

第4章Spark编程进阶实训

第4章Spark编程进阶实训

CH04实训1：竞赛网站访问日志分析

训练要点

需求说明

实现思路及步骤

实训作业要求

实现参考

准备数据

编码参考

打包上传

集群执行

预期结果

HDFS上结果

视频学习

CH04实训2：自定义分区器实现按人物标签进行数据分析

训练要点

需求说明

实现思路及步骤

实训作业要求

实现参考

准备数据

编码参考

打包上传

集群执行

预期结果

HDFS上结果

CH04实训1：竞赛网站访问日志分析

训练要点

1. 搭建Spark工程环境。
2. Spark编程。
3. 通过spark-submit提交应用。

需求说明

某竞赛网站每年都会开展数据挖掘的竞赛，在竞赛期间网站会有大量人群访问，生成了大量的用户访问记录。现在提供2016年10月到2017年6月的部分脱敏访问日志数据文件:jc_content_viewlog.txt。日志数据的基本内容如图所示，仅提供以下6个字段：

| 属性名称 | 属性解析 |
|------------|--------|
| Id | 序号 |
| Content_id | 网页ID |
| Page_path | 网址 |
| Userid | 用户ID |
| Sessionid | 缓存生成ID |
| Date_time | 访问时间 |

要求根据提供的用户访问日志数据，利用Spark技术统计访问的用户数、被访问的不同网页个数以及每月的访问量，并将结果保存到HDFS上。

实现思路及步骤

1. 配置好Spark的IntelliJ IDEA开发环境。
2. 启动IntelliJ IDEA，并进行Spark编程。
3. 读取文件：/user/myname/jc_content_viewlog.txt，对访问记录中的网页去重，统计本周期内被访问网页的个数。
4. userid为用户注册登录的标识，对userid去重，统计登录用户的数量。
5. 按月统计访问记录数。
6. 将结果保存到三个文件中：
 - 1) /user/myname/out_wy_count
 - 2) /user/myname/out_user_count
 - 3) /user/myname/out_nycount。
7. 打包Spark工程，在集群使用spark-submit提交应用程序：spark-submit --master spark://master:7077

实训作业要求

- 1, 截图IDEA上的源码
- 2, 截图crt上运行的结果
- 3, 截图在hdfs上的运行结果

实现参考

准备数据

```
1 cd /root/spark
2 wget https://biglab.site//b59510spark/file/jc_content_viewlog.tar.gz
3 tar -xzvf ./jc_content_viewlog.tar.gz
4 hdfs dfs -rm /user/myname/jc_content_viewlog.txt
5 hdfs dfs -put ./jc_content_viewlog.txt /user/myname/
```

