**总复习**

**【实验目的】**

**复习所学HDFS，Rdd和DataFrame的内容**

**【实验内容】**

**一、HDFS的操作：**

**1、在hdfs中新建目录test，并通过命令查看创建结果；**

**hdfs dfs -mkdir /test**

**hdfs dfs -ls /**

**2、将本地盘/root/test下方的test1.txt上传到hdfs的myspark目录下方，并查看结果；**

**hdfs dfs -put /root/test/test1.txt /myspark**

**hdfs dfs -ls /myspark**

**4、在hdfs上查看test1.txt文件的内容，通过命令查看结果；**

**hdfs dfs -cat /myspark/test1.txt**

**5、将hdfs上myspark目录中的test1.txt下载到虚拟机本地盘/usr/local/下方，并查看结果；**

**hdfs dfs -get /myspark/test1.txt /usr/local**

**ls /usr/local/**

**6、删除hdfs上的test1.txt文件。**

**hdfs dfs -rm /myspark/test1.txt**

**7、删除hdfs上创建的目录myspark，通过命令查看结果。**

**hdfs dfs -rm -r /myspark**

**hdfs dfs -ls /**

**二、Rdd的创建方法**

**1、将本地磁盘/root/test下方的student.txt创建成rdd1，并显示查看。**

**rdd1=sc.textFile("file:///root/test/student.txt")**

**rdd1.foreach(print)**

1. **将文件student.txt上传到HDFS的myspark目录下，使用相应方法将其转换成rdd2，并显示查看；**

**hdfs dfs -put /root/test/student.txt /myspark**

**rdd2=sc.textFile("/myspark/student.txt")**

**rdd2.foreach(print)**

1. **将10以内的奇数生成列表并转换为rdd3，并显示查看。**

**data=list(range(1,10,2))**

**rdd3=sc.parallelize(data)**

**rdd3.foreach(print)**

**三、spark之RDD的创建与转换操作。**

**1、将本地磁盘/root/test下的word.txt转换成rdd10，并打印输出其结果后保存为write文件；**

**rdd10=sc.textFile("file:///root/test/word.txt")**

**rdd10.foreach(print)**

**rdd10.saveAsTextFile("file:///root/test/write")**

**2、将文件word.txt转换的rdd10使用map转换，使用空格分离转换，并查看其结果。**

**rdd11=rdd10.map(lambda x:x.split(" "))**

**rdd11.foreach(print)**

**3、将文件word.txt转换的rdd10使用flatMap转换，使用空格分离转换，并查看其结果。**

**rdd12=rdd10.flatMap(lambda x:x.split(" "))**

**rdd12.foreach(print)**

**4、使用groupByKey()转换对rdd10进行转换，并查看结果；**

**rdd13=rdd10.flatMap(lambda x:x.split(" ")).map(lambda a:(a,1)).groupByKey()**

**rdd13.foreach(print)**

**5、使用reduceByKey()转换对rdd10进行求词频，并查看结果；**

**rdd14=rdd10.flatMap(lambda x:x.split(" ")).map(lambda y:(y,1)).reduceByKey(lambda a,b:a+b)**

**print(rdd14.collect())**

**6、将文件student.txt转换成RDD，并使用持久化对行动操作count(),first(),collect(),take(n) 查看结果。**

**rdd1=sc.textFile("file:///root/test/student.txt").cache()**

**print(rdd1.count())**

**print(rdd1.first())**

**print(",".join(rdd1.collect()))**

**print(",".join(rdd1.take(2)))**

**6、将test1.txt文档转换成rdd，并设置为3个分区，命名为rdd；**

**rdd=sc.textFile("file:///root/test/test1.txt",3)**

**7、显示分区个数；重新设置分区个数为2，并显示。**

**len(rdd.glom().collect())**

**rdd1=rdd.repartition(2)**

**len(rdd1.glom().collect())**

**8、在rdd1中筛选出包含“spark”的字符串保存到新的rdd2中，并打印显示；**

**rdd1=sc.textFile("file:///root/test/word.txt")**

**rdd2=rdd1.filter(lambda x:'spark' in x)**

**rdd2.foreach(print)**

1. **DataFrame创建**

**1、将本地路径/root/test下的文件people.json，people.txt，11.csv,分别使用2种方式创建DataFrame，并显示输出；**

**第一种方法：**

**df1=spark.read.json("file:///root/test/people.json")**

**df1.show()**

**df1=spark.read.text("file:///root/test/people.txt")**

**df2.show()**

**df3=spark.read.csv("file:///root/test/11.csv")**

**df3.show()**

**第二种方法：**

**df1=spark.read.format("json").load("file:///root/test/people.json")**

**df1.show()**

**df2=spark.read.format("text").load("file:///root/test/people.txt")**

**df2.show()**

**df3=spark.read.format("csv").load("file:///root/test/11.csv")**

**df3.show()**

**2、将1题已经创建好的DataFrame保存成newpeople.json，newpeople.txt，并查看保存成功与否。**

**df1.write.json("file:///root/test/newpeople.json")**

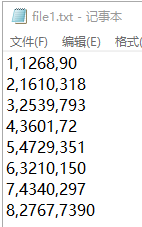
**df2.write.text("file:///root/test/newpeople.txt")**

