

# Spark 编程基础

## 选择题 (10 题 10 分)

- 1、spark 的四大组件下面哪个不是(D)  
A Spark Streaming B MLlib C Graph X D Spark R
- 2、Task 运行在下来哪里个选项中 Executor 上的工作单元 (C)  
A Driver program B spark master  
C worker node D Cluster manager
- 3、ClusterManager 是(A)  
A 主节点 B 从节点 C 执行器 D 上下文
- 4、下面哪个不是 RDD 的特点(C)  
A 可分区 B 可序列化 C 可修改 D 可持久化
- 5、Stage 的 Task 的数量由什么决定 (A)  
A Partition B Job C Stage D TaskScheduler
- 6、Spark 的集群部署模式不包括(D)  
A standalone B spark on mesos C spark on YARN D Local
- 7、下面哪个操作是窄依赖(B)  
A join B filter C group D sort
- 8 下面哪个操作肯定是宽依赖(C)  
A map B flatMap C reduceByKey D sample
- 9、下列哪个不是 RDD 的缓存方法(C)

A persist() B Cache() C Memory() D 以上都不是

10 Spark 默认的存储级别(A)

A MEMORY\_ONLY B MEMORY\_ONLY\_SER

C MEMORY\_AND\_DISK D MEMORY\_AND\_DISK\_SER

11、DataFrame 和 RDD 最大的区别(B)

A 科学统计支持 B 多了 schema

C 存储方式不一样 D 外部数据源支持

12、Spark Job 默认的调度模式(A)

A FIFO B FAIR C 无 D 运行时指定

13、下面哪个端口不是 spark 自带服务的端口(C)

A 8080 B 4040 C 8090 D 18080

14、Spark RDD 中没有的特性是(D)

A 位置优先 B 分布式 C 弹性 D 固定大小

15、hive 的元数据存储的在 derby 和 mysql 中有什么区别(B)

A 没区别 B 多会话 C 支持网络环境 D 数据库的区别

16、Hadoop 生态系统中用于构建数据仓库并允许用户输入 SQL 语句进行查询的功能组件是 (C)

A Flume B Pregel C Hive D Spark

17、大数据技术及其代表性的软件种类很多，不同的技术有其不同应用场景，都对应着不同的数据计算模式，请问软件产品 Pregel 主要应用于 (B) 计算模式

A 流计算 B 图计算 C 查询分析计算 D 批处理计算

18、以下操作中，哪个不是 DataFrame 的常用操作 (D)

A printSchema() B select() C filter() D sendto()

19、要把一个 DataFrame 保存到 people.json 文件中，下面语句哪个是正确的 (A)

A df.write.json("people.json")

B df.json("people.json")

C df.write.format("csv").save("people.json")

D df.write.csv("people.json")

20、val rdd=sc.parallelize(Array(1,2,3,4,5)) rdd.take(3) 的执行结果是 (A)

A Array(1,2,3) B Array(2,3,4) C 3 D 6

## 简答题 (3 题 15 分)

1. 比较 spark 和 hadoop 的区别? (8p)

答：主要有如下四点区别：①**编程方式**：Hadoop 的 MapReduce 在计算数据时，必须要转化为 Map 和 Reduce 两个过程，而 Spark 的计算模型不局限于 Map 和 Reduce 操作，还提供了**多种数据集的操作类型**，编程模型比 MapReduce 更加灵活。②**数据存储**：Hadoop 的 MapReduce 进行计算时，每次产生的中间结果都是存储在本地磁盘中，而 Spark 在计算时产生的中间结果存储在内存中。③**数据处理**：Hadoop 在每次执行数据处理时，都需要从磁盘中加载数据，导致磁盘的 I/O 开销较大，而 Spark 在执行数据处理时，只

需要将数据加载到内存中，之后直接在内存中加载中间结果数据集即可，减少了磁盘的 10 开销。④数据容错：MapReduce 计算的中间结果数据保存在磁盘中，并且 Hadoop 框架底层实现了备份机制，从而保证了数据容错；同样 Spark RDD 实现了基于 Lineage 的容错机制和设置检查点的容错机制，弥补了数据在内存处理时断电丢失的问题。在 Spark 与 Hadoop 的性能对比中，较为明显的缺陷是 Hadoop 中的 MapReduce 计算延迟较高，无法胜任当下爆发式的数据增长所要求的实时、快速计算的需求。