编码参考

```
1 package chap04t
2
3 import org.apache.spark.rdd.RDD
4 import org.apache.spark.{SparkConf, SparkContext}
5 object LogCount {
6     def main(args: Array[String]): Unit = {
7
8         if (args.length < 2) {
9             println("请指定input和output")
10            System.exit(1) //非0表示非正常退出程序
11        }
12
13        //TODO 1.env/准备sc/SparkContext/Spark上下文执行环境
14        val conf: SparkConf = new
15        SparkConf().setAppName("wc").setMaster("local")
16        val sc: SparkContext = new SparkContext(conf)
17        sc.setLogLevel("WARN")
18
19        //TODO 2.source/读取数据
20        //RDD:A Resilient Distributed Dataset (RDD):弹性分布式数据集,简单理解为分布式
21        集合!使用起来和普通集合一样简单!
22        //RDD[就是一行行的数据]
23        val logs_all: RDD[Array[String]] = sc.textFile(args(0)).map {
24            _.split(",")
25        }
26        //TODO 3.transformation/数据操作/转换
27        //对访问记录中的网页去重,统计本周期内被访问网页的个数
28        //
29        478896,1043,/jszx/1043.jhtml,14884,F6D362B9AFAC436D153B7084EF3BA332,2017-03-
30        01 00:23:07
31        val wy_log: RDD[String] = logs_all.map(x => (x(1).toString)).distinct()
32        val wy_count: RDD[(String, Int)] = wy_log.map(("wy_zs",
33        _)).groupByKey().map(x => (x._1, x._2.size))
34        //userid为用户注册登录的标识,对userid去重,统计登录用户的数量
35        val user_log: RDD[String] = logs_all.map(x =>
36        (x(3).toString)).distinct()
37        val user_count: RDD[(String, Int)] = user_log.map(("user_zs",
38        _)).groupByKey().map(x => (x._1, x._2.size))
39        //按月统计访问记录数
40        val logs_all_new = logs_all.map { x => (x(0), x(1), x(2), x(3), x(4),
41        x(5), date_time(x(5))) }
42        val ny_count: RDD[(String, Int)] = logs_all_new.map(x => (x._7,
43        1)).reduceByKey((a, b) => a + b)
44
45        //TODO 4.sink/输出
46        //输出到指定path(可以是文件/夹)
47        println("be writing to :" + args(1))
48        wy_count.repartition(1).saveAsTextFile(args(1))
49        println("be writing to :" + args(2))
50        user_count.repartition(1).saveAsTextFile(args(2))
51    }
52}
```

```

43     println("be writing to :" + args(3))
44     ny_count.repartition(1).saveAsTextFile(args(3))
45     //为了便于查看web-UI可以让程序睡一会
46     Thread.sleep(1000 * 60)
47
48     //TODO 5.关闭资源
49     sc.stop()
50 }
51
52 //获取年月，时间段作为输入参数
53 def date_time(date: String): String = {
54     if (date.trim.length == "2017-03-01 00:23:07".length) {
55         val nianye = date.trim.substring(0, 7)
56         nianye
57     }
58     else
59         "1900-01"
60 }
61
62 }
63

```

打包上传

1. 在项目菜单中编译生成artifact: 选择菜单栏中的“Build->Build Artifacts->word->Build,命令
2. 将生成的word.jar上传到集群slave1的/root/spark目录下

集群执行

```

1 cd /root/spark
2 hdfs dfs -rm -r /user/myname/out_wy_count
3 hdfs dfs -rm -r /user/myname/out_user_count
4 hdfs dfs -rm -r /user/myname/out_ny_count
5 spark-submit --master spark://master:7077 --class chap04t.LogCount \
6 /root/spark/word.jar \
7 /user/myname/jc_content_viewlog.txt \
8 /user/myname/out_wy_count \
9 /user/myname/out_user_count \
10 /user/myname/out_ny_count

```

注意：spark-submit后连续6行是要合在一起执行，它是一条命令

预期结果

```

1 [root@slave1 spark]# cd /root/spark
2 [root@slave1 spark]# hdfs dfs -rm -r /user/myname/out_wy_count
3 hdfs dfs -rm -r /user/myname/out_user_count
4 hdfs dfs -rm -r /user/myname/out_ny_count
5 spark-submit --master spark://master:7077 --class chap04t.LogCount \
6 /root/spark/word.jar \
7 /user/myname/jc_content_viewlog.txt \
8 /user/myname/out_wy_count \
9 /user/myname/out_user_count \
10 /user/myname/out_ny_count

