

1. java 编写 spark 程序

scala 原生集合或数组中的方法是针对集合或数组的单机操作；spark 中的方法是针对分布式数据集 RDD 的分布式操作；两者实现不同。

```
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.FlatMapFunction;
import org.apache.spark.api.java.function.Function2;
import org.apache.spark.api.java.function.PairFunction;
import scala.Tuple2;

import java.util.Arrays;

/**
 * Created by ZhaoXing on 2016/8/22.
 */
public class JavaWordCount {

    public static void main(String[] args) {
        SparkConf conf = new SparkConf().setAppName("JavaWordCount");
        //创建 java sparkcontext
        JavaSparkContext jsc = new JavaSparkContext(conf);
        //读取数据
        JavaRDD<String> lines = jsc.textFile(args[0]);
        //切分
        JavaRDD<String> words = lines.flatMap(new FlatMapFunction<String, String>() {
            @Override
            public Iterable<String> call(String line) throws Exception {
                return Arrays.asList(line.split(" "));
            }
        });
        //单词计数
        JavaPairRDD<String, Integer> wordAndOne = words.mapToPair(new
        PairFunction<String, String, Integer>() {
```

```
@Override
    public Tuple2<String, Integer> call(String word) throws Exception {
        return new Tuple2<String, Integer>(word, 1);
    }
});
//分组聚合
JavaPairRDD<String, Integer> result = wordAndOne.reduceByKey(new
Function2<Integer, Integer, Integer>() {
    @Override
    public Integer call(Integer i1, Integer i2) throws Exception {
        return i1 + i2;
    }
});
//反转顺序
JavaPairRDD<Integer, String> swapedPair = result.mapToPair(new
PairFunction<Tuple2<String, Integer>, Integer, String>() {
    @Override
    public Tuple2<Integer, String> call(Tuple2<String, Integer> tp) throws Exception {
        return new Tuple2<Integer, String>(tp._2, tp._1);
    }
});
//排序并调换顺序
JavaPairRDD<String, Integer> finalResult = swapedPair.sortByKey(false).mapToPair(new
PairFunction<Tuple2<Integer, String>, String, Integer>() {
    @Override
    public Tuple2<String, Integer> call(Tuple2<Integer, String> tp) throws Exception {
        return tp.swap();
    }
});

//保存
finalResult.saveAsTextFile(args[1]);
jsc.stop();
}
}
```