Spark高级练习二

**练习1：**统计HTTP日志返回代码

1. 创建com.aura.spark.LogStatistics类，定义JavaSparkContext，输入下列代码

|  |
| --- |
| package com.aura.spark;  import org.apache.spark.Accumulator; import org.apache.spark.SparkConf; import org.apache.spark.api.java.JavaRDD; import org.apache.spark.api.java.JavaSparkContext; import org.apache.spark.util.LongAccumulator;  public class LogStatistics {  public static void main(String[] args) {  SparkConf conf = new SparkConf()  .setMaster("local[1]")  .setAppName("LogStatistics");  JavaSparkContext jsc = new JavaSparkContext(conf);   Accumulator<Integer> total = jsc.accumulator(0);  Accumulator<Integer> count400 = jsc.accumulator(0);  Accumulator<Integer> count200 = jsc.accumulator(0);   JavaRDD<String> input = jsc.textFile(args[0]);  input.foreach(s -> {  String[] row = s.split(",");  // TODO add your code here  });   // TODO add your code here  jsc.stop();  } } |

2. 在//TODO注释处添加加入代码, 完成下列功能

* 使用累加器计数
* 打印出总数，400的个数，200的个数

3. 在IDE中运行代码检验结果

配置程序运行参数，让其从data/access.log中读取数据

**练习2**：哈姆雷特词频分析

1. data目录中Hamlet.txt和stopword.txt放到/tmp目录下

2.创建com.aura.spark.HamletStatistics类，定义JavaSparkContext，输入下列代码

|  |
| --- |
| package com.aura.spark;  import org.apache.spark.SparkConf; import org.apache.spark.api.java.JavaPairRDD; import org.apache.spark.api.java.JavaRDD; import org.apache.spark.api.java.JavaSparkContext; import org.apache.spark.api.java.function.Function; import org.apache.spark.api.java.function.Function2; import org.apache.spark.api.java.function.PairFunction; import org.apache.spark.sql.catalyst.expressions.IntegerLiteral; import scala.Function1; import scala.Tuple2;  import java.util.ArrayList; import java.util.List;  public class HamletStatistics {  public static void main(String[] args) {  SparkConf conf = new SparkConf()  .setMaster("local[1]")  .setAppName("HamletStatistics");  JavaSparkContext jsc = new JavaSparkContext(conf);   JavaRDD<String> stopwords = jsc.textFile(args[0]);  // TODO 1.collect stopwords and broadcast to executors  // TODO 2.define accumulators, countTotal and stopTotal  JavaRDD<String> input = jsc.textFile(args[1]);  JavaRDD<String> words = input.flatMap(s -> splitWords(s).iterator());  JavaRDD<String> filteredWords = words.filter(new Function<String, Boolean>() {  @Override  public Boolean call(String v1) throws Exception {  // TODO filter stop words, increase countTotal or stopTotal  return null;  }  });   JavaPairRDD<String, Integer> counts = filteredWords.mapToPair(new PairFunction<String, String, Integer>() {  @Override  public Tuple2<String, Integer> call(String s) throws Exception {  // TODO add your code here  return null;  }  }).reduceByKey(new Function2<Integer, Integer, Integer>() {  @Override  public Integer call(Integer v1, Integer v2) throws Exception {  // TODO add your code here  return null;  }  });  // TODO sort result   // TODO output result   jsc.stop();  }   public static List<String> splitWords(String line) {  List<String> result = new ArrayList<String>();  String[] words = line.replaceAll("['.,:?!-]", "").split("\\s");  for (String w: words) {  if (!w.trim().isEmpty()) {  result.add(w.trim());  }  }  return result;  } } |

2. 在//TODO注释处添加加入代码, 完成下列功能：

* 将读取的停止词广播到各个executor
* 使用累加器同时统计总单词数和总停止词数
* 输出出现次数最高的前10个单词

3. 在IDE中运行代码，检查结果