```

```

11 Deleted /user/myname/out_wy_count
12 [root@slave1 spark]# hdfs dfs -rm -r /user/myname/out_user_count
13 Deleted /user/myname/out_user_count
14 [root@slave1 spark]# hdfs dfs -rm -r /user/myname/out_ny_count
15 Deleted /user/myname/out_ny_count
16 [root@slave1 spark]# spark-submit --master spark://master:7077 --class
    chap04t.LogCount \
17 > /root/spark/word.jar \
18 > /user/myname/jc_content_viewlog.txt \
19 > /user/myname/out_wy_count \
20 > /user/myname/out_user_count \
21 > /user/myname/out_ny_count
22 SLF4J: Class path contains multiple SLF4J bindings.
23 SLF4J: Found binding in [jar:file:/usr/local/spark/jars/slf4j-log4j12-
    1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
24 SLF4J: Found binding in [jar:file:/usr/local/hadoop-
    3.3.1/share/hadoop/common/lib/slf4j-log4j12-
    1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
25 SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an
    explanation.
26 SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
27 be writing to :/user/myname/out_wy_count
28 be writing to :/user/myname/out_user_count
29 be writing to :/user/myname/out_ny_count
30
31 [root@slave1 spark]#

```

HDFS上结果

结果有三个输出文件：

| Permissions | Owner | Group | Size | Created | Replication | Block Size | Name | Actions |
|-------------|-------|------------|-----------|--------------|-------------|------------|-----------------------------|---------|
| drwxr-xr-x | root | supergroup | 0 B | Sep 24 14:56 | 0 | 0 B | bigDataOutPut12 | |
| -rw-r--r-- | root | supergroup | 74.06 MB | Sep 27 2022 | 3 | 128 MB | bigdata.hddly.cn-access_log | |
| -rw-r--r-- | root | supergroup | 92 B | Sep 15 17:43 | 3 | 128 MB | bigdata.txt | |
| drwxr-xr-x | root | supergroup | 0 B | Oct 09 2022 | 0 | 0 B | class | |
| drwxr-xr-x | root | supergroup | 0 B | Sep 15 17:23 | 0 | 0 B | csv_out | |
| drwxr-xr-x | root | supergroup | 0 B | Sep 24 12:42 | 0 | 0 B | csv_out1 | |
| drwxr-xr-x | root | supergroup | 0 B | Oct 03 03:13 | 0 | 0 B | csv_out1112 | |
| drwxr-xr-x | root | supergroup | 0 B | Sep 24 14:54 | 0 | 0 B | csv_out12 | |
| drwxr-xr-x | root | supergroup | 0 B | Sep 27 2022 | 0 | 0 B | gujia_zf5 | |
| -rw-r--r-- | root | supergroup | 15.02 MB | Oct 18 01:23 | 3 | 128 MB | jc_content_viewlog.txt | |
| drwxr-xr-x | root | supergroup | 0 B | Sep 15 15:18 | 0 | 0 B | json_out | |
| drwxr-xr-x | root | supergroup | 0 B | Sep 24 12:38 | 0 | 0 B | json_out1 | |
| drwxr-xr-x | root | supergroup | 0 B | Oct 03 00:19 | 0 | 0 B | json_out111 | |
| drwxr-xr-x | root | supergroup | 0 B | Sep 24 14:49 | 0 | 0 B | json_out12 | |
| drwxr-xr-x | root | supergroup | 0 B | Sep 24 16:24 | 0 | 0 B | json_out13 | |
| drwxr-xr-x | root | supergroup | 0 B | Nov 03 2022 | 0 | 0 B | musics | |
| -rw-r--r-- | root | supergroup | 711.57 KB | Oct 18 00:20 | 3 | 128 MB | online_retail.txt | |
| drwxr-xr-x | root | supergroup | 0 B | Oct 18 01:52 | 0 | 0 B | out_ny_count | |
| drwxr-xr-x | root | supergroup | 0 B | Oct 18 01:52 | 0 | 0 B | out_user_count | |
| drwxr-xr-x | root | supergroup | 0 B | Oct 18 01:52 | 0 | 0 B | out_wy_count | |

Showing 1 to 25 of 60 entries

Previous 1 2 3 Next

其中out_ny_count:

File information - part-00000

Download [Head the file \(first 32K\)](#) [Tail the file \(last 32K\)](#)

