第15课：RDD创建内幕

1. RDD的创建方式

Spark应用程序运行过程中，第一个RDD代表了Spark应用程序输入数据的来源，之后通过Trasformation来对RDD进行各种算子的转换，来实现具体的算法

Spark中的基本方式：

1. 使用程序中的集合创建

这种方式的实际意义主要用于测试。

1. 使用本地文件系统创建

这种方式的实际意义主要用于测试大量数据的文件

1. 使用HDFS创建RDD

这种方式为生产环境中最常用的创建RDD的方式

1. 基于DB创建
2. 基于NoSQL:例如HBase
3. 基于S3(SC3)创建
4. 基于数据流创建
5. RDD创建实战
6. 通过集合创建

代码：

**object** RDDBasedOnCollection {  
 **def** main (args: Array[String]) {  
 **val** conf = **new** SparkConf()*//create SparkConf* conf.setAppName(**"RDDBasedOnCollection"**)*//set app name* conf.setMaster(**"local"**)*//run local* **val** sc =**new** SparkContext(conf)  
 **val** numbers = 1 to 100 *//创建一个Scala集合* **val** rdd = sc.parallelize(numbers)  
 **val** sum =rdd.reduce(\_+\_) *//1+2=3 3+3=6 6+4=10  
 println*(**"1+2+...+99+100"**+**"="**+sum)  
 }  
}

结果：



1. 通过本地文件系统创建

代码：

**object** RDDBasedOnLocalFile {  
 **def** main (args: Array[String]) {  
 **val** conf = **new** SparkConf()*//create SparkConf* conf.setAppName(**"RDDBasedOnCollection"**)*//set app name* conf.setMaster(**"local"**)*//run local* **val** sc =**new** SparkContext(conf)  
 **val** rdd = sc.textFile(**"C:/Users/feng/IdeaProjects/WordCount/src/SparkText.txt"**)  
 **val** linesLength=rdd.map(line=>line.length())  
 **val** sum = linesLength.reduce(\_+\_)  
 *println*(**"the total characters of the file"**+**"="**+sum)  
 }  
}

结果：



1. 通过HDFS创建RDD

代码：

 val wordcount = sc.textFile("/library/wordcount/input/licenses").flatMap(\_.split(" ")).map(word=>(word,1)).reduceByKey(\_+\_).filter(pair=>pair.\_2>20).collect().foreach(println)

结果：