**ls /root/test/newpeople.json**

**ls /root/test/newpeople.text**

**五、Rdd应用题**

**在/root/test目录下有2个文本文件file1.txt和file2.txt，每个文件中有很多行数据，每行数据由3个字段的值构成，不同字段之间用逗号隔开，如下图所示。其中3个字段分别是：orderid，userid，payment，现要求通过rdd的各种转换操作实现求出payment字段Top值的前5个。**



**lines=sc.textFile("file:///root/test/file\*.txt") #读取/root/test/目录下file1、2文件**

**result=lines.filter(lambda x:(len(x.strip())>0) and (len(x.split(","))==3))** #将读取的文本文件中的空行或者字段数量不等3的行去掉

**result1=result.map(lambda x:x.split(",")[2])** #将读取的文本文件按英文逗号划分，并且取出第三列payment的值

**result2=result1.map(lambda x:(int(x),""))** #将每个元素由字符串变为整型，并且将其生成（key，value）的键值对类型

**result3=result2.repartition(1)** #对其进行重新分区，因为只有所有元素都在一个分区才能排序

**result4=result3.sortByKey(False)** #然后使用sort对其key值进行排序，false代表降序

**result5=result4.map(lambda x:x[0])** #将每个键值对中的key值取出

**result5.take(5)** #这时直接使用take取出前五个元素 即payment字段top值的前五个

详细过程可以参考教材p87-89

**六、/root/test下现有学生信息文档student.txt，课程信息文档course.txt和成绩信息文档grade.txt。**

**反射机制生成student临时表：**

**from pyspark.sql import Row**

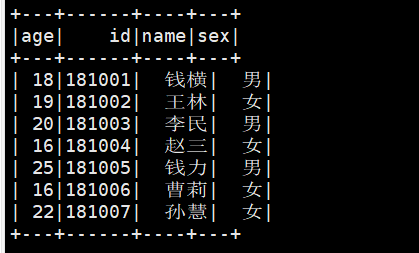
**student=spark.sparkContext.textFile("file:///root/test/student.txt").map(lambda x:x.split(",")).map(lambda p:Row(id=p[0],name=p[1],sex=p[2],age=int(p[3])))**

**student1=spark.createDataFrame(student)**

**2022-06-06 03:38:02 WARN ObjectStore:568 - Failed to get database global\_temp, returning NoSuchObjectException**

**student1.show()**

**student1.createOrReplaceTempView("student")**



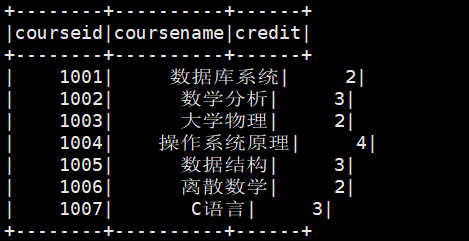
**反射机制生成course临时表：**

**course=spark.sparkContext.textFile("file:///root/test/course.txt").map(lambda x:x.split(",")).map(lambda p:Row(courseid=p[0],coursename=p[1],credit=int(p[2])))**

**course1=spark.createDataFrame(course)**

**course1.show()**

**course1.createOrReplaceTempView("course")**



**编程方式生成grade临时表：**

**from pyspark.sql.types import \***

**gradename="id courseid grade"**

**fields=[StructField(field\_name,StringType(),True) for field\_name in gradename.split(" ")]**

**grade=StructType(fields)**

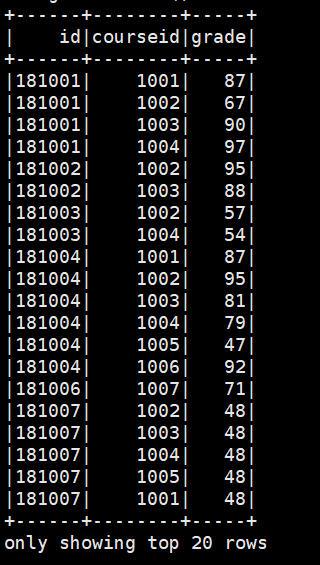
**grade1=spark.sparkContext.textFile("file:///root/test/grade.txt").map(lambda x:x.split(","))**

