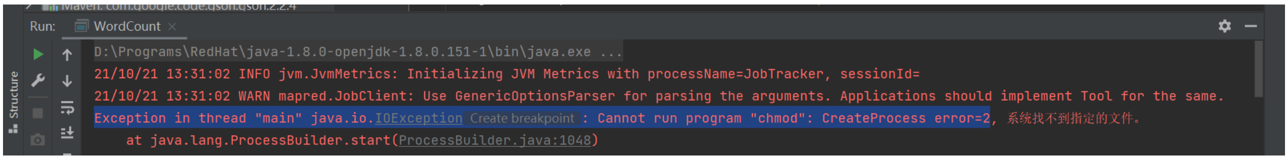
作业5 Hadoop编程

朱彤轩 191840376

1. 配置Intellij以本地运行和调试MapReduce程序

先安装Intellij Community版本与JDK，然后按照CSDN教程Hadoop: Intellij结合Maven本地运行和调试MapReduce程序 (无需搭载Hadoop和HDFS环境)进行配置，我将详细叙述我遇到的bug与解决方案。



```
D:\Programs\RedHat\java-1.8.0-openjdk-1.8.0.151-1\bin\java.exe ...
21/10/21 13:31:02 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker, sessionId=
21/10/21 13:31:02 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the same.
Exception in thread "main" java.io.IOException Create breakpoint : Cannot run program "chmod": CreateProcess error=2, 系统找不到指定的文件。
    at java.lang.ProcessBuilder.start(ProcessBuilder.java:1048)
```

解决方法： 按照教程Cygwin安装教程安装cygwin，把cygwin的bin目录加到windows的用户环境变量中然后重启电脑。

2. 代码编写思路

本作业代码在Hadoop官方示例wordcount2的基础上进行改进，官方示例已经可以读取多个文件、忽略大小写与标点符号。改进如下：

2.1 特殊词的处理

2.1.1 大小写不敏感

去掉原来代码中的判断，直接全部转为小写

2.1.2 去标点符号

复制wordcount2中的`parseSkipFile`名称改为`parseSkipPunctuation`，同时原来代码中是希望将文本中的标点符号替换为空字符，这里改为替换成空格。因为英文中所有格`'s`与前后文是没有空格的。

```
for (String pattern : punctuations) {
    line = line.replaceAll(pattern, " ");
}
```

2.1.3 去停用词

延用wordcount2中的`parseSkipFile`，读取filename路径下的文件，将文件中的需要词加入停用词列表`patternsToSkip`。

在map对句子按空白符分词之后，查看每个单词是否在停用词列表中，如果在则略过(continue)，如果不在，进行进一步处理。

```
if(patternsToSkip.contains(one_word)){
    continue;
}
```

2.1.4 忽略数字与单词长度小于3

用正则表达式判断是否是数字，计算字符串长度筛选复合条件的字符串。

```
//判断是否长度小于3
if(one_word.length()<3) {
    continue;
}
//判断是否是数字，用正则表达式
if(Pattern.compile("^[-\\+]?[\\d]*$").matcher(one_word).matches())
{
    continue;
}
```

2.2 输出每个作品以及所有作品前100个高频单词

我一共用4个job串行实现作业中`输出每个作品以及所有作品前100个高频单词`的要求。在完成“输出每个作品前100高频词”时，我想设置作品个数个reducer，每个reducer负责对该文件统计词频结果进行输出。但是在一个job中是无法对value进行排序的，要想实现纯粹对value的排序，应该先输出中间文件，再读取，之后将value与key倒置，让mpr程序自动帮忙排序。所以最终放弃了设置多个reducer这个想法。

2.2.1 第一个job-word count

原CLASS	继承CLASS
Mapper	TokenizerFileMapper
Combiner	IntSumReducer
Reducer	IntSumReducer

TokenizerFileMapper打开输入输入参数下文件夹，一个mapper负责一个文件。每次取一行进行分词，然后经过`2.1`节的判断，复合规范的加上获取的文件名，输出`<key：word#filename，value：1>`。

由于只有一个reducer，具有相同filename的单词会被分配到同一个reducer中，所以没有采用InvertedIndexer中的重写combiner和partitioner。

IntSumReducer对词频进行统计，并输出`<key：word#filename，value：count>`到中间文件夹`tmp-file-word-count`中，这里的词频是单词在一个文件中出现的次数。

2.2.2 第二个job-sort file

原CLASS	继承CLASS
Mapper	InverseMapper
Reducer	SortFileReducer

此job的输入为上一个job的输出，也即中间文件夹`tmp-file-word-count`。

InverseMapper为hadoop自带的mapper，把读取进来的key与value进行倒置。也就是原来`<key: word#filename, value: count>`，倒置之后变成`<key: count, value: word#filename>`。

重写一个IntWritableDecreasingComparator类，按照新的key（频数）进行降序排 序（原本默认是升序）传给reducer。

SortFileReducer实现的功能为输出每个文件的高频100词到名为`filename-r-0000`的文件中去，统一保存在中间文件夹`single-file-output`文件夹中。

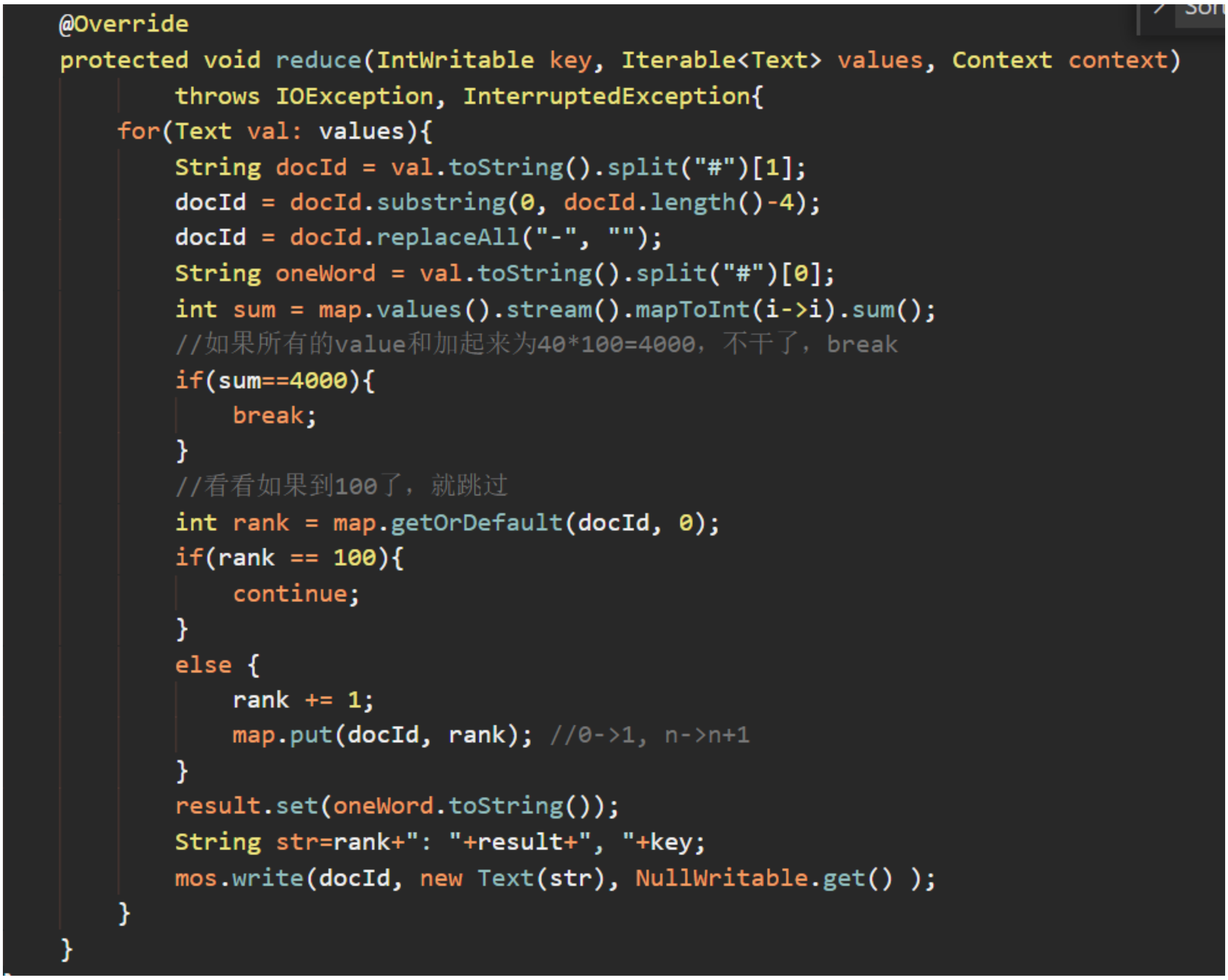
SortFileReducer实现思路：

mapper输出为`<key: count, value: word#filename>`，那么传到reducer看到的就是`<count, [word1#filename1, word2#filename2...]>`。

定义一个Hashmap，key为filename，value为从高频到低频遍历过程中，名为filename的已输出的高频词数。当map[filename]为100，遍历到filename文件时，直接continue；当map的value和为40*100=4000时，break出来，结束reduce。

没遇到一个词，就按照`<rank>: <word>, <count>`的格式输出到名为`filename-r-0000`的文件中去。此处用了hadoop的`MultipleOutputs`。

代码如下：



```java
    @Override
    protected void reduce(IntWritable key, Iterable<Text> values, Context context)
            throws IOException, InterruptedException{
        for(Text val: values){
            String docId = val.toString().split("#")[1];
            docId = docId.substring(0, docId.length()-4);
            docId = docId.replaceAll("-", "");
            String oneWord = val.toString().split("#")[0];
            int sum = map.values().stream().mapToInt(i->i).sum();
            //如果所有的value和加起来为40*100=4000，不干了，break
            if(sum==4000){
                break;
            }
            //看看如果到100了，就跳过
            int rank = map.getOrDefault(docId, 0);
            if(rank == 100){
                continue;
            }
            else {
                rank += 1;
                map.put(docId, rank); //0->1, n->n+1
            }
            result.set(oneWord.toString());
            String str=rank+": "+result+", "+key;
            mos.write(docId, new Text(str), NullWritable.get() );
        }
    }
```

2.2.3 第三个job-all word count

原CLASS	继承CLASS
Mapper	TokenizerFileMapper
Combiner	IntSumReducer
Reducer	IntSumReducer