Block information -- Block 0 ▾

Block ID: 1073829630
Block Pool ID: BP-1277059469-192.168.137.100-1662365964733
Generation Stamp: 88885
Size: 48
Availability:
• slave1
• slave2

File contents

```
(2017-03,46915)  
(2017-04,118470)  
(2017-05,6474)
```

Close

其中out_user_count:

File information - part-00000

Download [Head the file \(first 32K\)](#) [Tail the file \(last 32K\)](#)

Block information -- Block 0 ▾

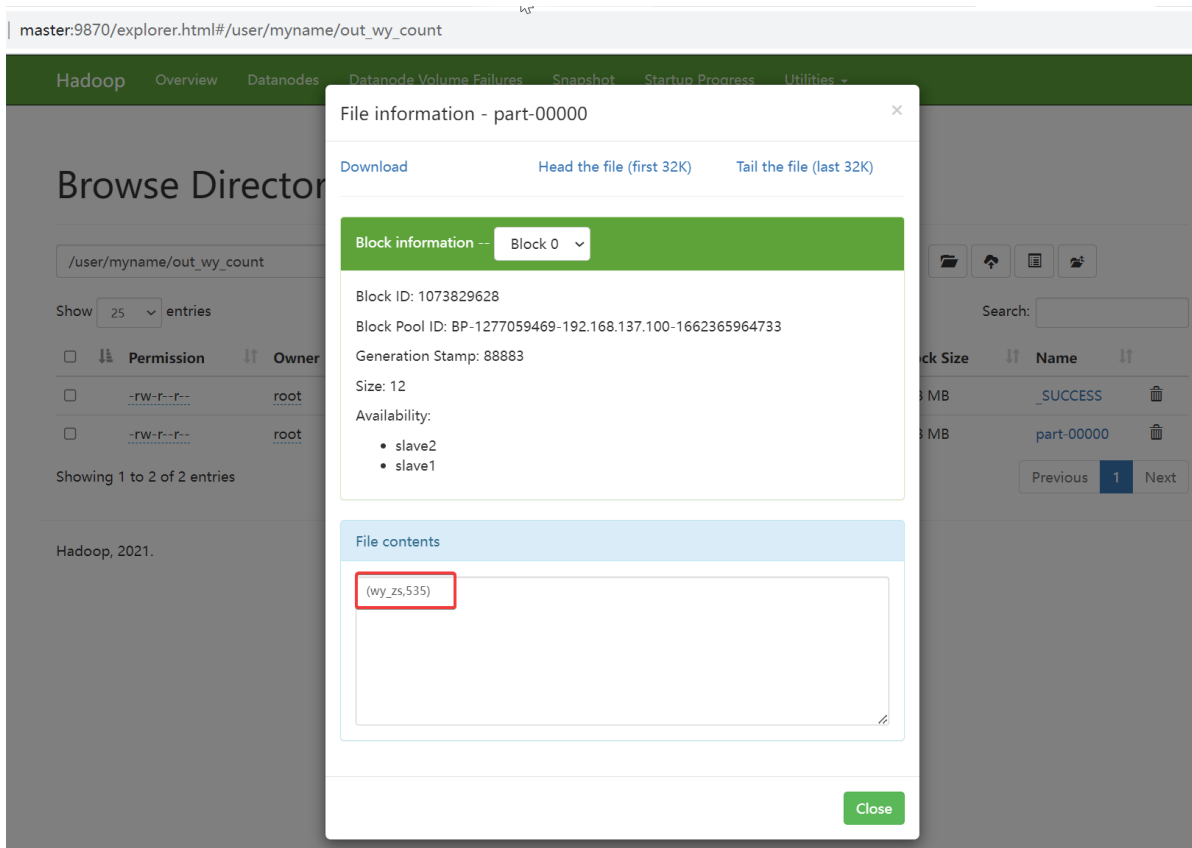
Block ID: 1073829629
Block Pool ID: BP-1277059469-192.168.137.100-1662365964733
Generation Stamp: 88884
Size: 15
Availability:
• slave2
• slave1

