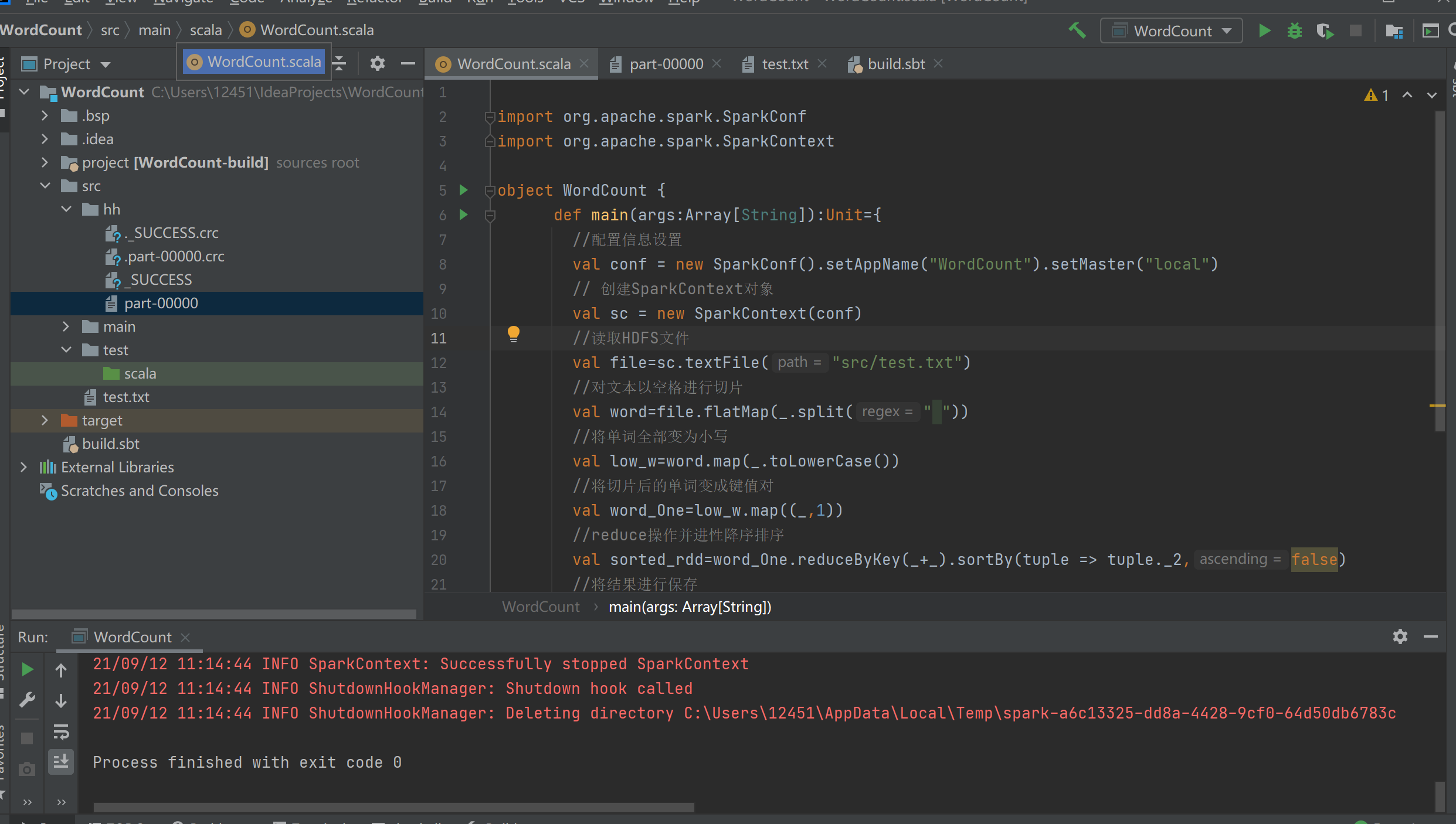
用Scala进行WordCount

1. 作业源码及截图



import org.apache.spark.SparkConf

import org.apache.spark.SparkContext

object WordCount {

def main(args:Array[String]):Unit={

//配置信息设置

val conf = new SparkConf().setAppName("WordCount").setMaster("local")