2. 简述 spark 架构运行的特点？（23p）

答：具有以下四点。①每个进程都有自己的专属 Executor 进程，且该进程会在程序运行期间一直驻留。②Spark 运行过程与资源管理器无关，只要能获取 Executor 进程并保存通信。③Executor 上有个 BlockManager 储存模块，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。

3. 简述 spark 的生态系统组成？（20p）

答：Spark 的生态系统组成有 Spark Core、Spark SQL、Spark Streaming、Structured Streaming、MLlib、GraphX 等。

4. 简述大数据时代到来的原因？

答：①移动互联网普及之后，智能设备将用户数据上传终端，形成了大量的用户行为数据。②电子导航如百度、高德出现后，产生大量的数据流数据。③进入新媒体时代后，互联网的数据和信息主要由数据制造，大量的用户在各大新媒体上产生的行为数

据。④线上交易积累大量数据。⑤电商平台的崛起产生大量的网上交易数据。⑥传统的互联网入口转向搜索引擎之后，用户的搜索行为和提问行为聚集海量数据。综上，大数据时代随着发展是必然的到来。

5. 简述 rdd 的特性？（26p）

答：Rdd 主要由三个特性：高容错性、中间结果持久化到内存、存放对象可以是 Java 对象。

6. 简述 saprk 的部署方式？（33p）

答：Saprk 的部署方式有以下三种：Standalone 模式、Spark on Mesos 模式、Spark on YARN 模式。

## 操作题

1、在 hdfs 中新建目录 test，并通过命令查看创建结果

```
root@master ~# hdfs dfs -mkdir /tmp/test
root@master ~# hdfs dfs -ls /tmp
22/06/06 07:06:00 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java-only versions to prevent unnecessary native lib loading
root supergroup          0 2020-08-27 08:19 /tmp/hadoop-yarn
root supergroup          0 2020-11-15 08:04 /tmp/hive
root supergroup          0 2022-06-06 07:00 /tmp/test
```

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

```
[root@master ~]# hdfs dfs -put /root/test/test1.txt /spark/myspark/
22/06/06 07:06:22 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java-only versions to prevent unnecessary native lib loading
[root@master ~]# hdfs dfs -ls /spark/myspark/
22/06/06 07:06:32 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java-only versions to prevent unnecessary native lib loading
Found 8 items
-rw-r--r--  3 root supergroup          51 2022-04-15 03:53 /spark/myspark/d1.txt
-rw-r--r--  3 root supergroup       123 2022-04-26 07:22 /spark/myspark/file1.txt
-rw-r--r--  3 root supergroup       203 2022-04-22 03:28 /spark/myspark/python.txt
-rw-r--r--  3 root supergroup       186 2022-04-15 03:09 /spark/myspark/result_math.txt
-rw-r--r--  3 root supergroup       179 2022-04-15 03:55 /spark/myspark/student.txt
-rw-r--r--  3 root supergroup         45 2022-06-06 07:06 /spark/myspark/test1.txt
-rw-r--r--  3 root supergroup         45 2022-04-10 01:06 /spark/myspark/word.txt
-rw-r--r--  3 root supergroup       174 2022-04-22 03:27 /spark/myspark/words.txt
```

3、在 hdfs 上查看 test1.txt 文件的内容，通过命令查看结果

```
[root@master ~]# hdfs dfs -cat /spark/myspark/test1.txt
22/06/06 07:07:41 WARN util.NativeCodeLoader: Unable to load nat
hadoop is good
spark is fast
spark is better
[root@master ~]#
```

- 4、 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

```
[root@master ~]# hdfs dfs -get /spark/myspark/test1.txt /usr/local/
22/06/06 07:09:55 WARN util.NativeCodeLoader: Unable to load native-hadoop
[root@master ~]# ls /usr/local/ | grep test1.txt
[root@master ~]# ls /usr/local/ | grep test1.txt
test1.txt
```

- 5、 删除 hdfs 上的 test1.txt 文件

```
[root@master ~]# hdfs dfs -rmr /spark/myspark/test1.txt
rmr: DEPRECATED: Please use 'rm -r' instead.
22/06/06 07:13:35 WARN util.NativeCodeLoader: Unable to load native
rmr: '/spark/myspark/test1.txt': No such file or directory
```