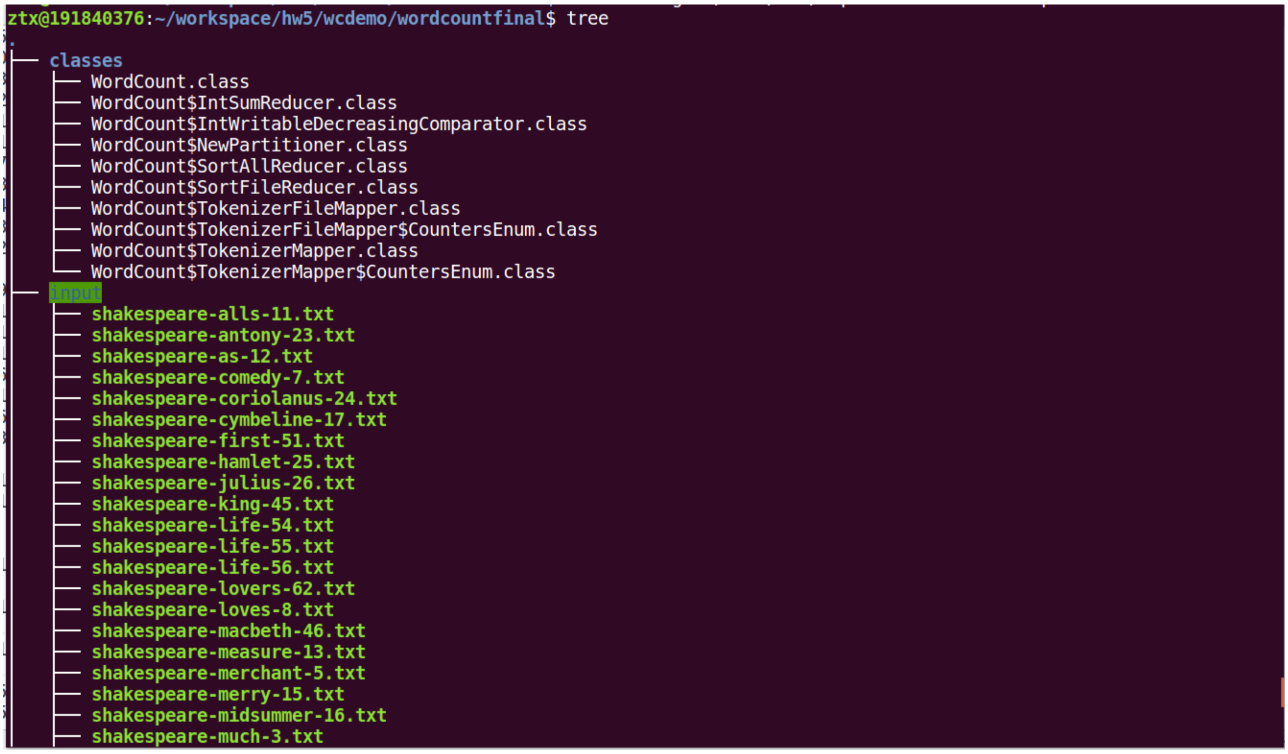
与第一个job类似，唯一不同的是这次mapper输出为`<key: word, value: 1>`。最终输出每个词在所有文件的词频到文件夹`tmp-all-word-count`中。

2.2.4 第四个job-sort all

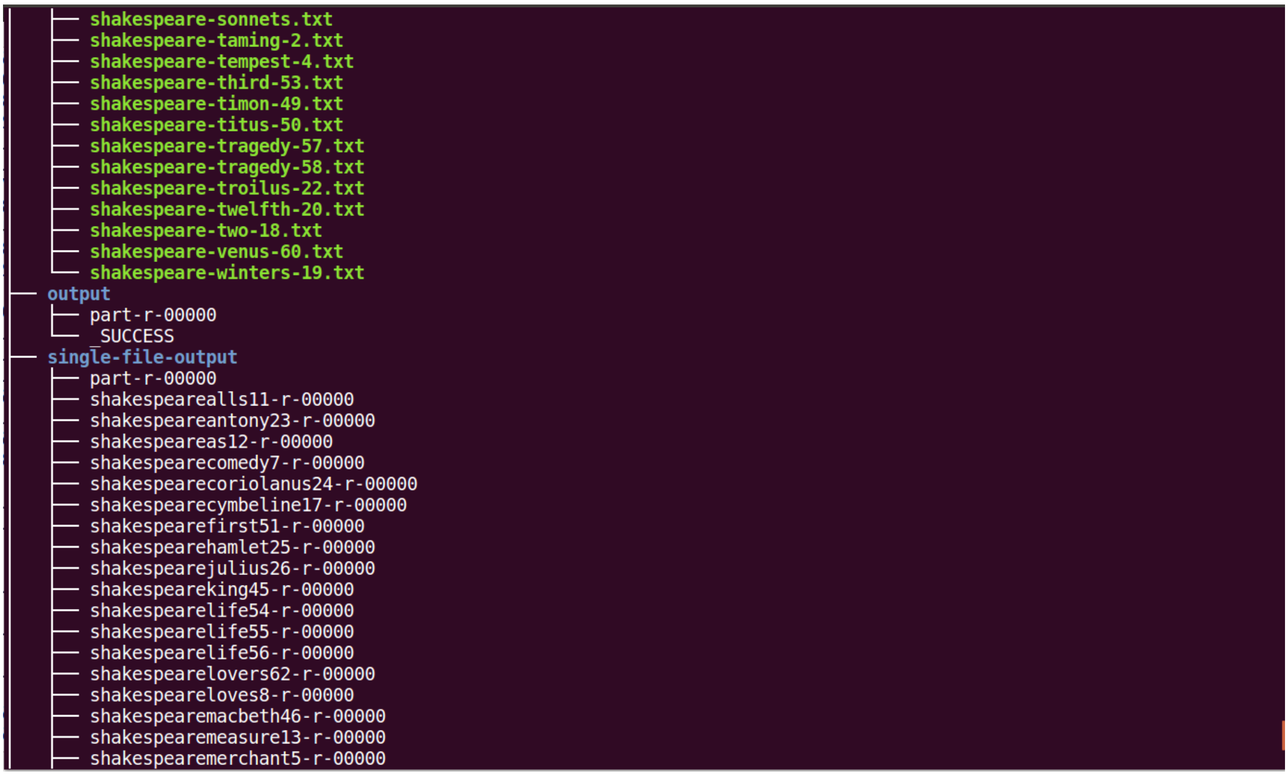
原CLASS	继承CLASS
Mapper	InverseMapper
Reducer	SortAllReducer

前面与第二个job实现一样，在reducer上更为简单，直接按格式输出前100个高频词到output文件夹中即可。新的key为符合输出格式的Text类（字符串拼接），value为NullWritable。

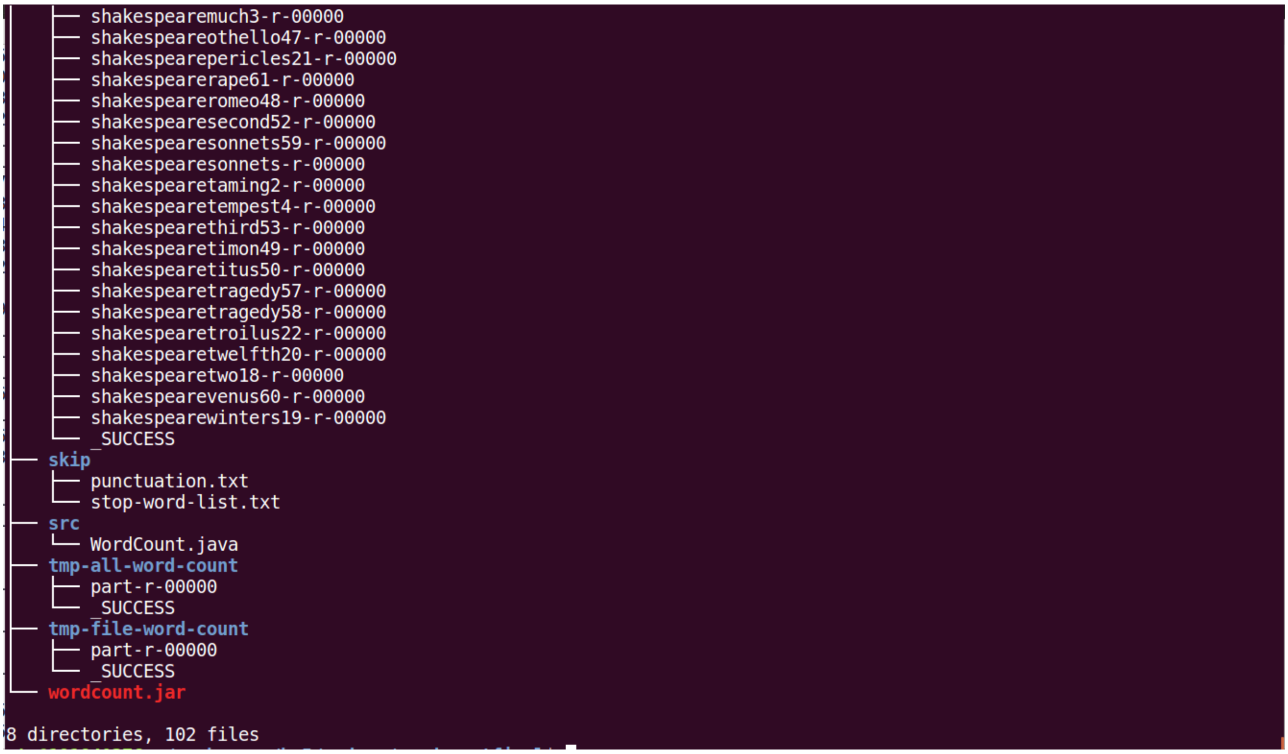
3. 文件夹目录结构



```
ztx@191840376:~/workspace/hw5/wcdemo/wordcountfinal$ tree
.
├── classes
│   ├── WordCount.class
│   ├── WordCount$IntSumReducer.class
│   ├── WordCount$IntWritableDecreasingComparator.class
│   ├── WordCount$NewPartitioner.class
│   ├── WordCount$SortAllReducer.class
│   ├── WordCount$SortFileReducer.class
│   ├── WordCount$TokenizerFileMapper.class
│   ├── WordCount$TokenizerFileMapper$CountersEnum.class
│   ├── WordCount$TokenizerMapper.class
│   └── WordCount$TokenizerMapper$CountersEnum.class
├── input
│   ├── shakespeare-alls-11.txt
│   ├── shakespeare-antony-23.txt
│   ├── shakespeare-as-12.txt
│   ├── shakespeare-comedy-7.txt
│   ├── shakespeare-coriolanus-24.txt
│   ├── shakespeare-cymbeline-17.txt
│   ├── shakespeare-first-51.txt
│   ├── shakespeare-hamlet-25.txt
│   ├── shakespeare-julius-26.txt
│   ├── shakespeare-king-45.txt
│   ├── shakespeare-life-54.txt
│   ├── shakespeare-life-55.txt
│   ├── shakespeare-life-56.txt
│   ├── shakespeare-lovers-62.txt
│   ├── shakespeare-loves-8.txt
│   ├── shakespeare-macbeth-46.txt
│   ├── shakespeare-measure-13.txt
│   ├── shakespeare-merchant-5.txt
│   ├── shakespeare-merry-15.txt
│   ├── shakespeare-midsummer-16.txt
│   ├── shakespeare-much-3.txt
```



```
│   ├── shakespeare-sonnets.txt
│   ├── shakespeare-taming-2.txt
│   ├── shakespeare-tempest-4.txt
│   ├── shakespeare-third-53.txt
│   ├── shakespeare-timon-49.txt
│   ├── shakespeare-titus-50.txt
│   ├── shakespeare-tragedy-57.txt
│   ├── shakespeare-tragedy-58.txt
│   ├── shakespeare-troilus-22.txt
│   ├── shakespeare-twelfth-20.txt
│   ├── shakespeare-two-18.txt
│   ├── shakespeare-venus-60.txt
│   └── shakespeare-winters-19.txt
├── output
│   ├── part-r-00000
│   └── _SUCCESS
├── single-file-output
│   ├── part-r-00000
│   ├── shakespearealls11-r-00000
│   ├── shakespeareantony23-r-00000
│   ├── shakespeareas12-r-00000
│   ├── shakespearecomedy7-r-00000
│   ├── shakespearecoriolanus24-r-00000
│   ├── shakespearecymbeline17-r-00000
│   ├── shakespearefirst51-r-00000
│   ├── shakespearehamlet25-r-00000
│   ├── shakespearejulius26-r-00000
│   ├── shakespeareking45-r-00000
│   ├── shakespearelife54-r-00000
│   ├── shakespearelife55-r-00000
│   ├── shakespearelife56-r-00000
│   ├── shakespearelovers62-r-00000
│   ├── shakespeareloves8-r-00000
│   ├── shakespearemacbeth46-r-00000
│   ├── shakespearemeasure13-r-00000
│   ├── shakespearemerchant5-r-00000
```



