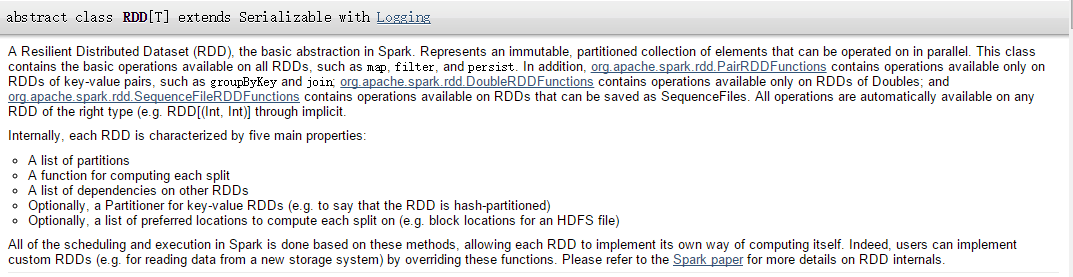
Spark-RDD操作解析

这两天学习了RDD的操作，于是今天就自己总结一下关于RDD操作（参考百度），写的都是自己的理解，如果有哪里不够严谨或理解不正确，多多指正，OK，接下来我们进入主题。

Spark的核心就是RDD，那么RDD到底是什么？首先来看一下官方文档上是怎么说的吧



1.RDD是一个弹性分布数据集，那么什么又是弹性分布呢,分开来理解，弹性其实就是在计算数据内存不够时会和磁盘进行数据交换，分布就是可以分布在多台机器上进行计算。

2.RDD是只读的、分区记录的集合。RDD只能基于在稳定物理存储中的数据集和其他已有的RDD上执行确定性操作来创建。这些确定性操作称之为转换，如map、filter、groupBy、join（转换是程开发人员在RDD上执行的操作）。

3.更简单的理解是，RDD是Spark中的抽象数据类型，任何的数据在Spark中都能表示为RDD，对于编程来说，RDD你可以理解成为一个数组，只不过和普通数组的区别是RDD的数据是分区存储的，这样不同分区的数据就可以在不同的机器上可以并行处理。因此，Spark应用程序所做的无非就是把需要处理的数据转换为RDD，然后对RDD进行一系列的变换和操作从而得到我们需要的结果。

创建一个RDD，可以从普通数组中创建，也可以从文件系统或者HDFS中的文件创建出来。

val test1 = sc.parallelize(1 to 9,3)//从普通数组中创建的RDD，里面包含了1到9个数字，分别在3个分区中。

Val test2=sc.textFile(“README.txt”)//读取文件README.txt来创建RDD，文件中的每一行就是RDD中的一个元素。

RDD操作的两种形式：

1. Actions：对数据集计算后返回一个数值value给驱动程序
2. Transformation：根据数据集创建一个新的数据集，计算后返回一个新RDD

常用的Action操作总结：

reduce(func)

通过函数func聚集数据集中的所有元素。Func函数接受2个参数，返回一个值。这个函数必须是关联性的，确保可以被正确的并发执行

collect()

在Driver的程序中，以数组的形式，返回数据集的所有元素。这通常会在使用filter或者其它操作后，返回一个足够小的数据子集再使用，直接将整个RDD集Collect返回，很可能会让Driver程序OOM

count()

返回数据集的元素个数

take(n)

返回一个数组，由数据集的前n个元素组成。注意，这个操作目前并非在多个节点上，并行执行，而是Driver程序所在机器，单机计算所有的元素

(Gateway的内存压力会增大，需要谨慎使用）

first()

返回数据集的第一个元素（类似于take（1））

saveAsTextFile(path)

将数据集的元素，以textfile的形式，保存到本地文件系统，hdfs或者任何其它hadoop支持的文件系统。Spark将会调用每个元素的toString方法，并将它转换为文件中的一行文本

saveAsSequenceFile(path)

将数据集的元素，以sequencefile的格式，保存到指定的目录下，本地系统，hdfs或者任何其它hadoop支持的文件系统。RDD的元素必须由key-value对组成，并都实现了Hadoop的Writable接口，或隐式可以转换为Writable（Spark包括了基本类型的转换，例如Int，Double，String等等）

foreach(func)

