**创建一个学生成绩说明的类**

case class ScoreDetail(studentName: String, subject: String, score: Float)

* 1

**下面是一些测试数据**，加载测试数据集合 key = Students name and value = ScoreDetail instance

val scores = List(

ScoreDetail("xiaoming", "Math", 98),

ScoreDetail("xiaoming", "English", 88),

ScoreDetail("wangwu", "Math", 75),

ScoreDetail("wangwu", "English", 78),

ScoreDetail("lihua", "Math", 90),

ScoreDetail("lihua", "English", 80),

ScoreDetail("zhangsan", "Math", 91),

ScoreDetail("zhangsan", "English", 80))

**将集合转换成二元组**， 也可以理解成转换成一个map, 利用了for 和 yield的组合

val scoresWithKey = for { i <- scores } yield (i.studentName, i)

* 1

**创建RDD, 并且指定三个分区**

val scoresWithKeyRDD = sc.parallelize(scoresWithKey).partitionBy(new HashPartitioner(3)).cache

* 1

**输出打印一下各个分区的长度和各个分区的一些数据**

println(">>>> Elements in each partition")

scoresWithKeyRDD.foreachPartition(partition => println(partition.length))

// explore each partition...

println(">>>> Exploring partitions' data...")

scoresWithKeyRDD.foreachPartition(

partition => partition.foreach(

item => println(item.\_2)))

/\*

会输出

>>>> Elements in each partition

6

2

0

>>>> Exploring partitions' data...

ScoreDetail(xiaoming,Math,98.0)

ScoreDetail(xiaoming,English,88.0)

ScoreDetail(lihua,Math,90.0)

ScoreDetail(lihua,English,80.0)

ScoreDetail(zhangsan,Math,91.0)

ScoreDetail(zhangsan,English,80.0)

ScoreDetail(wangwu,Math,75.0)

ScoreDetail(wangwu,English,78.0)

* 27

**聚合求平均值让后打印**

val avgScoresRDD = scoresWithKeyRDD.combineByKey(

(x: ScoreDetail) => (x.score, 1) /\*createCombiner\*/,

(acc: (Float, Int), x: ScoreDetail) => (acc.\_1 + x.score, acc.\_2 + 1) /\*mergeValue\*/,

(acc1: (Float, Int), acc2: (Float, Int)) => (acc1.\_1 + acc2.\_1, acc1.\_2 + acc2.\_2) /\*mergeCombiners\*/

// calculate the average

).map( { case(key, value) => (key, value.\_1/value.\_2) })

avgScoresRDD.collect.foreach(println)

/\*输出:

(zhangsan,85.5)

(lihua,85.0)

(xiaoming,93.0)

(wangwu,76.5)

\*/

**解释一下scoresWithKeyRDD.combineByKey**   
**createCombiner:** (x: ScoreDetail) => (x.score, 1)   
这是第一次遇到zhangsan，创建一个函数，把map中的value转成另外一个类型 ，这里是把(zhangsan,(ScoreDetail类))**转换成**(zhangsan,(91,1))   
**mergeValue:** (acc: (Float, Int), x: ScoreDetail) => (acc.\_1 + x.score, acc.\_2 + 1) 再次碰到张三， 就把这两个合并, 这里是将(zhangsan,(91,1)) 这种类型 和 (zhangsan,(ScoreDetail类))这种类型合并，合并成了(zhangsan,(171,2))   
**mergeCombiners** (acc1: (Float, Int), acc2: (Float, Int)) 这个是将多个分区中的zhangsan的数据进行合并， 我们这里zhansan在同一个分区，这个地方就没有用上