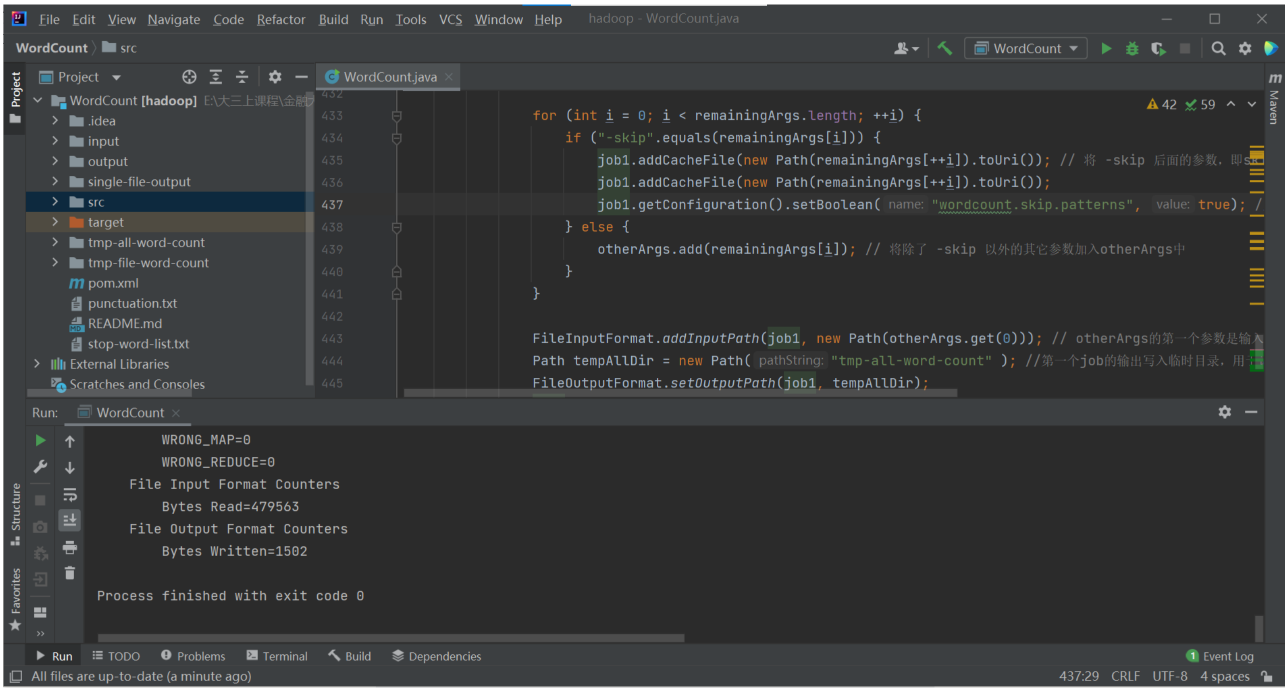
```
│   ├── shakespearemuch3-r-00000
│   ├── shakespeareothello47-r-00000
│   ├── shakespearepericles21-r-00000
│   ├── shakespearerape61-r-00000
│   ├── shakespeareromeo48-r-00000
│   ├── shakespearesecond52-r-00000
│   ├── shakespearesonnets59-r-00000
│   ├── shakespearesonnets-r-00000
│   ├── shakespearetaming2-r-00000
│   ├── shakespearetempest4-r-00000
│   ├── shakespearethird53-r-00000
│   ├── shakespearetimon49-r-00000
│   ├── shakespearetitus50-r-00000
│   ├── shakespearetragedy57-r-00000
│   ├── shakespearetragedy58-r-00000
│   ├── shakespearetroilus22-r-00000
│   ├── shakespearetwelfth20-r-00000
│   ├── shakespearetwo18-r-00000
│   ├── shakespearevenus60-r-00000
│   ├── shakespearewinters19-r-00000
│   └── _SUCCESS
├── skip
│   ├── punctuation.txt
│   └── stop-word-list.txt
├── src
│   └── WordCount.java
├── tmp-all-word-count
│   ├── part-r-00000
│   └── _SUCCESS
├── tmp-file-word-count
│   ├── part-r-00000
│   └── _SUCCESS
└── wordcount.jar

8 directories, 102 files
```

- classes：各个类
- input：40个莎士比亚作品txt
- output：所有文件的高频100词文件
- single-file-output：每个文件的高频100词文件
- skip：标点以及停用词存放
- src：源码
- tmp-all-word-count：中间文件之每个单词在所有文件中词频
- tmp-file-word-count：中间文件之每个单词在单个文件中词频

4. 实验结果

4.1 windows系统下运行截图



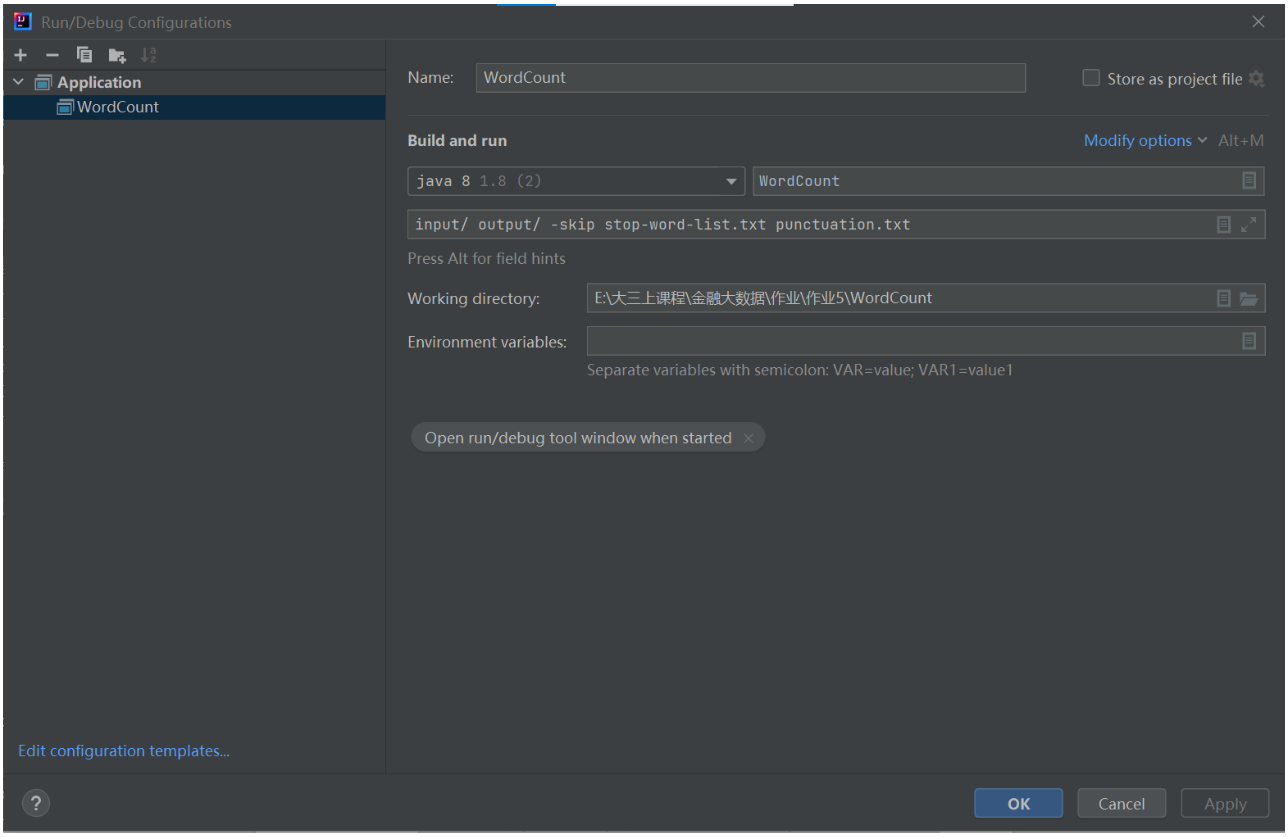
```
        for (int i = 0; i < remainingArgs.length; ++i) {
            if ("-skip".equals(remainingArgs[i])) {
                job1.addCacheFile(new Path(remainingArgs[++i]).toUri()); // 将 -skip 后面的参数, 即sk
                job1.addCacheFile(new Path(remainingArgs[++i]).toUri());
                job1.getConfiguration().setBoolean( name: "wordcount.skip.patterns",  value: true); /
            } else {
                otherArgs.add(remainingArgs[i]); // 将除了 -skip 以外的其它参数加入otherArgs中
            }
        }

        FileInputFormat.addInputPath(job1, new Path(otherArgs.get(0))); // otherArgs的第一个参数是输入
        Path tempAllDir = new Path( pathString: "tmp-all-word-count" ); //第一个job的输出写入临时目录, 用
        FileOutputFormat.setOutputPath(job1, tempAllDir);
```

```
		WRONG_MAP=0
		WRONG_REDUCE=0
	File Input Format Counters 
		Bytes Read=479563
	File Output Format Counters 
		Bytes Written=1502

Process finished with exit code 0
```

All files are up-to-date (a minute ago)



Run/Debug Configurations

Name: WordCount

□ Store as project file

Build and run

java 8 1.8 (2)