在数据集的每一个元素上，运行函数func。这通常用于更新一个累加器变量，或者和外部存储系统做交互

常用的Transformation操作总结：

map(func)

返回一个新的分布式数据集，由每个原元素经过func函数转换后组成(里面跟一个函数)

filter(func)

返回一个新的数据集，由经过func函数后返回值为true的原元素组成，可以用于过滤掉你需要的数据

flatMap(func)

类似于map，但是每一个输入元素，会被映射为0到多个输出元素（因此，func函数的返回值是一个Seq，而不是单一元素）

sample(withReplacement, frac, seed)

根据给定的随机种子seed，随机抽样出数量为frac的数据

union(otherDataset)

返回一个新的数据集，由原数据集和参数联合而成

groupByKey([numTasks])

在一个由（K,V）对组成的数据集上调用，返回一个（K，Seq[V])对的数据集。注意：默认情况下，使用8个并行任务进行分组，你可以传入numTask可选参数，根据数据量设置不同数目的Task

reduceByKey(func, [numTasks])

在一个（K，V)对的数据集上使用，返回一个（K，V）对的数据集，key相同的值，都被使用指定的reduce函数聚合到一起。和groupbykey类似，任务的个数是可以通过第二个可选参数来配置的。

join(otherDataset, [numTasks])

在类型为（K,V)和（K,W)类型的数据集上调用，返回一个（K,(V,W))对，每个key中的所有元素都在一起的数据集

groupWith(otherDataset, [numTasks])

在类型为（K,V)和(K,W)类型的数据集上调用，返回一个数据集，组成元素为（K, Seq[V], Seq[W]) Tuples。这个操作在其它框架，称为CoGroup

cartesian(otherDataset)

笛卡尔积。但在数据集T和U上调用时，返回一个(T，U）对的数据集，所有元素交互进行笛卡尔积。

sortByKey([ascendingOrder])

在类型为( K, V )的数据集上调用，返回以K为键进行排序的（K，V）对数据集。升序或者降序由可选的boolean型的ascendingOrder参数决定，默认为true,升序。

接下来就用我写的一个例子分析一下

任务数据：在Test2文件中有三个文件

1.txt

1-lienlian

2-wangyan

3-kexu //学号-学生姓名

4-sunhaiyu

5-sunwen

2.txt

1-chenhongjun

2-huangmanqi

3-liyue

4-xiecaiyun //老师编号-姓名

5-luxinwen

6-huheng

7-zhouqianyin

3.txt

1-1-7

1-2-9

1-3-3

1-4-5

1-5-3

1-6-2

1-7-6

2-1-4

2-2-5

2-3-7

2-4-5

2-5-3

2-6-8

2-7-6 //学生学号-老师编号-学生对老师的评分

3-1-4

3-2-5

3-3-4

3-4-7

3-5-10

3-6-1

3-7-8

4-1-5

4-2-1

4-3-6

4-4-4

4-5-9

4-6-5

4-7-7

5-1-10

5-2-7

5-3-4

5-4-6

5-5-1

5-6-8

5-7-9

任务目的

输出每个老师被评的最低分以及评最低分的学生的姓名

样式：

(老师姓名,最低评分,评最低分的学生的姓名)

(huangmanqi,1.0,ArrayBuffer(sunhaiyu))

(chenhongjun,4.0,ArrayBuffer(kexu, wangyan))

(zhouqianyin,6.0,ArrayBuffer(lienlian, wangyan))

(liyue,3.0,ArrayBuffer(lienlian))

(xiecaiyun,4.0,ArrayBuffer(sunhaiyu))

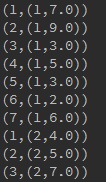
(huheng,1.0,ArrayBuffer(kexu))

(luxinwen,1.0,ArrayBuffer(sunwen))

分析

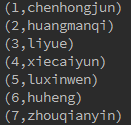
第3.txt文件中封装了老师以及学生的编号和学生编号以及评分,所以我们肯定从这个文件入手，将所需要的数据提取出来

执行结果(只显示了部分)