- 6、 删除 hdfs 上创建的目录 myspark，通过命令查看结果

```
root supergroup 174 2022-04-22 0
hdfs dfs -rm -rf /spark/myspark/
```

- 7、 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)
[Stage 0:>
1 李正明
```

- 8、 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 /spark/myspark/
```

```
rdd2 = sc.textFile("hdfs://master:9000/spark/myspark/student.txt")
rdd2.foreach(print)
1 李正明
```

- 9、 将 10 以内的奇数生成列表并转换为 rdd3，并显示查看

```
rdd3 = sc.parallelize([i for i in range(1,11,1) if i%2 == 1])
rdd3.foreach(print)
```

- 10、 将本地磁盘/root/test 下的 word.txt 转换成 rdd4，并打印输出其结果后保存为 write 文件

```

rdd4 = sc.textFile("file:///root/test/word.txt")
rdd4.foreach(print)
hadoop is good
park is fast
park is better
hadoop is basics
java also good
python is nosql
sql is relational database
god is nosql
relational database or nosql is good
rdd4.saveAsTextFile("file:///test/write")

```

- 11、 将文件 word.txt 转换的 rdd4 使用 map 转换，使用空格分离转换，并查看其结果

```

rdd4.map(lambda x: x.split(" ")).foreach(print)
hadoop, is, good
park, is, fast
park, is, better
python, is, basics
java, also, good

```

- 12、 将文件 word.txt 转换的 rdd4 使用 flatMap 转换，使用空格分离转换，并查看其结果

```

>>> rdd4.flatMap(lambda x: x.split(" ")).foreach(print)
hadoop
mysql
is
good

```

- 13、 使用 groupByKey() 转换对 rdd4 进行转换，并查看结果

```

>>> rdd4.flatMap(lambda x: x.split(" ")).map(lambda x: (x,1)).groupByKey().foreach(print)
hadoop', <pyspark.resultiterable.ResultIterable object at 0x7f30198b40b8>)
spark', <pyspark.resultiterable.ResultIterable object at 0x7f30198b40b8>)
fast', <pyspark.resultiterable.ResultIterable object at 0x7f30198b4438>)
better', <pyspark.resultiterable.ResultIterable object at 0x7f30198b4a58>)
is', <pyspark.resultiterable.ResultIterable object at 0x7f30198b4a58>)
also', <pyspark.resultiterable.ResultIterable object at 0x7f30198b4438>)
good', <pyspark.resultiterable.ResultIterable object at 0x7f30198b40b8>)

```

- 14、 使用 reduceByKey() 转换对 rdd4 进行求词频，并查看结果

```

>>> rdd4.flatMap(lambda x: x.split(" ")).map(lambda x: (x,1)).reduceByKey(lambda a,b:a+b).foreach(print)
spark', 2)
fast', 1)
better', 1)
also', 1)
hbase', 1)
mysql', 1)

```

- 15、 将文件 student.txt 转换成 RDD，并使用持久化对行进行操作 count(), first(), collect(), take(n) 查看结果

```

rdd5.cache()
e:///root/test/student.txt MapPartitionsRDD[78] at textFile at NativeMethodAccessor
print(rdd5.count(), rdd5.first(), rdd5.collect(), rdd5.take(3))
1001 李正明 ['1001\t李正明', '1002\t王 磊', '1003\t陈志华', '1004\t张永丽', '1005\t李正明', '1006\t王 磊', '1007\t陈志华', '1008\t张永丽', '1009\t李正明', '1010\t王 磊', '1011\t陈志华', '1012\t张永丽']

```

- 16、 7、将 test1.txt 文档转换成 rdd，并设置为 3 个分区，命

名为 rdd6，显示分区个数；重新设置分区个数为 2，并显示。

```
(12) root@ts-189:~#  
>>> rdd6 = sc.textFile("file:///root/test/test1.txt", 3)  
>>> len(rdd6.glom().collect())  
3  
>>> len(rdd6.repartition(2).glom().collect())  
2
```

17、 8、在 rdd4 中筛选出包含“spark”的字符串保存到新的 rdd7 中，并打印显示

```
root@ts-189:~#  
>>> rdd7 = rdd4.filter(lambda line: "spark" in line)  
>>> rdd7.foreach(print)  
spark is fast
```