// 创建SparkContext对象

val sc = new SparkContext(conf)

//读取HDFS文件

val file=sc.textFile("src/test.txt")

//对文本以空格进行切片

val word=file.flatMap(\_.split(" "))

//将单词全部变为小写

val low\_w=word.map(\_.toLowerCase())

//将切片后的单词变成键值对

val word\_One=low\_w.map((\_,1))

//执行reduce操作并进行降序排序

val sorted\_rdd=word\_One.reduceByKey(\_+\_).sortBy(tuple => tuple.\_2,false)

//将结果进行保存

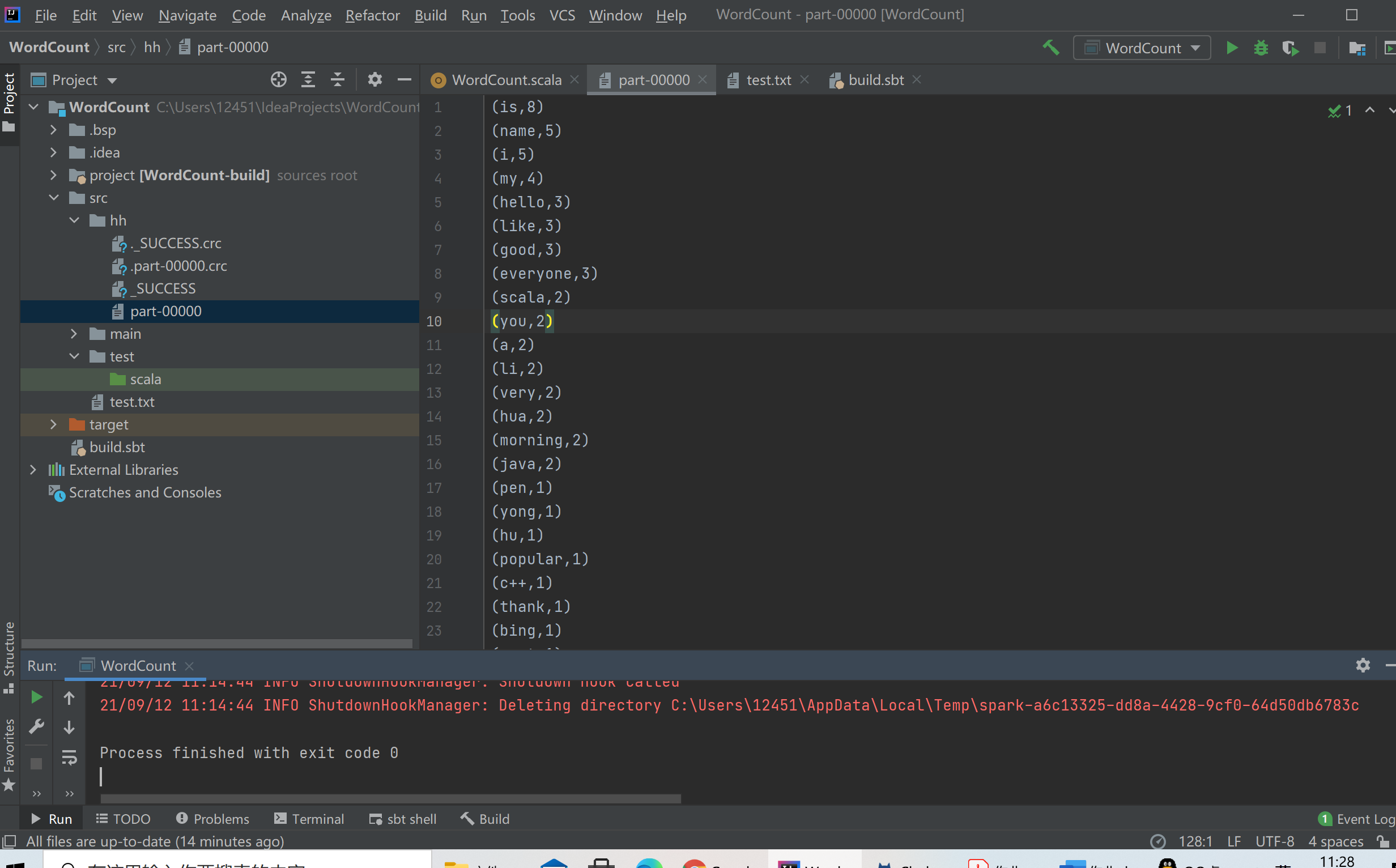
sorted\_rdd.saveAsTextFile("src/hh")

sc.stop()

