Spark RDD API 作业实践

作业一 使用RDD API实现带词频的倒排索引

spark-shell(scala)

单个文件的倒排索引

读取文件

```
scala> val x = sc.textFile("/Users/chenhao/github_code/
study_made_me_happy/bigdatacamp/phoneData/src/sparkcore/data/
0.txt")
x: org.apache.spark.rdd.RDD[String] = /Users/chenhao/github_code/
study_made_me_happy/bigdatacamp/phoneData/src/sparkcore/data/
0.txt MapPartitionsRDD[88] at textFile at <console>:24
```

进行倒排索引

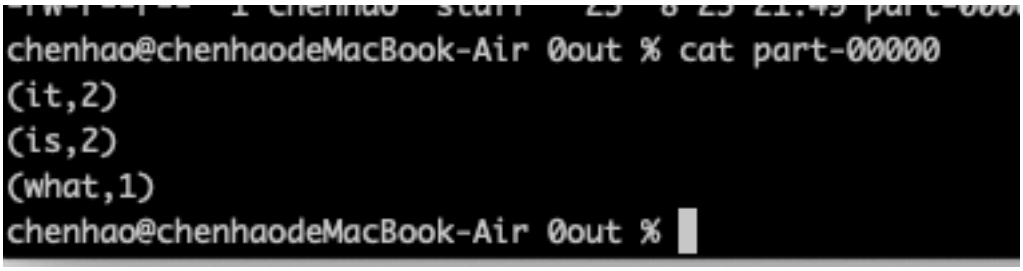
```
scala> val y = x.flatMap(_.split(" ")).map((_,
1)).reduceByKey(_+_, 1).sortBy(_._2, false)
y: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[94]
at sortBy at <console>:25

scala> print(y.collect().mkString(";"))
(it,2);(is,2);(what,1)
```

输出结果至本地文件

```
scala> y.saveAsTextFile("/Users/chenhao/github_code/
study_made_me_happy/bigdatacamp/phoneData/src/sparkcore/0out")
```

观察结果



多个文件的倒排索引

读取所有文件

```
val data = sc.wholeTextFiles("/Users/chenhao/github_code/
study_made_me_happy/bigdatacamp/phoneData/src/sparkcore/data")
```

清洗文件目录和分词

```
val dataTu = data.map(data => (data._1.replace("file:/Users/
chenhao/github_code/study_made_me_happy/bigdatacamp/phoneData/
src/sparkcore/
data/","").replace(".txt",""),data._2.split(System.getProperty("l
ine.separator")).flatMap(_.split(" "))))
```

变量dataTu的结果显示

```
scala> print(dataTu.collect().mkString(";"))
(2,[Ljava.lang.String;@6cafc524);(0,
[Ljava.lang.String;@7390d6a6);(1,[Ljava.lang.String;@bd126cc)
```

转变为(word, file)的形式

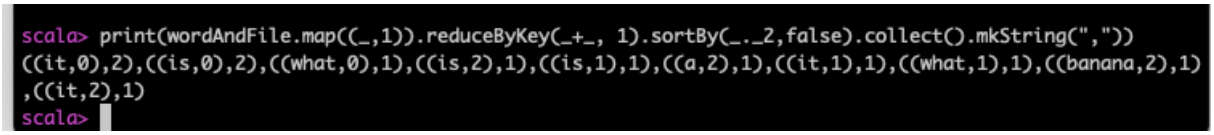
```
val wordAndFile = dataTu.flatMap(a=>a._2.map(word=>(word,a._1)))
```

(word, file)的结果

```
scala> print(wordAndFile.collect().mkString(";"))
(it,2);(is,2);(a,2);(banana,2);(it,0);(is,0);(what,0);(is,0);
(it,0);(what,1);(is,1);(it,1)
```

倒排结果

```
scala> print(wordAndFile.map((_,1)).reduceByKey(_+_,
1).sortBy(_._2,false).collect().mkString(","))
((it,0),2),((is,0),2),((what,0),1),((is,2),1),((is,1),1),
((a,2),1),((it,1),1),((what,1),1),((banana,2),1),((it,2),1)
```



本地单文件倒排索引Scala代码

```
package sparkcore

import org.apache.spark.{SparkConf, SparkContext}

/**
 * localMode
 */
object WordCount {
  def main(args: Array[String]): Unit = {
    val conf = new SparkConf().setMaster("local").setAppName("My
App")
    val sc = new SparkContext(conf)

    //使用sc创建RDD并执行相应的transformation和action
    sc.textFile("/Users/chenhao/github_code/study_made_me_happy/
bigdatacamp/phoneData/src/sparkcore/data/1.txt")
      .flatMap(_.split(" "))
      .map((_, 1))
      .reduceByKey(_ + _, 1)
      .sortBy(_._2, false)
      .saveAsTextFile("/Users/chenhao/github_code/
study_made_me_happy/bigdatacamp/phoneData/src/sparkcore/1out")
  }
}
```

实现结果与上述spark-shell的类似。

多文件实现暂时没有好的解决方法，希望老师可以提供参考方案。

作业二 Distcp的spark实现

暂时实现不了。