

金融大数据处理技术 实验二（任务二： 星期交易量统计）

221275010 屈航

1、设计思路

1.1、Mapper的设计思路

Mapper 重构了一个 map 函数。

map 函数主要实现了对 任务一生成的 part-r-00000 中的统计一周七天中每天的平均资金流入与流出情况，并按照资金流入量从大到小排序。主要涉及到对于日期转化为星期几，日期转星期的过程可以用一个 Calendar 类来实现，实现的主要代码例如

calendar.setTime(dateFormat.parse(dateString)); 然后再把 calendar 类型转化为可以表征星期的 int 类型数字，代码为 int dayOfWeek = calendar.get(Calendar.DAY_OF_WEEK) - 1; // 转换为0（周日）到6（周六）这样就可以再通过 int 类型转化为星期的字符串。

因为考虑到求和得到的数据量可能很大，导致会超出 int 类型的表示范围，故实现的过程中的资金流入流出量的数据类型都是 BigInteger。

因为要传输两个值，即每日的 total_purchase_amt 资金流入与 total_redeem_amt 资金流出，故可以把这两个数据作为一个 BalanceWritable 类来作为 value 进行键值对的传输。

以下是 BalanceWritable 类的部分构造语句：

```
public static class BalanceWritable implements Writable {
    private BigInteger inflow;
    private BigInteger outflow;

    public BalanceWritable() {
        this.inflow = BigInteger.ZERO;
        this.outflow = BigInteger.ZERO;
    }

    public void set(BigInteger inflow, BigInteger outflow) {
        this.inflow = inflow;
        this.outflow = outflow;
    }

    @Override
    public void write(DataOutput out) throws IOException {
        out.writeUTF(inflow.toString());
        out.writeUTF(outflow.toString());
    }

    @Override
    public void readFields(DataInput in) throws IOException {
        inflow = new BigInteger(in.readUTF());
        outflow = new BigInteger(in.readUTF());
    }
}
```

以下是 `map` 函数的主要功能语句：

```
String[] fields = value.toString().split("\\t");
if (fields.length != 2)
    return;

String[] amounts = fields[1].split(",");
if (amounts.length != 2)
    return;

BigInteger inflow = new BigInteger(amounts[0]);
BigInteger outflow = new BigInteger(amounts[1]);

// int lineIndex = (int) (key.get() + 1);
// int dayIndex = (lineIndex - 1) % 7; // 0到6表示一周中的天
// String weekDay = getWeekDay(dayIndex);
String dateString = fields[0];
Calendar calendar = Calendar.getInstance();
try {
    calendar.setTime(dateFormat.parse(dateString));
} catch (ParseException e) {
    e.printStackTrace();
}
int dayOfWeek = calendar.get(Calendar.DAY_OF_WEEK) - 1; // 转换为0（周日）到6（周六）

String weekDay = getWeekDay(dayOfWeek);

weekKey.set(weekDay);
balancewritable.set(inflow, outflow);
context.write(weekKey, balancewritable);
```

注意：实验开始前要对 `user_balance_table.csv` 中的第一行删去，第一行并不是需要统计的内容。

1.2、Reducer的设计思路

Reducer 包含有 `reduce` 函数和 `cleanup` 函数。

1. `reduce` 函数就是实现对于同一个key的 `Iterable<Balancewritable> values` 中的每一个元素也就是统计数求平均，最后再将求平均的结果存入一个叫做 `TreeMap<BigInteger, String> sortedResults = new TreeMap<>(Comparator.reverseOrder())` 的类里面，这个类会针对键 `key`，此时的键是平均资金流入量，进行倒序排序，以待 `cleanup` 函数实现对输出格式的处理。

以下是 `reduce` 函数的主要功能语句：

```
BigInteger totalInflow = BigInteger.ZERO;
BigInteger totalOutflow = BigInteger.ZERO;
int count = 0;

for (Balancewritable val : values) {
    totalInflow = totalInflow.add(val.getInflow());
    totalOutflow = totalOutflow.add(val.getOutflow());
    count++;
}

BigInteger avgInflow = totalInflow.divide(BigInteger.valueOf(count));
```

```

BigInteger avgOutflow = totalOutflow.divide(BigInteger.valueOf(count));

// 使用 TreeMap 存储结果，按资金流入量排序
sortedResults.put(avgInflow, key.toString() + "," + avgOutflow);
}

```

2. `cleanup` 函数就是实现对输出结果的格式控制。对于 `reduce` 函数中实现的键为**平均资金流入量**，值为**String(星期, 平均流出量)**，这已经按照平均资金流入量完成了倒序排序了，所以只需要把键值对的3个值修改一下位置依次输出即可。

以下是 `cleanup` 函数的主要功能语句：

```

for (Map.Entry<BigInteger, String> entry : sortedResults.entrySet()) {
    String[] values = entry.getValue().split(","); // 分割星期和流入、流出
    String day = values[0].trim(); // 获取星期
    String out_num = values[1].trim();
    result.set(entry.getKey().toString() + "," + out_num);
    context.write(new Text(day), result);
}