File contents

```
(user_zs,8422)
```

Close

其中out_wy_count:



视频学习

视频学习：

1. [Hive安装和配置视频](#)
2. [源码下载](#)
3. [竞赛网站访问日志分析视频](#) [补充视频2](#)
4. [如何减小hive和spark-shell的日志输出](#)
5. http://home.hddly.cn:50090//media/2023-11-03%2017.24.43%20biglab%209562804926/meeting_01.mp4

CH04实训2：自定义分区器实现按人物标签进行数据分析

训练要点

1. 掌握使用IntelliJ IDEA搭建spark开发环境
2. 掌握创建spark工程
3. 掌握使用spark自定义分区
4. 掌握打包spark工程
5. 掌握通过spark-submit提交应用

需求说明

中国女排曾多次夺得奥运会冠军，团结协作、顽强拼搏的女排精神始终代代相传，极大地激发了中国人的自豪之情、提高了自尊和增加了自信，为我们奋进在新征程上提供了强大的精神力量。现有一份某年度中国女排（包括国家女子排球队和国家青年女子排球队）的集训运动员数据文件Volleyball-Players.csv，包含4个数据字段，数据字段说明如表所示排球运动员按所属位置可分为5类，分别是主攻、接应、二传、副攻和自由人。为了解中国女排各运动员的所属位置，现要求在IntelliJ IDEA中进行spark编程，通过自定义分区实现将集训运动员数据按运动员所属位置进行分区，并将程序打包，通过spark-submit提交应用。设置5个分区，第1个分区保存位置标签为“主攻”的运动员数据，第2个分区保存

位置标签为“接应”的运动员数据，第3个分区保存位置标签为“二传”的运动员数据，第4个分区保存位置标签为“副攻”的运动员数据，第5个分区保存位置标签为“自由人”的运动员数据，将分区结果输出到HDFS上。

表:中国女排集训运动员数据字段说明

| 字段名称 | 说明 |
|--------------|---------------------------|
| Name | 中国女排集训运动员名称 |
| Birthday | 出生日期 |
| Height | 身高, 单位为cm (厘米) |
| Shot_Height | 扣球高度(cm) |
| Block_Height | 拦网高度(cm) |
| Position | 位置, 分为5类: 主攻、接应、二传、副攻和自由人 |
| Province | 位置,省、市 |

Volleyball-Players.csv文件内容如:

姓名,出生日期,身高(cm),扣球高度(cm),拦网高度(cm),位置,省、市

刘晓彤,1990.02.16,187,315,300,主攻,北京

曾春蕾,1989.03.11,187,315,312,接应,北京

实现思路及步骤

1. 配置好IntelliJ IDEA和Spark开发环境，并启动IntelliJ IDEA
2. 使用textFile()方法读取Volleyball-Players.csv数据以创建RDD，并设置分区数为5
3. 使用map()方法将读取的数据按“,” (逗号) 进行分割，筛选出"Position"和"Name" 字段，并转化成"(Position,Name)"的形式。
4. 自定义Mypartitioner，继承自定义partitioner类，重写partitioner类的numpartitions 和 getpartition()方法，实现自定义分区。
5. 在主函数中调用自定义分区类Mypartitioner
6. 打包spark工程，并将应用程序提交至集群运行。

实训作业要求

- 1, 截图IDEA上的源码
- 2, 截图crt上运行的结果
- 3, 截图在hdfs上的运行结果

实现参考

准备数据

在slave1下执行

```
1 rm -f spark_t_data.tar.gz*
2 wget https://biglab.site//b59510spark/file/spark_t_data.tar.gz
3 tar -xzvf ./spark_t_data.tar.gz
4 hdfs dfs -rm /user/myname/Volleyball_Players.csv
5 hdfs dfs -rm -r /user/myname/output_volleyball
6 hdfs dfs -put ./spark_t_data/Volleyball_Players.csv /user/myname/
```

