莎士比亚文集词频统计并行化算法

1. 设计思路

本题的要求是根据给定的停词表，统计出莎士比亚文集中频度最高的100个单词。因此首先需要像Problem1一样统计出每个单词的出现次数，根据出现次数排序之后再选取前100个单词。单词的次数统计基本如同简单的WordCount程序，空格分隔需替换成特殊标点符号的分隔（利用正则表达式），统计时忽略停词表中存在的单词即可。排序时可将键值交换，利用sortByKey等接口进行排序。

2. 实现方案

2.1 停词表的存储

考虑到停词表中的停词不是特别多，因此将它们从文件中逐个读入内存（注意处理每个停词之后的空格）形成停词列表。

2.2 词频统计

循环读入每一行文本，利用正则表达式将文本分割成一个个单词并转换成小写形式，过滤掉所有不包含在停词列表中的单词以及分割导致的空字符串。利用简单的单词计数算法求出每个单词的频数形成<key, value>对。根据频数进行排序时将key和value交换（swap）位置，sortedByKey(False)按频数从大到小排序。最后take(100)得到频度最高的100个单词。

3. 测试结果

3.1 jar包使用说明

spark-submit

—class tongji.dataspark.p4.Problem4

—master <master-url>

DataSpark_problem4.jar

data-path

stopword-file-path

result-file-path

3.2 任务提交

```
[bookcold@namenode ~]$ cd mathpanda/
[bookcold@namenode mathpanda]$ ../spark-1.0.1/bin/spark-submit --class tongji.da
taspark.p4.Problem4 --master spark://192.168.1.18:7077 ./DataSpark_problem4.jar
 /user/mathpanda/WordCount/shakespear /user/mathpanda/WordCount/stopword.txt  `p
wd`/problem4_result.txt
```

3.3 运行情况

Completed Applications

ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
app-20141212011217-0017	Problem4	192	40.0 GB	2014/12/12 01:12:17	bookcold	FINISHED	11 s

3.4 执行结果

problem4_result.txt

```
thou
him
so
what
thy
all
by
we
king
lord
thee
o
sir
now
good
they
come
which
let
more
then
love
well
how
when
enter
them
man
hath
one
```