```

1.3、项目运行的配置设计

- 此次项目主要使用 Maven 进行项目管理，通过编辑 `pom.xml` 文件对该项目进行配置。`pom.xml` 文件的配置信息包含有该项目需要哪些库文件需要下载，该项目的项目文件有哪些。
- 依次使用 `mvn clean install` 进行配置，同时还可以使用 `mvn compile` 对 `.class` 文件进行生成，`mvn package` 实现对项目文件的 `.class` 文件打包成 `jar` 文件。
- 将 `part-r-00000` 上传至 **HDFS** 的 `/input` 文件夹里面，最后运行该项目的 `jar` 文件，运行命令为：

```

./hadoop jar /home/njucs/shiyan2_2/target/shiyan2_2-1.0-SNAPSHOT.jar
weeklyBalanceAnalysis /input /output_2

```

注意要把导出出来的 `part-r-00000` 解锁，以实现普通用户可以打开，命令如下：

```

sudo chown $USER part-r-00000

```

2、程序运行结果

以下即为 `weeklyBalanceAnalysis.java` 程序执行的任务二（基于任务一的结果，编写统计一周七天中每天的平均资金流入与流出情况，并按照资金流入量从大到小排序。输出格式为" TAB < 资金流入量 >,< 资金流出量 >"）的运行结果：

- **程序运行结果图：**

```
lab1 [正在运行] - Oracle VM VirtualBox
五 15:28
root@njucs-VirtualBox: /usr/local/hadoop/bin

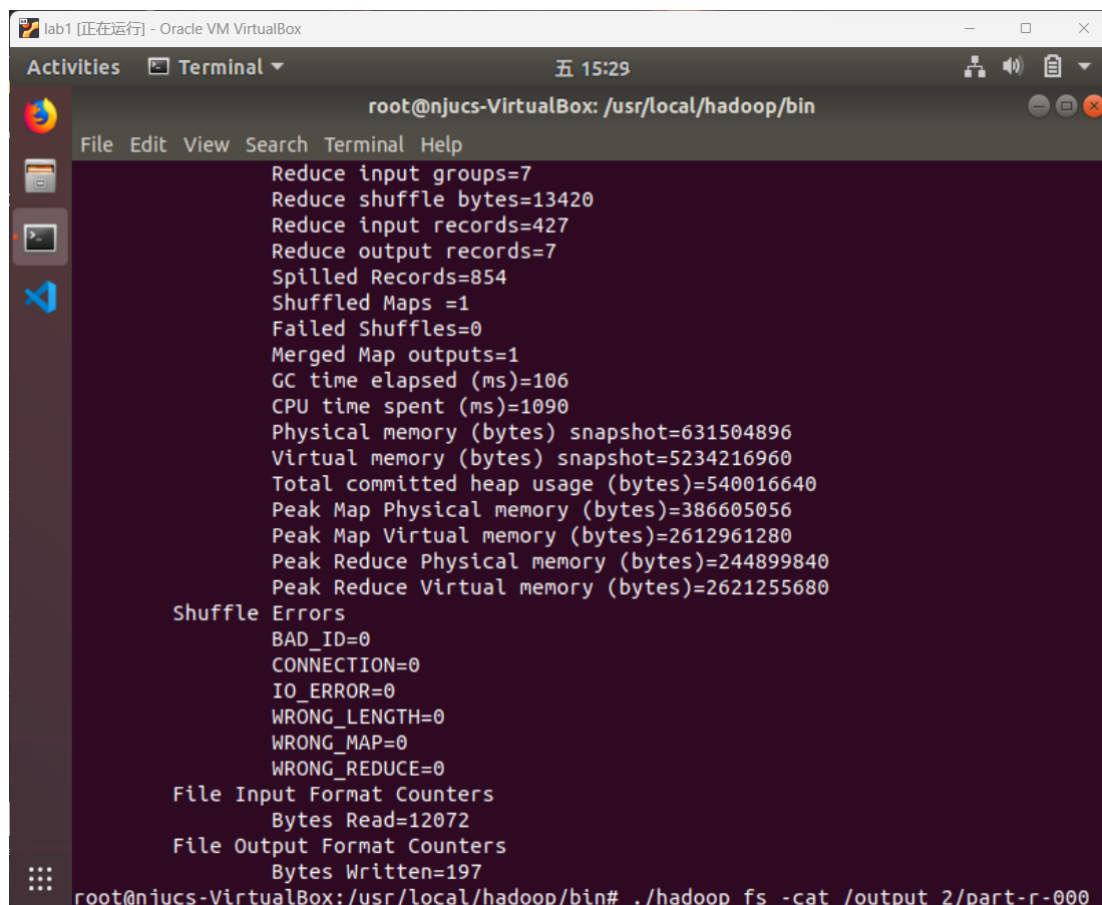
File Edit View Search Terminal Help

root@njucs-VirtualBox:/usr/local/hadoop/bin# ./hadoop jar /home/njucs/shiyan2_2/target/shiyan2_2-1.0-SNAPSHOT.jar WeeklyBalanceAnalysis /input /output_2
2024-11-01 15:23:55,077 INFO client.DefaultNoHARMAFailoverProxyProvider: Connecting to ResourceManager at localhost/127.0.0.1:8032
2024-11-01 15:23:55,501 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2024-11-01 15:23:55,550 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1730445791627_0001
2024-11-01 15:23:56,344 INFO input.FileInputFormat: Total input files to process : 1
2024-11-01 15:23:56,809 INFO mapreduce.JobSubmitter: number of splits:1
2024-11-01 15:23:56,943 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1730445791627_0001
2024-11-01 15:23:56,943 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-11-01 15:23:57,112 INFO conf.Configuration: resource-types.xml not found
2024-11-01 15:23:57,113 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-11-01 15:23:57,520 INFO impl.YarnClientImpl: Submitted application application_1730445791627_0001
2024-11-01 15:23:57,559 INFO mapreduce.Job: The url to track the job: http://njucs-VirtualBox:8088/proxy/application_1730445791627_0001/
2024-11-01 15:23:57,559 INFO mapreduce.Job: Running job: job_1730445791627_0001
2024-11-01 15:24:03,692 INFO mapreduce.Job: Job job_1730445791627_0001 running in uber mode : false
2024-11-01 15:24:03,693 INFO mapreduce.Job: map 0% reduce 0%
2024-11-01 15:24:07,760 INFO mapreduce.Job: map 100% reduce 0%
2024-11-01 15:24:12,803 INFO mapreduce.Job: map 100% reduce 100%
2024-11-01 15:24:13,817 INFO mapreduce.Job: Job job_1730445791627_0001 completed successfully
```

```
lab1 [正在运行] - Oracle VM VirtualBox
五 15:29
root@njucs-VirtualBox: /usr/local/hadoop/bin

File Edit View Search Terminal Help

2024-11-01 15:24:12,803 INFO mapreduce.Job: map 100% reduce 100%
2024-11-01 15:24:13,817 INFO mapreduce.Job: Job job_1730445791627_0001 completed successfully
2024-11-01 15:24:13,896 INFO mapreduce.Job: Counters: 54
  File System Counters
    FILE: Number of bytes read=13420
    FILE: Number of bytes written=644747
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=12177
    HDFS: Number of bytes written=197
    HDFS: Number of read operations=8
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=2
    HDFS: Number of bytes read erasure-coded=0
  Job Counters
    Launched map tasks=1
    Launched reduce tasks=1
    Data-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=2155
    Total time spent by all reduces in occupied slots (ms)=2084
    Total time spent by all map tasks (ms)=2155
    Total time spent by all reduce tasks (ms)=2084
    Total vcore-milliseconds taken by all map tasks=2155
    Total vcore-milliseconds taken by all reduce tasks=2084
    Total megabyte-milliseconds taken by all map tasks=2206720
    Total megabyte-milliseconds taken by all reduce tasks=2134016
  Map-Reduce Framework
```