然后我们再看要求,老师姓名和学生姓名给输出出来,我们现在输出的只是编号,需要转换为对应的老师的姓名,那么怎么转呢,老师姓名与编号的对应关系在2.txt的文件,接下来我们就是读取第二个文件的信息,然后将两个RDD关联起来

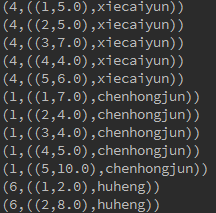
执行结果



那么怎么将这两个RDD关联起来呢,我们来查找一下RDD的操作方法,join方法在类型为（K,V)和（K,W)类型的数据集上调用，返回一个（K,(V,W))对，每个key中的所有元素都在一起的数据集,在图中的显示结果里，发现我们两个RDD的K值都是老师编号，那就意味着我们能够将这两个RDD连接起来组成一个新的RDD.那么我们就开始操作吧



执行结果(部分)

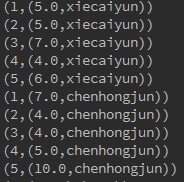


这样就将老师名字和评分显示在同一个RDD中,再来观察一下，我们的得到结果,现在有哪些是不需要的呢？老师编号现在肯定不需要，我们已经有老师的姓名来了，其他学生的学号和评分还是保留,我们接下来就得将不需要的数据剔除掉,那么怎么做呢,再次查看RDD的操作,map(func)方法可以返回一个新的RDD，由每个原元素经过func函数转换后组成(里面跟一个函数)



这里面那么多.\_1.\_2的什么意思？在scala map元素中取值第一个值就用.\_1第二个就用.\_2.

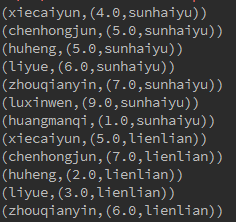
i.\_2.\_1.\_1是什么意思呢？拆开来看,i.\_2得到第二个元素里面还是一个map得到的就是((2,8.0),huheng),在.\_1得到的就是(2,8.0)最后在.\_1得到的就是学生学号,我们来看一下结果



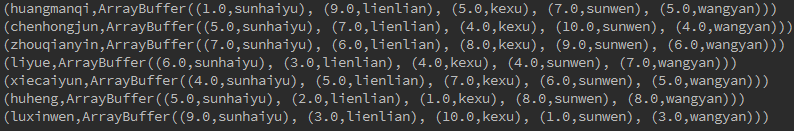
（学生学好,（评分,老师姓名））

现在，同样的我们将学生姓名连接起来,利用的是学生的学号,然后剔除掉不需要的数据

执行结果：



（老师姓名,（学生评分,学生姓名））

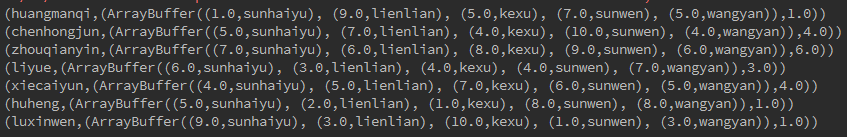
我们发现K值有相同的,我们能不能相同K值的放在一起，查看关于RDD的操作,有一个goupByKey的方法，返回的是相同K值,Value值的集合,再来看下结果

我们的要求是找到最低分以及打分的同学,那么我们肯定得先找到最低分，在scala ArrayBuffer有一个得到最小值min的方法

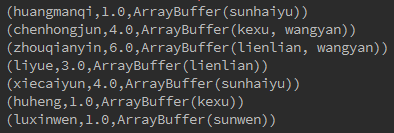


最后将（老师姓名，老师得到的最低分）的RDD和上一步得到的（老师姓名，所有学生及对应评分）的RDD关联起来,然后过滤掉评分不等于最低分的同学,就完成了我们的要求

join后的结果

（老师姓名，学生对应评分的集合，最低分数）

filter后的最终结果



这样就得到了我们的需要的数据

从这个例子中可以看到RDD的操作就是RDD之间的转换,用各中不同的方法得到新的RDD,最终目的是为了得到我们需要的结果,RDD更多的操作请查看官方文档