WordCount

```
input/ output/ -skip stop-word-list.txt punctuation.txt
```

Press Alt for field hints

Working directory: E:\大三上课程\金融大数据\作业\作业5\WordCount

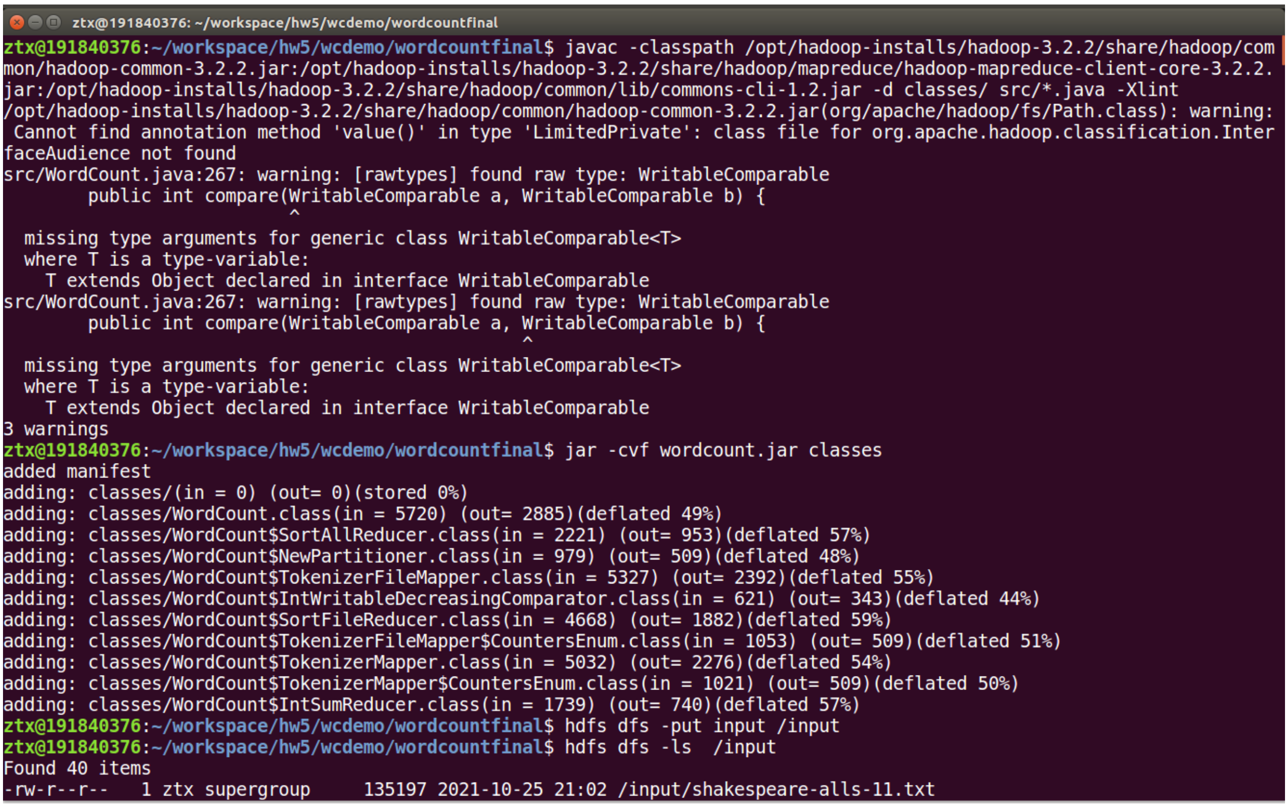
Environment variables:

Separate variables with semicolon: VAR=value; VAR1=value1

Open run/debug tool window when started

4.2 Linux系统运行

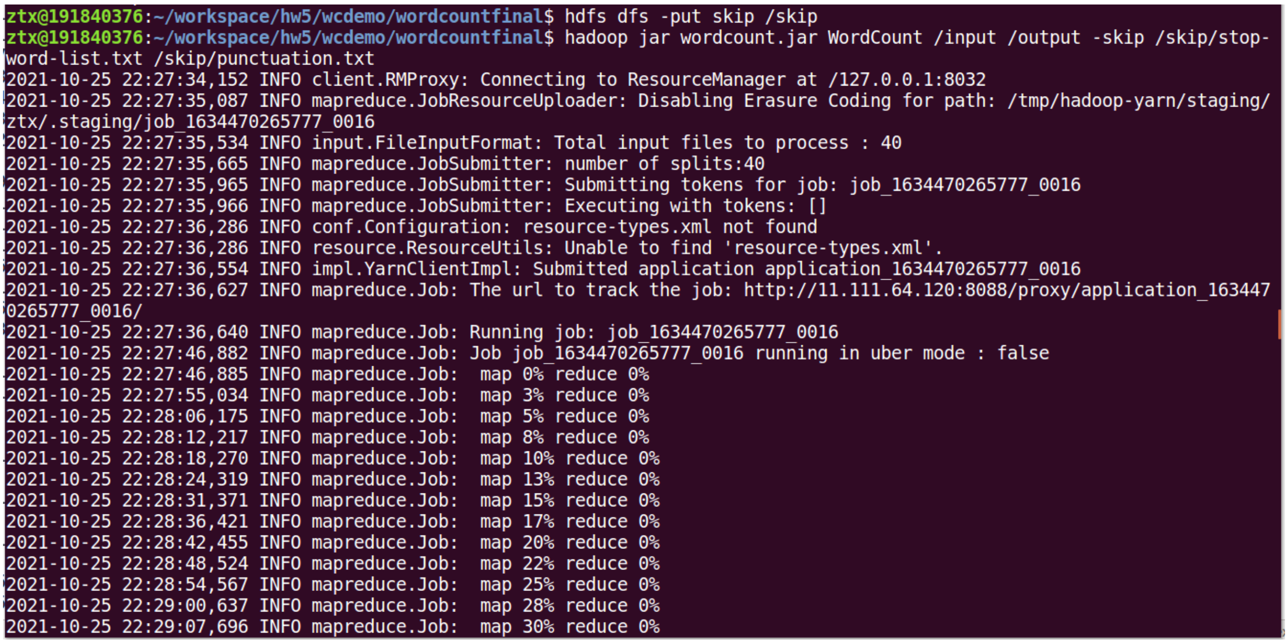
代码编译：



```
ztx@191840376:~/workspace/hw5/wcdemo/wordcountfinal$ javac -classpath /opt/hadoop-installs/hadoop-3.2.2/share/hadoop/com
mon/hadoop-common-3.2.2.jar:/opt/hadoop-installs/hadoop-3.2.2/share/hadoop/mapreduce/hadoop-mapreduce-client-core-3.2.2.
jar:/opt/hadoop-installs/hadoop-3.2.2/share/hadoop/common/lib/commons-cli-1.2.jar -d classes/ src/*.java -Xlint
/opt/hadoop-installs/hadoop-3.2.2/share/hadoop/common/hadoop-common-3.2.2.jar(org/apache/hadoop/fs/Path.class): warning:
 Cannot find annotation method 'value()' in type 'LimitedPrivate': class file for org.apache.hadoop.classification.Inter
faceAudience not found
src/WordCount.java:267: warning: [rawtypes] found raw type: WritableComparable
        public int compare(WritableComparable a, WritableComparable b) {
                           ^
  missing type arguments for generic class WritableComparable<T>
  where T is a type-variable:
    T extends Object declared in interface WritableComparable
src/WordCount.java:267: warning: [rawtypes] found raw type: WritableComparable
        public int compare(WritableComparable a, WritableComparable b) {
                                                 ^
  missing type arguments for generic class WritableComparable<T>
  where T is a type-variable:
    T extends Object declared in interface WritableComparable
3 warnings
ztx@191840376:~/workspace/hw5/wcdemo/wordcountfinal$ jar -cvf wordcount.jar classes
added manifest
adding: classes/(in = 0) (out= 0)(stored 0%)
adding: classes/WordCount.class(in = 5720) (out= 2885)(deflated 49%)
adding: classes/WordCount$SortAllReducer.class(in = 2221) (out= 953)(deflated 57%)
adding: classes/WordCount$NewPartitioner.class(in = 979) (out= 509)(deflated 48%)
adding: classes/WordCount$TokenizerFileMapper.class(in = 5327) (out= 2392)(deflated 55%)
adding: classes/WordCount$IntWritableDecreasingComparator.class(in = 621) (out= 343)(deflated 44%)
adding: classes/WordCount$SortFileReducer.class(in = 4668) (out= 1882)(deflated 59%)
adding: classes/WordCount$TokenizerFileMapper$CountersEnum.class(in = 1053) (out= 509)(deflated 51%)
adding: classes/WordCount$TokenizerMapper.class(in = 5032) (out= 2276)(deflated 54%)
adding: classes/WordCount$TokenizerMapper$CountersEnum.class(in = 1021) (out= 509)(deflated 50%)
adding: classes/WordCount$IntSumReducer.class(in = 1739) (out= 740)(deflated 57%)
ztx@191840376:~/workspace/hw5/wcdemo/wordcountfinal$ hdfs dfs -put input /input
ztx@191840376:~/workspace/hw5/wcdemo/wordcountfinal$ hdfs dfs -ls  /input
Found 40 items
-rw-r--r--   1 ztx supergroup     135197 2021-10-25 21:02 /input/shakespeare-alls-11.txt
```