18、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

```
08-08 08:18:43 WARN ObjectStore:org.apache.hadoop.hive.metastore.ObjectStore - Failed to get database global_temp, returning database default_database  
df2_json = spark.read.format("json").load("file:///root/test/people.json")  
df1_json = spark.read.json("file:///root/test/people.json")  
df1_txt = spark.read.text("file:///root/test/people.txt")  
df1_txt = spark.read.format("text").load("file:///root/test/people.txt")  
df1_csv = spark.read.csv("file:///root/test/11.csv")  
df2_csv = spark.read.format("csv").load("file:///root/test/11.csv")
```

19、将已经创建好的 DataFrame 保存成 newpeople.json，newpeople.txt，并查看保存成功与否

```
root@ts-189:~#  
> df2_json.write.json("file:///test/newpeople.json")  
> df1_txt.write.format("text").save("file:///test/newpeople.txt")  
>
```

## Rdd 应用题

1、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

```
>>> rdd_file1 = sc.textFile("file:///root/test/file1.txt")
>>> rdd_file2 = sc.textFile("file:///root/test/file2.txt")
>>> rdd_file = rdd_file1.union(rdd_file2).distinct()
>>> rdd_file.map(lambda x: x.split(",")).map(lambda x: (int(x[2]),1)).repartition(1).sortByKey(False).take(5)
[(7390, 1), (793, 1), (541, 1), (498, 1), (351, 1)]
>>>
```

## spark sql 应用题

1、/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

准备阶段:

student:id,name,sex,age;

course:c\_id,c\_name,credit,nknow

grade:id,c\_id,fraction

```
>>> from pyspark.sql import Row
>>> from pyspark.sql.types import *
>>> student = spark.sparkContext.textFile("file:///root/test/student.txt").map(lambda line: line.split(",")).map(lambda p: Row(id=p[0], name=p[1], sex=p[2], age=p[3]))
>>> sch_s = spark.createDataFrame(student)
>>> sch_s.createOrReplaceTempView("student")
>>> schemaString = "id c_id grade"
>>> fields = [StructField(field_name, StringType(), True) for field_name in schemaString.split(" ")]
>>> schema = StringType(fields)
>>> grade = spark.sparkContext.textFile("file:///root/test/grade.txt").map(lambda x: x.split(",")).map(lambda p: Row(p[0], p[1], p[2]))
>>> sch_g = spark.createDataFrame(grade, schema)
>>> sch_g.createOrReplaceTempView("grade")
>>> course = spark.sparkContext.textFile("file:///root/test/course.txt").map(lambda x: x.split(",")).map(lambda p: Row(c_id=p[0], d_name=p[1], credit=p[2], nknow=p[3]))
>>> sch_c = spark.createDataFrame(course, schema)
>>> sch_c.createOrReplaceTempView("course")
```

① 查询 181005 学号学生的姓名和年龄;

```
>>> df1 = spark.sql("select name, age from student where id='181005'")
>>> df1.show()
+-----+
|name|age|
+-----+
| 马明| 20|
+-----+
```

② 查询孙慧选修课程的课号和成绩; (嵌套查询)

```
>>> df2 = spark.sql("select c_id, fraction from grade where id = (select id from student where name='孙慧')")
>>> df2.show()
+-----+
|c_id|fraction|
+-----+
|09012030| 73|
|09011020| 51|
+-----+
```

③ 查询选修大学物理课程的学分、课号和成绩; (连接查询)

```

2  SELECT course.credit, grade.c_id, grade.fraction FROM course,grade
   WHERE course.c_id=grade.c_id AND course.c_name='大学物理'
3  |

```

信息	摘要	结果 1	剖析	状态
		credit	c_id	fraction
▶	3	3021020	90	
	3	3021020	100	
	3	3021020	94	

④ 查询选修离散数学课程的学号、姓名、课号。(内连接查询)

```

3  SELECT a.id, a.`name`,b.c_id FROM student a INNER JOIN grade c on a.id=
   c.id INNER JOIN course b on b.c_id=c.c_id AND b.c_name='离散数学'

```

信息	摘要	结果 1	剖析	状态
		id	name	c_id
▶	181004	单丹丹	8012010	
	181006	李薇	8012010	
	181010	安宇杰	8012010	