编码参考

```
1 package chap04t
2
3 import org.apache.spark.{Partitioner, SparkConf, SparkContext}
4 import org.apache.spark.rdd.RDD
5 object Partition {
6   def main(args: Array[String]): Unit = {
7     if (args.length < 2) {
8       println("请指定input和output")
9       System.exit(1) //非0表示非正常退出程序
10    }
11    val sparkConf = new
SparkConf().setMaster("local").setAppName("Players_Partition")
12    val sc = new SparkContext(sparkConf)
13    val input: RDD[String] = sc.textFile(args(0),5)
14    // args(0) 或 "hdfs://master:9864/user/myname/Volleyball_Players.csv"
15    val rdd: RDD[(String, String)] = input.map(x => {
16      val data = x.split(",")
17      (data(5), data(0))
18    })
19    rdd.foreach(println)
20    val partRDD: RDD[(String, String)] = rdd.partitionBy(new MyPartitioner)
21    partRDD.saveAsTextFile(args(1))
22    // args(1) 或 "hdfs://master:9864/user/myname/output_volleyball"
23    sc.stop()
24  }
25
26  /**
27   * 自定义分区器
28   * 1.继承Partitioner
29   * 2.重写方法
30   *
31   */
32  class MyPartitioner extends Partitioner{
33    //分区数量
34    override def numPartitions: Int = 5
35    /// 根据数据的key值返回数据所在的分区索引（从0开始）
36    override def getPartition(key: Any): Int = {
37
38      key match{
39        case "主攻" =>0
40        case "接应" =>1
```

```

41         case "二传" =>2
42         case "副攻" =>3
43         case _ =>4
44     }
45 }
46 }
47
48 }
49

```

打包上传

1. 在项目菜单中编译生成artifact: 选择菜单栏中的“Build->Build Artifacts->word->Build,命令
2. 将生成的word.jar上传到集群slave1的/root/spark目录下

集群执行

```

1 cd /root/spark
2 hdfs dfs -rm -r /user/myname/output_volleyball
3 spark-submit --master spark://master:7077 --class chap04t.Partition \
4 /root/spark/word.jar \
5 /user/myname/Volleyball_Players.csv \
6 /user/myname/output_volleyball

```

注意: spark-submit后连续4行是要合在一起执行, 它是一条命令

预期结果

```

1 [root@slave1 spark]# cd /root/spark
2 [root@slave1 spark]# hdfs dfs -rm -r /user/myname/output_volleyball
3 spark-submit --master spark://master:7077 --class chap04t.Partition \
4 /root/spark/word.jar \
5 /user/myname/Volleyball_Players.csv \
6 /user/myname/output_volleyball
7 Deleted /user/myname/output_volleyball
8 [root@slave1 spark]# spark-submit --master spark://master:7077 --class
  chap04t.Partition \
9 > /root/spark/word.jar \
10 > /user/myname/Volleyball_Players.csv \
11 > /user/myname/output_volleyball
12 SLF4J: Class path contains multiple SLF4J bindings.
13 SLF4J: Found binding in [jar:file:/usr/local/spark/jars/slf4j-log4j12-
  1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
14 SLF4J: Found binding in [jar:file:/usr/local/hadoop-
  3.3.1/share/hadoop/common/lib/slf4j-log4j12-
  1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
15 SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an
  explanation.
16 SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
17 (位置,姓名)
18 (主攻,刘晓彤)
19 (接应,曾春蕾)
20 (二传,姚迪)
21 (主攻,李盈莹)