The screenshot shows a terminal window titled 'lab1 [正在运行] - Oracle VM VirtualBox'. The terminal is running a Hadoop job and displays the following statistics:

```
root@njucs-VirtualBox: /usr/local/hadoop/bin
File Edit View Search Terminal Help
Reduce input groups=7
Reduce shuffle bytes=13420
Reduce input records=427
Reduce output records=7
Spilled Records=854
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=106
CPU time spent (ms)=1090
Physical memory (bytes) snapshot=631504896
Virtual memory (bytes) snapshot=5234216960
Total committed heap usage (bytes)=540016640
Peak Map Physical memory (bytes)=386605056
Peak Map Virtual memory (bytes)=2612961280
Peak Reduce Physical memory (bytes)=244899840
Peak Reduce Virtual memory (bytes)=2621255680
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=12072
File Output Format Counters
Bytes Written=197
root@njucs-VirtualBox: /usr/local/hadoop/bin# ./hadoop fs -cat /output 2/part-r-000
```

- part-r-00000输出结果图：



The screenshot shows a terminal window with the following output from the command `./hadoop fs -cat /output_2/part-r-00000`:

```
root@njucs-VirtualBox: /usr/local/hadoop/bin# ./hadoop fs -cat /output_2/part-r-00000
Tuesday 263582058,191769144
Monday 260305810,217463865
Wednesday 254162607,194639446
Thursday 236425594,176466674
Friday 199407923,166467960
Sunday 155914551,132427205
Saturday 148088068,112868942
root@njucs-VirtualBox: /usr/local/hadoop/bin#
```

3、WEB页面截图

lab1 [正在运行] - Oracle VM VirtualBox

Activities Firefox Web Browser 五 15:30

All Applications - Mozilla Firefox

All Applications

localhost:8088/cluster/apps 50%

doop All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resources
1	0	0	1	0	<memory:0 B, vCores:0>

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Last Nodes
1	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application
Capacity Scheduler	[memory-mb (unit-Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Run Cont
application_1730445791627_0001	root	Weekly Balance Analysis	MAPREDUCE		root.default	0	Fri Nov 1 15:23:57 +0800 2024	Fri Nov 1 15:23:57 +0800 2024	Fri Nov 1 15:24:12 +0800 2024	FINISHED	SUCCEEDED	N/A

Showing 1 to 1 of 1 entries