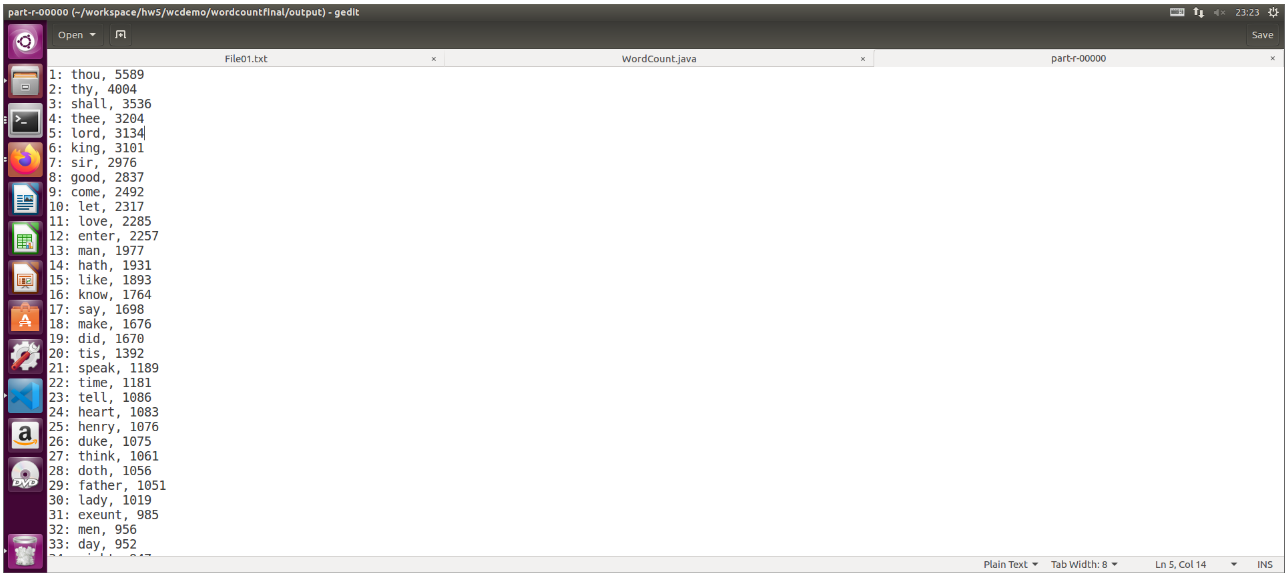
在终端运行代码：

```
ztx@191840376:~/workspace/hw5/wcdemo/wordcountfinal$ hadoop jar
wordcount.jar /input /output -skip /skip/stop-word-list.txt
/skip/punctuation.txt
```



```
ztx@191840376:~/workspace/hw5/wcdemo/wordcountfinal$ hdfs dfs -put skip /skip
ztx@191840376:~/workspace/hw5/wcdemo/wordcountfinal$ hadoop jar wordcount.jar WordCount /input /output -skip /skip/stop-
word-list.txt /skip/punctuation.txt
2021-10-25 22:27:34,152 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0.1:8032
2021-10-25 22:27:35,087 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/
ztx/.staging/job_1634470265777_0016
2021-10-25 22:27:35,534 INFO input.FileInputFormat: Total input files to process : 40
2021-10-25 22:27:35,665 INFO mapreduce.JobSubmitter: number of splits:40
2021-10-25 22:27:35,965 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1634470265777_0016
2021-10-25 22:27:35,966 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-10-25 22:27:36,286 INFO conf.Configuration: resource-types.xml not found
2021-10-25 22:27:36,286 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-10-25 22:27:36,554 INFO impl.YarnClientImpl: Submitted application application_1634470265777_0016
2021-10-25 22:27:36,627 INFO mapreduce.Job: The url to track the job: http://11.111.64.120:8088/proxy/application_163447
0265777_0016/
2021-10-25 22:27:36,640 INFO mapreduce.Job: Running job: job_1634470265777_0016
2021-10-25 22:27:46,882 INFO mapreduce.Job: Job job_1634470265777_0016 running in uber mode : false
2021-10-25 22:27:46,885 INFO mapreduce.Job:  map 0% reduce 0%
2021-10-25 22:27:55,034 INFO mapreduce.Job:  map 3% reduce 0%
2021-10-25 22:28:06,175 INFO mapreduce.Job:  map 5% reduce 0%
2021-10-25 22:28:12,217 INFO mapreduce.Job:  map 8% reduce 0%
2021-10-25 22:28:18,270 INFO mapreduce.Job:  map 10% reduce 0%
2021-10-25 22:28:24,319 INFO mapreduce.Job:  map 13% reduce 0%
2021-10-25 22:28:31,371 INFO mapreduce.Job:  map 15% reduce 0%
2021-10-25 22:28:36,421 INFO mapreduce.Job:  map 17% reduce 0%
2021-10-25 22:28:42,455 INFO mapreduce.Job:  map 20% reduce 0%
2021-10-25 22:28:48,524 INFO mapreduce.Job:  map 22% reduce 0%
2021-10-25 22:28:54,567 INFO mapreduce.Job:  map 25% reduce 0%
2021-10-25 22:29:00,637 INFO mapreduce.Job:  map 28% reduce 0%
2021-10-25 22:29:07,696 INFO mapreduce.Job:  map 30% reduce 0%
```

所有文件高频100词：



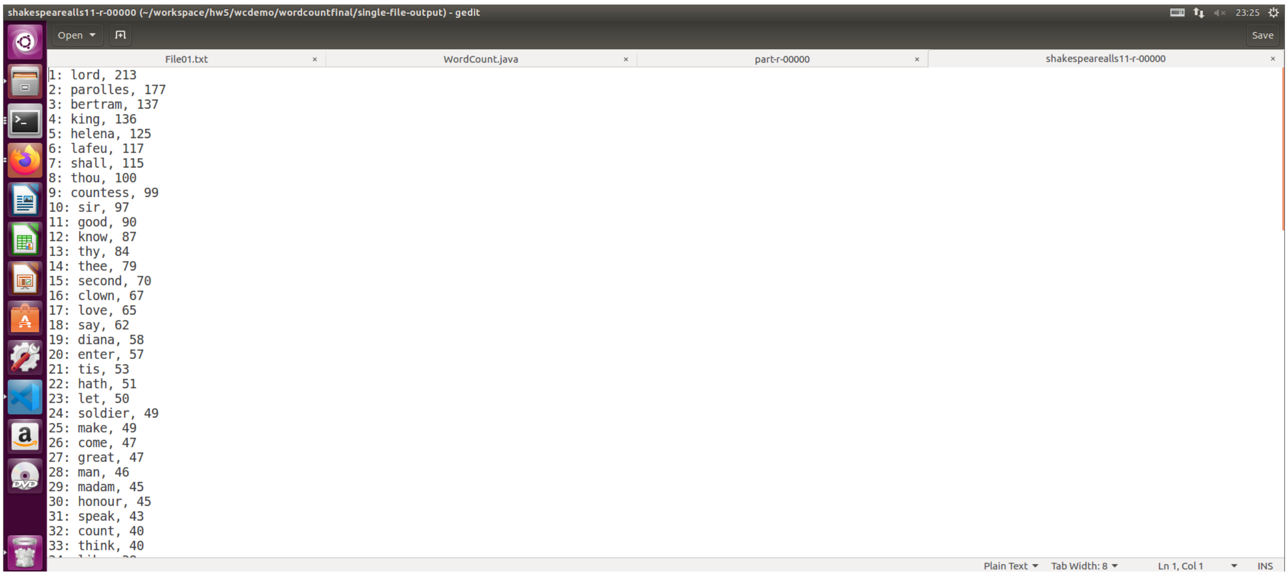
part-r-00000 (~/workspace/hw5/wcdemo/wordcountfinal/output) - gedit

File01.txt | WordCount.java | part-r-00000

```
1: thou, 5589
2: thy, 4004
3: shall, 3536
4: thee, 3204
5: lord, 3134
6: king, 3101
7: sir, 2976
8: good, 2837
9: come, 2492
10: let, 2317
11: love, 2285
12: enter, 2257
13: man, 1977
14: hath, 1931
15: like, 1893
16: know, 1764
17: say, 1698
18: make, 1676
19: did, 1670
20: tis, 1392
21: speak, 1189
22: time, 1181
23: tell, 1086
24: heart, 1083
25: henry, 1076
26: duke, 1075
27: think, 1061
28: doth, 1056
29: father, 1051
30: lady, 1019
31: exeunt, 985
32: men, 956
33: day, 952
```

Plain Text ▾ Tab Width: 8 ▾ Ln 5, Col 14 INS

每个作品高频100词（选`shakespeareal1s11-r-00000`展示）



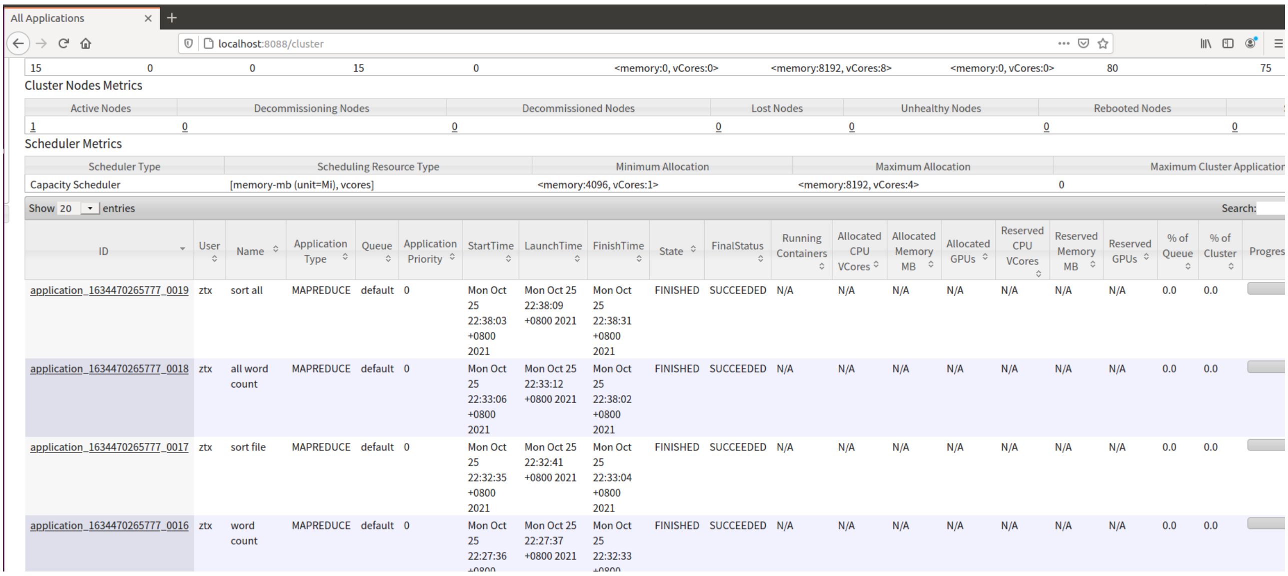
shakespeareall s11-r-00000 (~/workspace/hw5/wcdemo/wordcountfinal/single-file-output) - gedit

File01.txt | WordCount.java | part-r-00000 | shakespeareall s11-r-00000

```
1: lord, 213
2: parolles, 177
3: bertram, 137
4: king, 136
5: helena, 125
6: lafeu, 117
7: shall, 115
8: thou, 100
9: countess, 99
10: sir, 97
11: good, 90
12: know, 87
13: thy, 84
14: thee, 79
15: second, 70
16: clown, 67
17: love, 65
18: say, 62
19: diana, 58
20: enter, 57
21: tis, 53
22: hath, 51
23: let, 50
24: soldier, 49
25: make, 49
26: come, 47
27: great, 47
28: man, 46
29: madam, 45
30: honour, 45
31: speak, 43
32: count, 40
33: think, 40
```

Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 1 INS

All Applications截图：



All Applications

localhost:8088/cluster

15	0	0	15	0	<memory:0, vCores:0>	<memory:8192, vCores:8>	<memory:0, vCores:0>	80	75