```

- 22 (副攻,王媛媛)
- 23 (自由人,孟子旋)
- 24 (二传,丁霞)
- 25 (副攻,颜妮)
- 26 (副攻,胡铭媛)
- 27 (二传,孙海平)
- 28 (主攻,段放)
- 29 (主攻,张轶婵)
- 30 (主攻,张常宁)
- 31 (接应,龚翔宇)
- 32 (二传,刁琳宇)
- 33 (自由人,倪非凡)
- 34 (接应,吴晗)
- 35 (自由人,林莉)
- 36 (副攻,郑益昕)
- 37 (自由人,王梦洁)
- 38 (副攻,杨涵玉)
- 39 (二传,蔡雅倩)
- 40 (主攻,杜清清)
- 41 (主攻,朱婷)
- 42 (副攻,袁心玥)
- 43 (主攻,刘晏含)
- 44 (副攻,高意)
- 45 (副攻,李学林)
- 46 (主攻,张景胤)
- 47 (二传,陈磊炆)
- 48 (副攻,缪阮彤)
- 49 (自由人,陈嘉杰)
- 50 (副攻,彭世坤)
- 51 (接应,陈禧龙)
- 52 (主攻,郭顺祥)
- 53 (二传,毛天一)
- 54 (主攻,袁党毅)
- 55 (接应,修成城)
- 56 (接应,唐川航)
- 57 (副攻,杜昊昱)
- 58 (自由人,马晓腾)
- 59 (接应,江川)
- 60 (主攻,张秉龙)
- 61 (副攻,王东宸)
- 62 (主攻,刘力宾)
- 63 (副攻,谷佳丰)
- 64 (副攻,陈龙海)
- 65 (二传,詹国俊)
- 66 (副攻,张哲嘉)
- 67 (主攻,戴卿尧)
- 68 (自由人,童嘉骅)
- 69 (二传,于垚辰)
- 70 (接应,刘向东)
- 71 (副攻,戴海波)
- 72 (副攻,饶书涵)
- 73 (主攻,季道帅)
- 74 (接应,王径一)
- 75 (副攻,张祖源)
- 76 (二传,李润铭)

77 (主攻,付侯文)

78 [root@slave1 spark]#

HDFS上结果

结果有一个输出目录和五个文件: /user/myname/output_volleyball

master:9870/explorer.html#/user/myname/output_volleyball

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Browse Directory

/user/myname/output_volleyball Go!

Show 25 entries Search:

| Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name |
|------------|-------|------------|-------|---------------|-------------|------------|------------|
| -rw-r--r-- | root | supergroup | 0 B | Oct 18 03:05 | 3 | 128 MB | SUCCESS |
| -rw-r--r-- | root | supergroup | 300 B | Oct 18 03:05 | 3 | 128 MB | part-00000 |
| -rw-r--r-- | root | supergroup | 166 B | Oct 18 03:05 | 3 | 128 MB | part-00001 |
| -rw-r--r-- | root | supergroup | 185 B | Oct 18 03:05 | 3 | 128 MB | part-00002 |
| -rw-r--r-- | root | supergroup | 338 B | Oct 18 03:05 | 3 | 128 MB | part-00003 |
| -rw-r--r-- | root | supergroup | 168 B | Oct 18 03:05 | 3 | 128 MB | part-00004 |

Showing 1 to 6 of 6 entries Previous 1 Next

其中文件之一: part_00000

master:9870/explorer.html#/user/myname/output_volleyball

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Browse Directory

/user/myname/output_volleyball

Show 25 entries

Showing 1 to 6 of 6 entries

Hadoop, 2021.

File information - part-00000

Download Head the file (first 32K) Tail the file (last 32K)

Block information -- Block 0

Block ID: 1073829638
Block Pool ID: BP-1277059469-192.168.137.100-1662365964733
Generation Stamp: 88893
Size: 300
Availability:

- slave2
- slave1

File contents

(主攻,刘晓彤)
(主攻,李盈莹)
(主攻,段放)
(主攻,张轶婵)
(主攻,张常宁)
(主攻,杜清清)
(主攻,朱婷)
(主攻,刘晏含)

Close