}

}

1. 运行结果截图



1. 实现思路

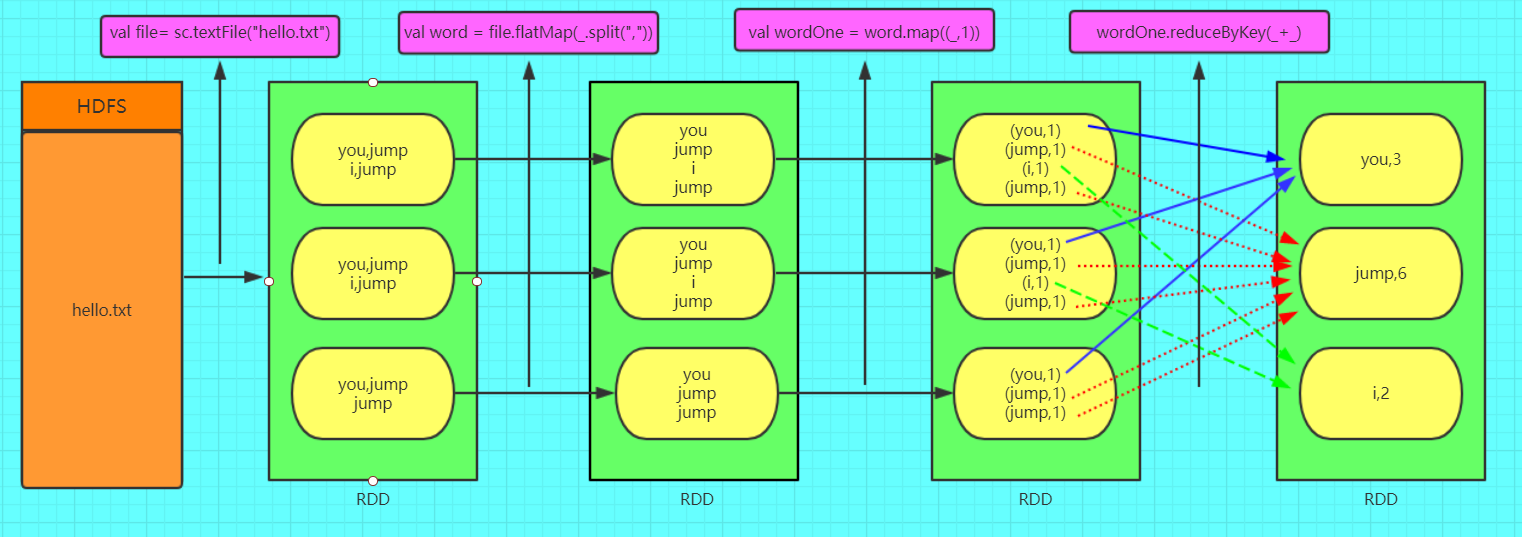
1、关于环境的配置

在IEDA中下载SCALA插件。从网上下载Spark包，导入其中的jar文件。运行时发现缺少HADOOP的HDFS作为文件管理系统。再下载HADOOP，并配置相应环境。

2.关于算法的实现

将文档做切片处理（map），并全部转化为小写（以实现不区分大小写），再将单词进行合并（reduce）。

1. 实验心得

环境搭建那部分耗时较长，但根据报错一步步来，就能解决。算法实现部分，由于对Spark不是太熟悉，不知道有哪些具体的方法可以调用。我查找了一些调用方法。

（该图选自网络）

上图对我的启发较大，也让我搞清楚了原理。