**grade2=grade1.map(lambda p:Row(p[0],p[1],int(p[2])))**

**grade3=spark.createDataFrame(grade2,grade)**

**grade3.show()**

**grade3.createOrReplaceTempView("grade")**



**1、查询181005学号学生的姓名和年龄；**

**spark.sql("select name,age from student where id='181005'").show()**

**2、查询孙慧选修课程的课号和成绩；（嵌套查询）**

**spark.sql("select courseid,grade from grade where id in (select id from student where name='孙慧')").show()**

**3、查询选修大学物理课程的学分、课号和成绩；（连接查询）**

**spark.sql("select credit,a.courseid,grade from grade a join course b on a.courseid=b.courseid where coursename='大学物理'").show()**

**4、查询选修离散数学课程的学号、姓名、课号。（内连接查询）**

**spark.sql("select c.id,c.name,d.courseid from student c inner join (select a.id,b.courseid from grade a inner join (select courseid from course where coursename='离散数学') b on a.courseid=b.courseid) d on c.id=d.id").show()**

**简答题**

**比较Spark和Hadoop的区别**

Hadoop中mapreduce计算框架的缺点：

1.表达能力有限

2.磁盘I/O开销大

3.延迟高

Spark在借鉴mapreduce优点的同时，很好的解决了mapreduce所面临的问题，主要具有以下优点：

1.Spark的计算模式也属于MapReduce，但不局限于Map和Reduce操作，还提供了多种数据集操作类型，编程模型比MapReduce更灵活。

2.Spark提供了内存计算，中间结果直接存放到内存中，带来了更高的迭代运算效率。

3.Saprk基于DAG的任务调度执行机制，要优于MapReduce的迭代执行机制

**简述Spark运行架构的特点**

1.每个应用都有自己专属的Executor进程，并且该进程在应用运行期间一直驻留

2.Spark运行过程与资源管理器无关，只要能够获取executor进程并保持通信即可。

3.Executor上有一个BlockManger存储模块，在处理迭代计算任务时可以直接将中间结果放在这个存储系统上，后续有需要可以直接读取，在交互式查询场景下也可以把表提前缓存到这个存储系统上，提高了读写I/O性能。

4.任务采用了数据本地性和推测执行等优化机制，而且spark采用了延时调度机制可以在更大的程度上实现执行过程优化。

**简述Spark的生态系统组成**

Spark的生态系统主要由以下组件组成：

Spark Core：包含了spark最基础和最核心的功能

Spakr SQL：用于结构化数据处理

Spark Streaming：是一种流计算框架

Structured Streaming：是一种基于spark SQL引擎构建的、可扩展且容错的流处理引擎

MLlib：提供了常用机器学习算法的实现

GraphX：是spark中用于图计算的API

**简述大数据时代到来的原因**

首先，信息技术和计算机网络技术的发展，为大数据时代的到来奠定了坚实的技术基础。 其次hadoop、spark、flink、beam等各种代表性技术的发展，使得数据的来源、类型、数量变得越来越多，由此产生出了的数目庞大且不断急剧增长的非结构化数据、半结构化数据，整个社会发展进入到了大数据时代

**简述RDD的特性**

1.高容错性，且大大降低了数据密集型应用中的容错开销

2.中间结果持久化到内存，避免了不必要的读写磁盘的开销

3.存放的数据可以是java对象，避免不必要的对象序列化和反序列化开销

**简述Spark的部署方式**

1、local模式：是单机模式，常用于本地开发测试

2、standalone模式：spark本身自带了完整的资源调度管理服务，可以独立部署到一个集群中，而不需要其他系统来为其提供资源管理调度服务

3、Spark on Mesos模式：mesos是一种资源调度管理框架，可以为运行在它上面的spark提供服务，目前官方推荐采用这种模式，所以许多公司在实际应用中也采用该模式

4、spark on YARN模式：spark可运行于YARN之上，与hadoop进行统一部署，资源管理和调度依赖YARN，分布式存储系统则依赖HDFS。

**RDD应用题题型**

1. **求图书的平均销量**

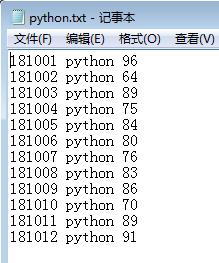
**现有一组键值对（“spark”，2），（“hadoop”，6），（“hadoop”，4），（“spark”，6），键值对的key表示图书名称，value表示某天的图书销量，现在需要计算每个键对应的平均值，也就是计算每种图书的平均销量**