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
1	0	0	0	0	0

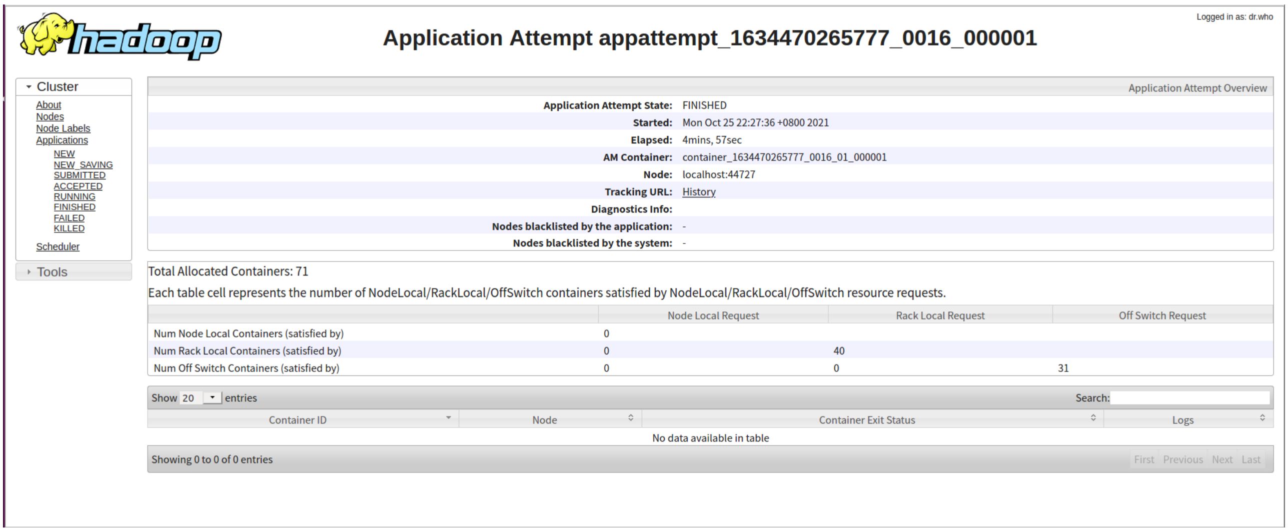
Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Applicatio
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:4096, vCores:1>	<memory:8192, vCores:4>	0

Show 20 entries Search:

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Allocated GPUs	Reserved CPU VCores	Reserved Memory MB	Reserved GPUs	% of Queue	% of Cluster	Progres
application_1634470265777_0019	ztx	sort all	MAPREDUCE	default	0	Mon Oct 25 22:38:03 +0800 2021	Mon Oct 25 22:38:09 +0800 2021	Mon Oct 25 22:38:31 +0800 2021	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.0	0.0	
application_1634470265777_0018	ztx	all word count	MAPREDUCE	default	0	Mon Oct 25 22:33:06 +0800 2021	Mon Oct 25 22:33:12 +0800 2021	Mon Oct 25 22:38:02 +0800 2021	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.0	0.0	
application_1634470265777_0017	ztx	sort file	MAPREDUCE	default	0	Mon Oct 25 22:32:35 +0800 2021	Mon Oct 25 22:32:41 +0800 2021	Mon Oct 25 22:33:04 +0800 2021	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.0	0.0	
application_1634470265777_0016	ztx	word count	MAPREDUCE	default	0	Mon Oct 25 22:27:36 +0800	Mon Oct 25 22:27:37 +0800 2021	Mon Oct 25 22:32:33 +0800	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.0	0.0	

job-word count：



Application Attempt appattempt_1634470265777_0016_000001

Application Attempt Overview

Application Attempt State:	FINISHED
Started:	Mon Oct 25 22:27:36 +0800 2021
Elapsed:	4mins, 57sec
AM Container:	container_1634470265777_0016_01_000001
Node:	localhost:44727
Tracking URL:	History
Diagnostics Info:	
Nodes blacklisted by the application:	-
Nodes blacklisted by the system:	-

Total Allocated Containers: 71

Each table cell represents the number of NodeLocal/RackLocal/OffSwitch containers satisfied by NodeLocal/RackLocal/OffSwitch resource requests.

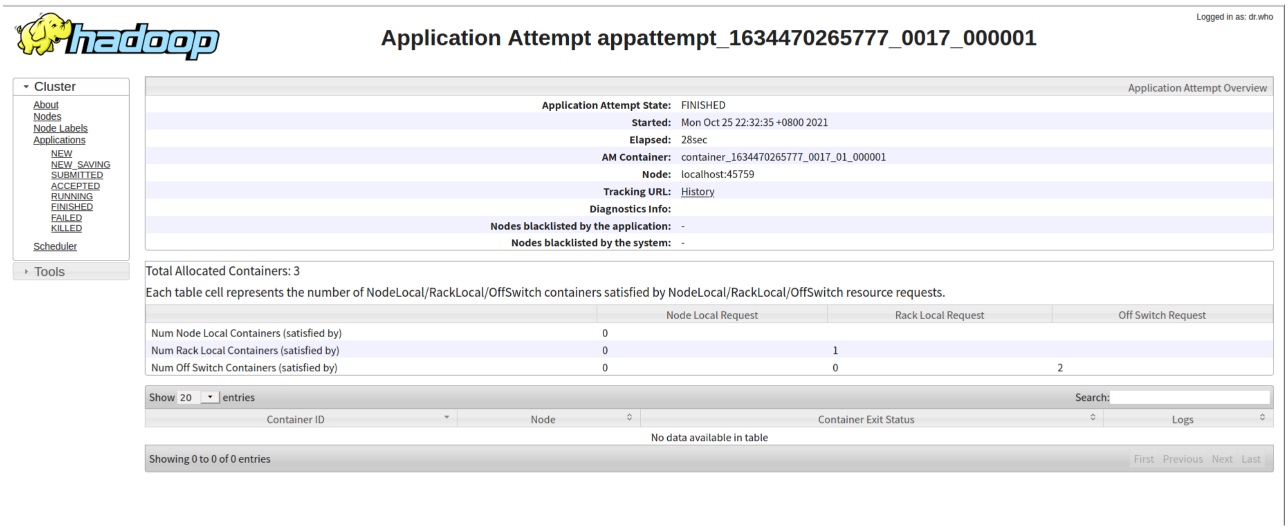
	Node Local Request	Rack Local Request	Off Switch Request
Num Node Local Containers (satisfied by)	0		
Num Rack Local Containers (satisfied by)	0	40	
Num Off Switch Containers (satisfied by)	0	0	31

Show 20 entries Search:

Container ID	Node	Container Exit Status	Logs
No data available in table			

Showing 0 to 0 of 0 entries First Previous Next Last

job-sort file：



Application Attempt appattempt_1634470265777_0017_000001

Application Attempt Overview

Application Attempt State:	FINISHED
Started:	Mon Oct 25 22:32:35 +0800 2021
Elapsed:	28sec
AM Container:	container_1634470265777_0017_01_000001
Node:	localhost:45759
Tracking URL:	History
Diagnostics Info:	
Nodes blacklisted by the application:	-
Nodes blacklisted by the system:	-

Total Allocated Containers: 3

Each table cell represents the number of NodeLocal/RackLocal/OffSwitch containers satisfied by NodeLocal/RackLocal/OffSwitch resource requests.

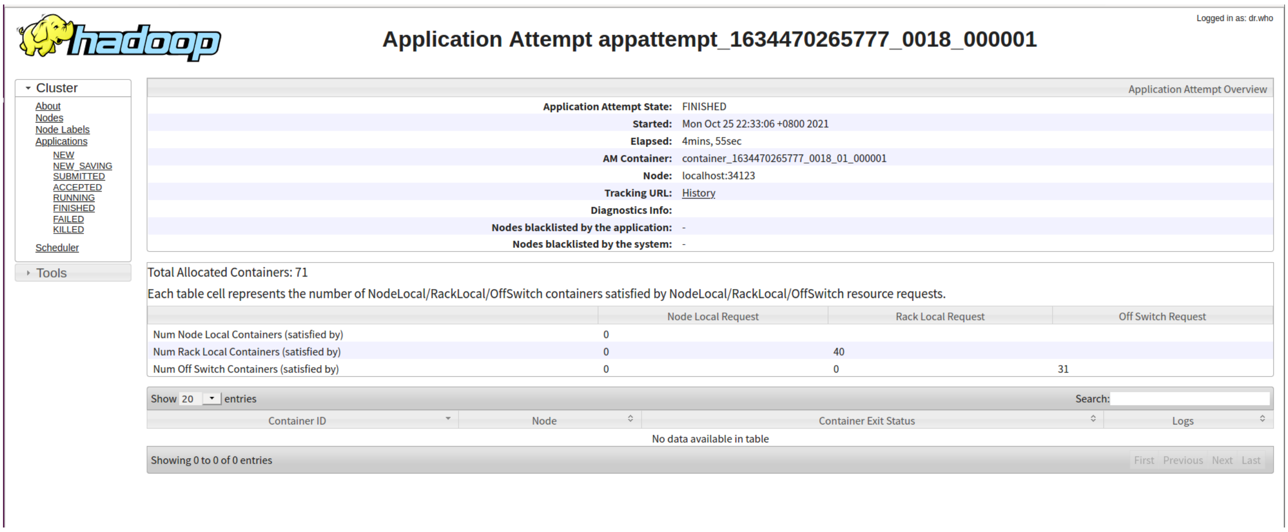
	Node Local Request	Rack Local Request	Off Switch Request
Num Node Local Containers (satisfied by)	0		
Num Rack Local Containers (satisfied by)	0	1	
Num Off Switch Containers (satisfied by)	0	0	2

Show 20 entries Search:

Container ID	Node	Container Exit Status	Logs
No data available in table			

Showing 0 to 0 of 0 entries First Previous Next Last

job-all word count：



Application Attempt appattempt_1634470265777_0018_000001

Application Attempt Overview

Application Attempt State:	FINISHED
Started:	Mon Oct 25 22:33:06 +0800 2021
Elapsed:	4mins, 55sec
AM Container:	container_1634470265777_0018_01_000001
Node:	localhost:34123
Tracking URL:	History
Diagnostics Info:	
Nodes blacklisted by the application:	-
Nodes blacklisted by the system:	-

Total Allocated Containers: 71

Each table cell represents the number of NodeLocal/RackLocal/OffSwitch containers satisfied by NodeLocal/RackLocal/OffSwitch resource requests.

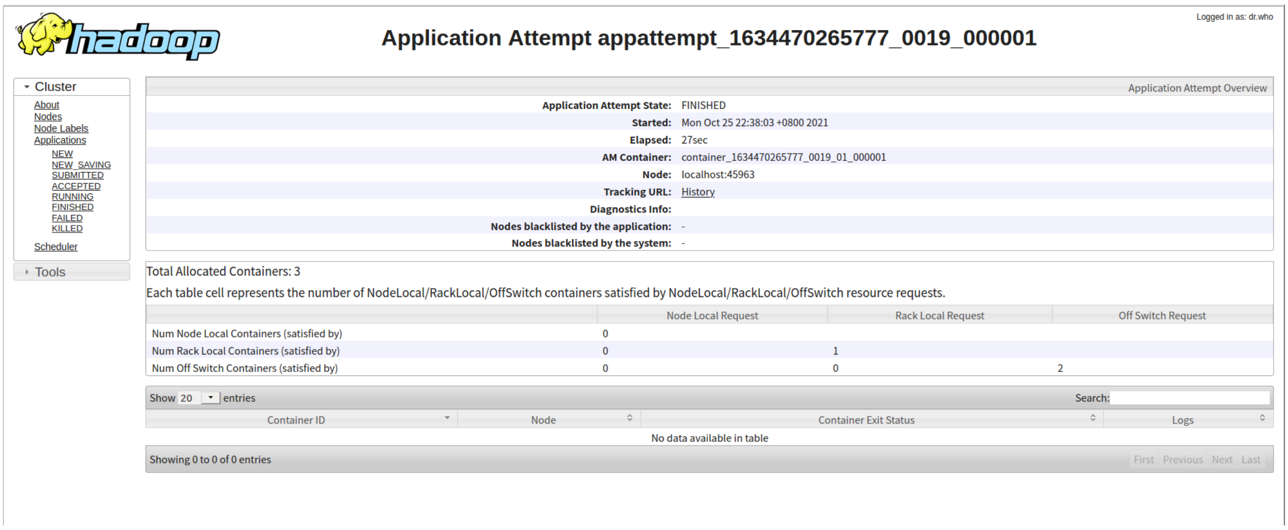
	Node Local Request	Rack Local Request	Off Switch Request
Num Node Local Containers (satisfied by)	0		
Num Rack Local Containers (satisfied by)	0	40	
Num Off Switch Containers (satisfied by)	0	0	31

Show 20 entries Search:

Container ID	Node	Container Exit Status	Logs
No data available in table			

Showing 0 to 0 of 0 entries First Previous Next Last

job-sort all：



Logged in as: dr.who

Application Attempt appattempt_1634470265777_0019_000001

Application Attempt Overview

Application Attempt State:	FINISHED
Started:	Mon Oct 25 22:38:03 +0800 2021
Elapsed:	27sec
AM Container:	container_1634470265777_0019_01_000001
Node:	localhost:45963
Tracking URL:	History
Diagnostics Info:	
Nodes blacklisted by the application:	-
Nodes blacklisted by the system:	-

Total Allocated Containers: 3

Each table cell represents the number of NodeLocal/RackLocal/OffSwitch containers satisfied by NodeLocal/RackLocal/OffSwitch resource requests.

