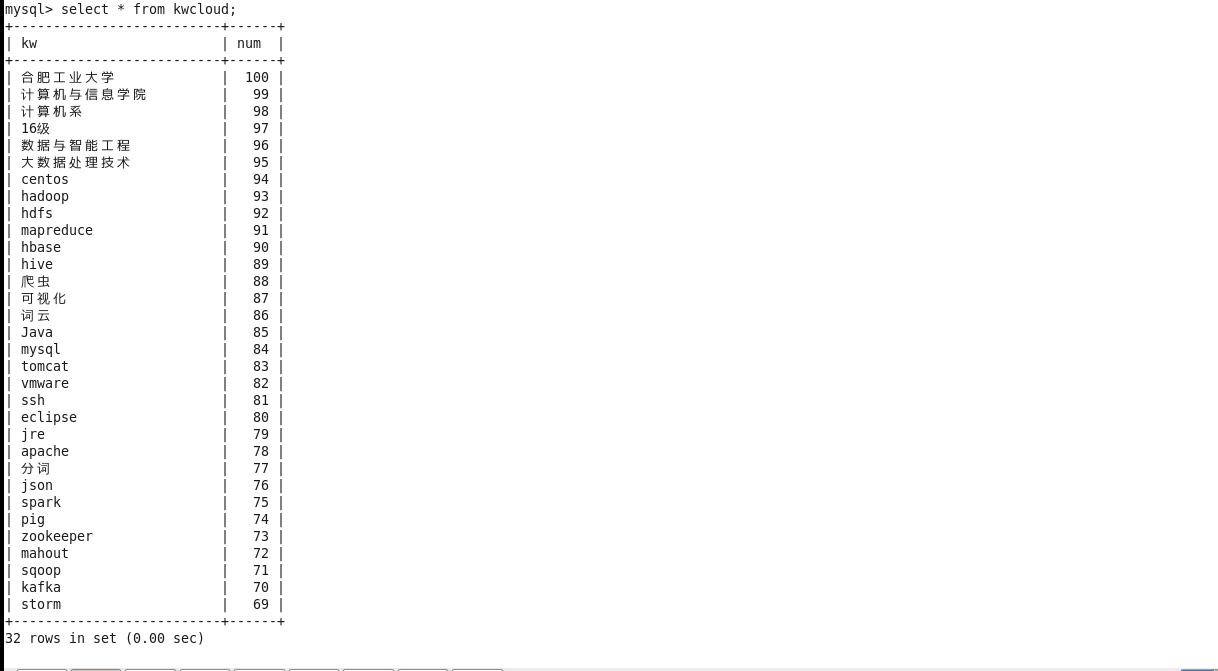
|  |
| --- |
| create database ciyun default character set utf8 collate utf8\_general\_ci; |



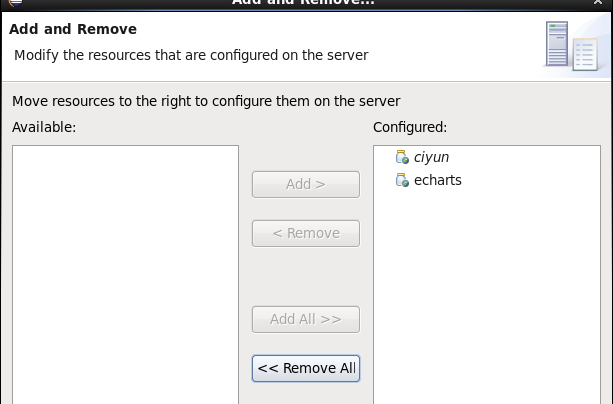
5.3 向kwcloud表中插入如下数据：



5.4 使用查询语句检查是否有乱码数据：



5.5 将词云项目加到configured中，使其在server中运行：



5.6 在浏览器中输入<http://localhost:8080/ciyun/1/ksh1.jsp>查看词云展示。



1. **感想、体会、建议**

在本次实验中，我们使用了几种分词工具，比较了这几种分词工具的好坏，使用了mysql，tomcat server，也使用词云，使用了简易的爬虫工具，生成了网页。在这个过程中学到了很多东西，其中有些也是在后面的综合设计中会使用到的，例如可视化工具。在这个实验中也涉及到了一些以前没有接触过的东西，动手解决了一些没有接触到的方向中的问题，例如在导入echart包的时候，配置jre System Library但是没有教程中的解决方法，最后得出的解决方法为先配置Tomcat Server，问题迎刃而解，这种情况也在其他地方出现过，最终凭借详细的实验指导得到解决。通过本课程/实验，我相信我具备了学习一种新的大数据技术的基本能力，达到了本课程/实验的目的。通过有意思的实验寓教于乐是我的一个很大的感受。

1. **实验成绩**

**指导教师签名：**

年 月 日