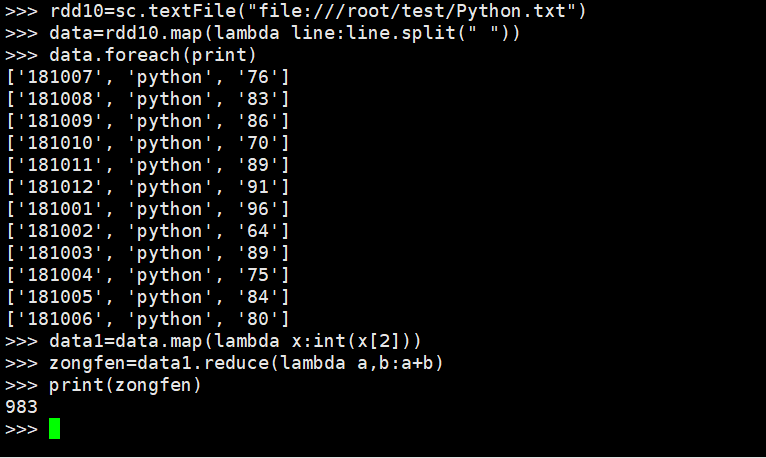
**rdd=sc.parallelize([("spark",2),("hadoop",6),("hadoop",4),("spark",6)])**

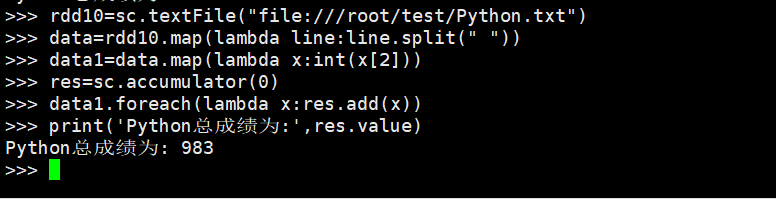
**rdd.mapValues(lambda x:(x,1)).reduceByKey(lambda x,y:(x[0]+y[0],x[1]+y[1])).mapValues(lambda x:x[0]/x[1]).collect()**

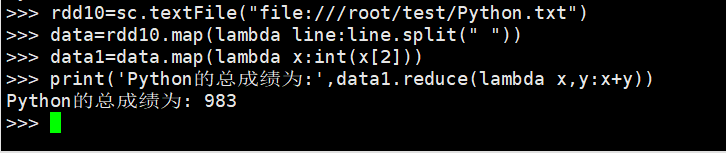
1. **文件排序参考总复习第五大题，教材P86-P89**
2. **求总分**

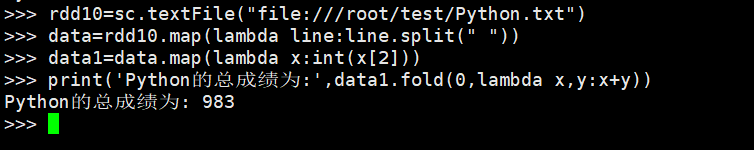
**通过RDD转换操作计算python课程的总分**

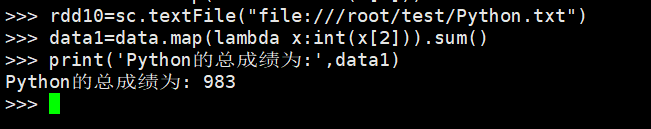
****





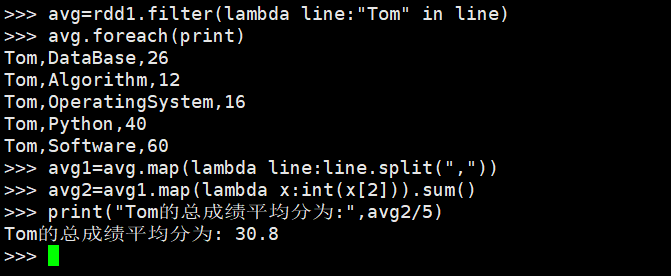




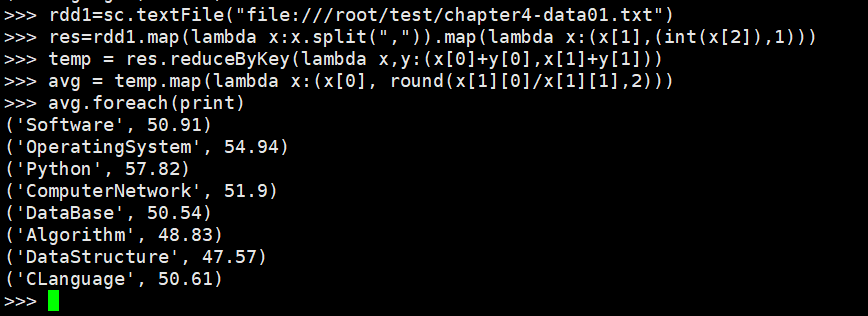


1. **求平均分**

**Tom同学的总成绩平均分**



**各门课程的平均分**



主要可以参考课本95页实验的题目类型

**Spark sql题型**

**主要就是生成表然后进行查询的完整过程，可以参考总复习的第六大题**