	Node Local Request	Rack Local Request	Off Switch Request
Num Node Local Containers (satisfied by)	0		
Num Rack Local Containers (satisfied by)	0	1	
Num Off Switch Containers (satisfied by)	0	0	2

Show 20 entries Search:

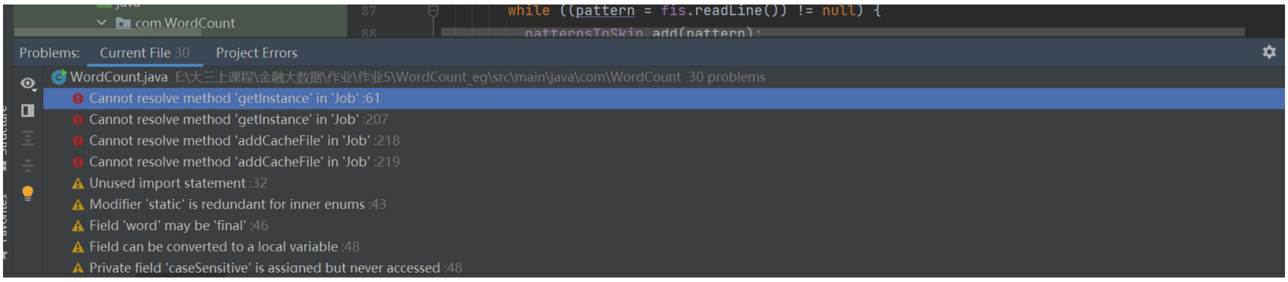
Container ID	Node	Container Exit Status	Logs
No data available in table			

Showing 0 to 0 of 0 entries First Previous Next Last

5. 遇到问题及解决

5.1 无法运行wordcount2

出现了按照教程配置好Intellij之后能运行wordcount但是运行不了wordcount2，显示有一些类无法识别：



```
87    while ((pattern = fis.readLine()) != null) {
88      patternsToSkip.add(pattern);
```

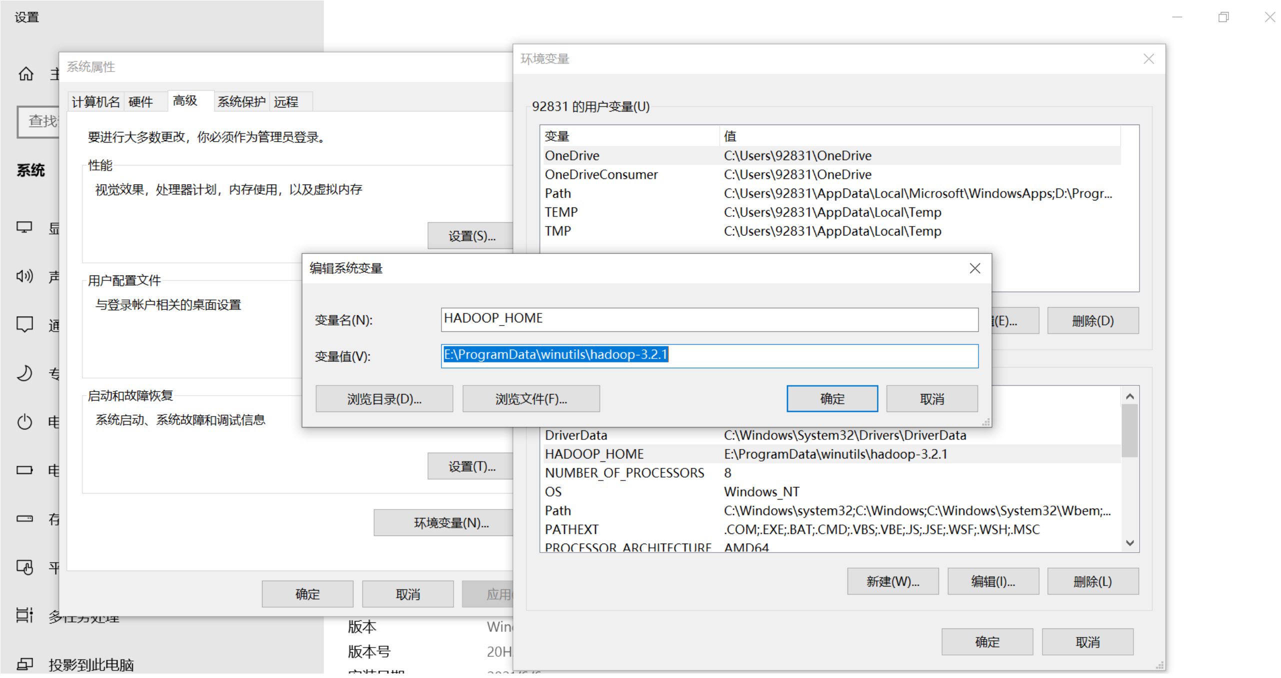
```
WordCount.java E:\大三上课程\金融大数据\作业\作业5\WordCount_eg\src\main\java\com\WordCount 30 problems
  Cannot resolve method 'getInstance' in 'Job' :61
  Cannot resolve method 'getInstance' in 'Job' :207
  Cannot resolve method 'addCacheFile' in 'Job' :218
  Cannot resolve method 'addCacheFile' in 'Job' :219
  Unused import statement :32
  Modifier 'static' is redundant for inner enums :43
  Field 'word' may be 'final' :46
  Field can be converted to a local variable :48
  Private field 'caseSensitive' is assigned but never accessed :48
```

更换了配置文件，具体见文件夹中的`pom.xml`。

5.2 更换配置后找不到HADOOP_HOME

`java.io.FileNotFoundException: HADOOP_HOME and hadoop.home.dir are unset.`

本地远程连接Hadoop系统时需要在本地配置相关的Hadoop变量，主要包括hadoop.dll 与 winutils.exe 等。在GitHub上下载与配置文件中版本相符的**hadoop.dll** 与 **winutils.exe**，设置环境变量，把hadoop.dll文件复制到`C:\Windows\System32`下，最后重启。



设置
系统属性
计算机名 硬件 高级 系统保护 远程
要进行大多数更改，你必须作为管理员登录。
性能
视觉效果，处理器计划，内存使用，以及虚拟内存
设置(S)...
用户配置文件
与登录帐户相关的桌面设置
启动和故障恢复
系统启动、系统故障和调试信息
设置(T)...
环境变量(N)...
确定 取消 应用
环境变量
92831 的用户变量(U)
编辑系统变量
浏览目录(D)... 浏览文件(F)... 确定 取消
新建(W)... 编辑(I)... 删除(L)
删除(D)
确定 取消

变量名(N): HADOOP_HOME

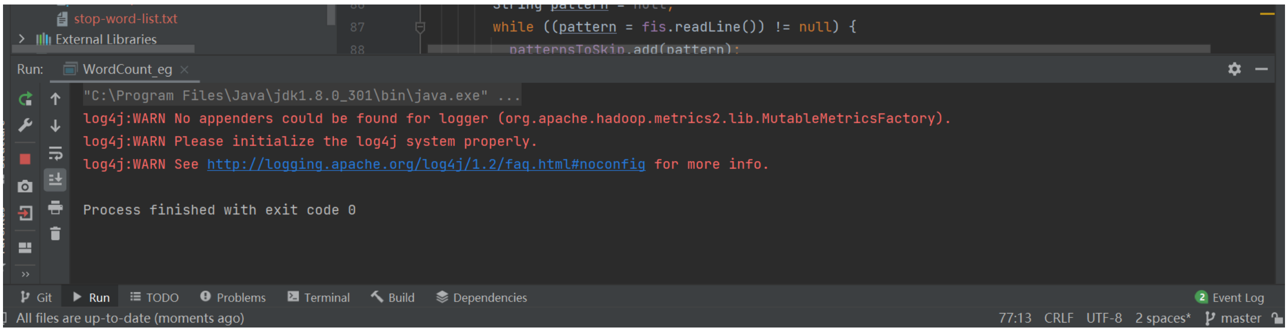
变量值(V): E:\ProgramData\winutils\hadoop-3.2.1

变量	值
OneDrive	C:\Users\92831\OneDrive
OneDriveConsumer	C:\Users\92831\OneDrive
Path	C:\Users\92831\AppData\Local\Microsoft\WindowsApps;D:\Progr...
TEMP	C:\Users\92831\AppData\Local\Temp
TMP	C:\Users\92831\AppData\Local\Temp

DriverData	C:\Windows\System32\Drivers\DriverData
HADOOP_HOME	E:\ProgramData\winutils\hadoop-3.2.1
NUMBER_OF_PROCESSORS	8
OS	Windows_NT
Path	C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

5.3 Intellij下没有输出提示直接执行结束

如下图所示，没有输出任何提示性输出，比如执行进程，直接显示执行成功：



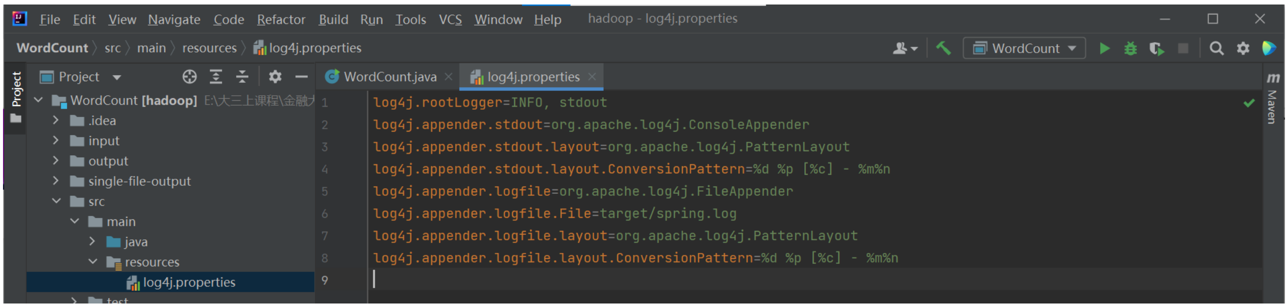
stop-word-list.txt
External Libraries
Run: WordCount_eg
Git Run TODO Problems Terminal Build Dependencies
Event Log
All files are up-to-date (moments ago)
77:13 CRLF UTF-8 2 spaces* master

```
while ((pattern = fis.readLine()) != null) {
    patternsToSkip.add(pattern);
```

```
"C:\Program Files\Java\jdk1.8.0_301\bin\java.exe" ...
log4j:WARN No appenders could be found for logger (org.apache.hadoop.metrics2.lib.MutableMetricsFactory).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.

Process finished with exit code 0
```

在`src/main/resources`目录下创建`log4j.properties`，文件内容为：



File Edit View Navigate Code Refactor Build Run Tools VCS Window Help hadoop - log4j.properties
WordCount src main resources log4j.properties
WordCount
Project
WordCount [hadoop] E:\大三上课程\金融
.idea
input
output
single-file-output
src
main
java
resources
log4j.properties
test
WordCount.java
log4j.properties
Maven

```
log4j.rootLogger=INFO, stdout
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d %p [%c] - %m%n
log4j.appender.logfile=org.apache.log4j.FileAppender
log4j.appender.logfile.File=target/spring.log
log4j.appender.logfile.layout=org.apache.log4j.PatternLayout
log4j.appender.logfile.layout.ConversionPattern=%d %p [%c] - %m%n
```

5.4 试图魔改Combiner

我希望在Combiner中计算出词频，然后将key与value倒置以进行词频排序，但是输出错误，错误提示：

`wrong value class: class org.apache.hadoop.io.Text is not class org.apache.hadoop.io.IntWritable`

StackOverFlow解释：

> *Output types of a combiner **must** match output types of a mapper. Hadoop makes no guarantees on how many times the combiner is applied, or that it is even applied at all. And that's what happens in your case.*
>
> *Values from map (`<Text, IntWritable>`) go directly to the reduce where types `<Text, Text>` are expected.*

所以放弃这种做法，转而采取报告中所写的。

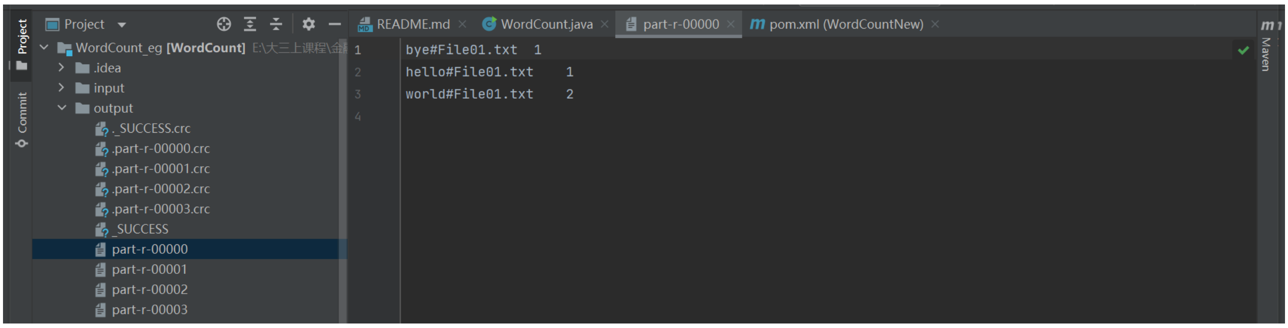
5.5 采用倒排索引中的NewPartitioner但是只有一个reducer

一个reducer应该有一个对应的输出文件。参考倒排索引中的NewPartitioner，我想让所有具有相同filename的文件进入同一个reducer然后被输出。在这种逻辑下，应该有多个文件输出，但是试试并非如此。

后来发现，需要在设置partitioner之后规定reduce的task个数：

```
job.setPartitionerClass(NewPartitioner.class);
job.setNumReduceTasks(4);
```

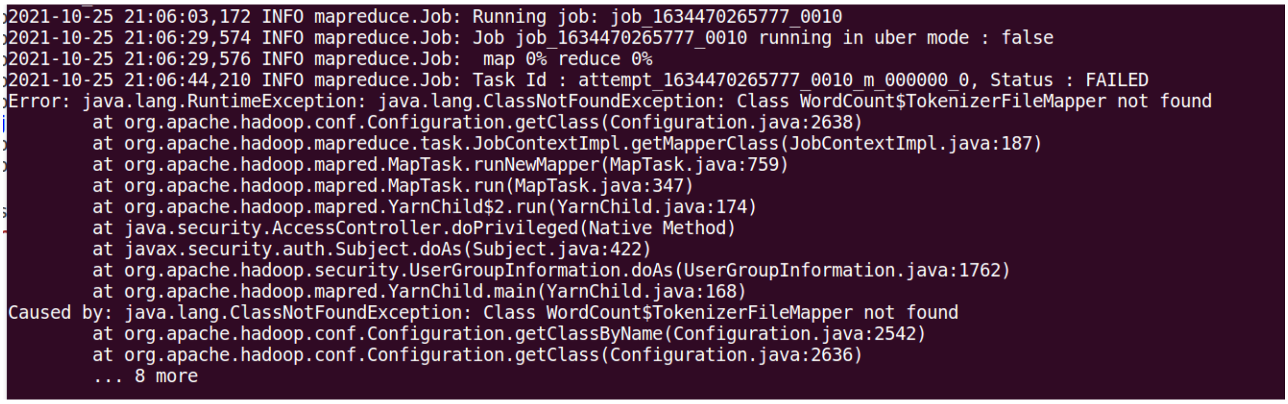
这样就可以做到用partition将文件分到不同reducer，并输出多个文件：



```
bye#File01.txt	1
hello#File01.txt	1
world#File01.txt	2
```

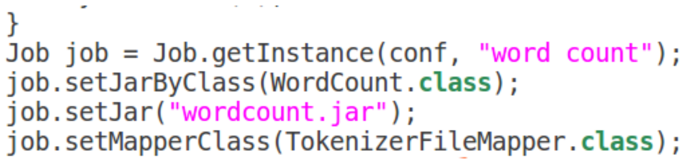
5.6 将代码转移至Linux执行报错

执行过程中报找不到Class的错误：



```
2021-10-25 21:06:03,172 INFO mapreduce.Job: Running job: job_1634470265777_0010
2021-10-25 21:06:29,574 INFO mapreduce.Job: Job job_1634470265777_0010 running in uber mode : false
2021-10-25 21:06:29,576 INFO mapreduce.Job:  map 0% reduce 0%
2021-10-25 21:06:44,210 INFO mapreduce.Job: Task Id : attempt_1634470265777_0010_m_000000_0, Status : FAILED
Error: java.lang.RuntimeException: java.lang.ClassNotFoundException: Class WordCount$TokenizerFileMapper not found
        at org.apache.hadoop.conf.Configuration.getClass(Configuration.java:2638)
        at org.apache.hadoop.mapreduce.task.JobContextImpl.getMapperClass(JobContextImpl.java:187)
        at org.apache.hadoop.mapred.MapTask.runNewMapper(MapTask.java:759)
        at org.apache.hadoop.mapred.MapTask.run(MapTask.java:347)
        at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:174)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:422)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1762)
        at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:168)
Caused by: java.lang.ClassNotFoundException: Class WordCount$TokenizerFileMapper not found
        at org.apache.hadoop.conf.Configuration.getClassByName(Configuration.java:2542)
        at org.apache.hadoop.conf.Configuration.getClass(Configuration.java:2636)
        ... 8 more
```

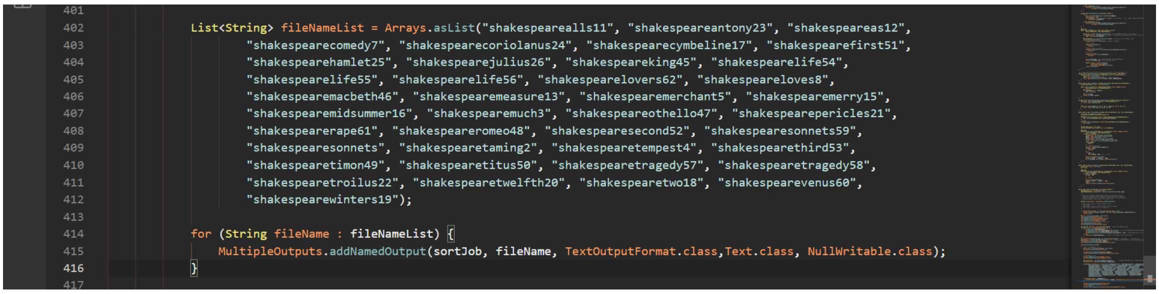
在代码中加上`job.setJar("wordcount.jar")`即可：



```
}
Job job = Job.getInstance(conf, "word count");
job.setJarByClass(WordCount.class);
job.setJar("wordcount.jar");
job.setMapperClass(TokenizerFileMapper.class);
```

6. 任务可以改进的地方

1. 现阶段都是在一个reducer上进行，可以多用几个reducer提升效率。
2. 存在硬编码情况，多文件输出时需要定义，我获取了input目录下所有文件名循环执行了定义，不知道有没有更加高效的方法。



```
        List<String> fileNameList = Arrays.asList("shakespeareall s11", "shakespeareantony23", "shakespeareas12",
                "shakespearecomedy7", "shakespearecoriolanus24", "shakespearecymbeline17", "shakespearefirst51",
                "shakespearehamlet25", "shakespearejulius26", "shakespeareking45", "shakespearelife54",
                "shakespearelife55", "shakespearelife56", "shakespearelovers62", "shakespeareloves8",
                "shakespearemacbeth46", "shakespearemeasure13", "shakespearemerchant5", "shakespearemerry15",
                "shakespearemidsummer16", "shakespearemuch3", "shakespeareothello47", "shakespearepericles21",
                "shakespearerape61", "shakespeareromeo48", "shakespearesecond52", "shakespearesonnets59",
                "shakespearesonnets", "shakespearetaming2", "shakespearetempest4", "shakespearethird53",
                "shakespearetimon49", "shakespearetitus50", "shakespearetragedy57", "shakespearetragedy58",
                "shakespearetroilus22", "shakespearetwelfth20", "shakespearetwo18", "shakespearevenus60",
                "shakespearewinters19");

        for (String fileName : fileNameList) {
            MultipleOutputs.addNamedOutput(sortJob, fileName, TextOutputFormat.class,Text.class, NullWritable.class);
        }
```

3. 程序功能可以更多样， 例如进行名词单、复数、动词时态的还原等。
4. 现阶段计算“每个文件高频前100”与“所有文件高频前100”比较割裂没有联系，可以探索是否有方法减少job，通过使得前后计算结果可以被充分利用。