金融大数据实验 2 报告

221275029 杨珺雯

一、设计思路

1. 任务一

题目：根据 user_balance_table 表中的数据，编写 MapReduce 程序，统计所有用户每日的资金流入与流出情况。资金流入意味着申购行为，资金流出为赎回行为。

思路：设计四个类，分别是 FlowJob、FlowMapper、FlowReducer 和 FlowWritable。

FlowJob 里设置主方法，接收两个命令行参数，args[0] 是输入路径，args[1] 是输出路径。第一步，程序检查命令行参数是否正确，如果不符合预期的两个参数，程序会输出错误提示并退出；第二步，创建 Configuration 对象和 Job 实例；第三步，设置 Mapper 类和 Reducer 类；第四步，设置输出类型以及输入输出路径，最后提交作业。

FlowMapper 负责获取输入，并把输入处理为正确的格式传给 Reducer。在这个题目中，mapper 首先读取 csv 文件中的每一行（跳过第一行即表头），并转换为 String 类型；第二步，把每一行切分为单词，把申购金额和赎回金额从 String 类型转换为 double 类型，两个值存入新创立的 flow 对象里；第三步，日期作为键，flow 对象作为值传输给 Reducer。

FlowReducer 负责接收 mapper 传过来的键值对，对于相同的键，把对应的值进行累加，获得每个日期对应的累加申购金额和赎回金额。

FlowWritable 中设计了构造函数，Getter 方法、ToString 等方法，用于修改值的类型，便于输出。

2. 任务二

题目：基于任务一的结果，编写 MapReduce 程序，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。

思路：设计三个类，FundFlowDriver、FundFlowMapper 和 FundFlowReducer。

FundFlowDriver 设置主方法，大致思路和任务一相同。

FundFlowMapper 负责获取任务一的结果 output，处理后传输给 reducer。在这个题目中，mapper 首先提取 output_1 文件的每一行，转换为 String 类型后切分为单词（日期、申购金额、赎回金额）；第二步在已知 20130701 为星期一的

情况下，设计一个函数用于判断每个日期对应星期几；第三步把修改后的键值对传入 reducer。

FundFlowReducer 负责累计每天的资金流入和流出，进行排序后写入 context。第一步，创建数组，用于存储星期一到星期天的资金流入、资金流出和天数；第二步，把 mapper 传过来的键值对中的值进行存储，同时计算天数（count），分别存入数组中；第三步，利用存储的天数分别计算星期一到星期天的平均资金流入和流出，同时把星期天-星期六对应为 0-6，一起存入列表 dayFlows；最后利用列表的 sort 函数进行排序后写入 context。

3. 任务三

题目：根据 user_balance_table 表中的数据，编写 MapReduce 程序，统计每个用户的活跃天数，并按照活跃天数降序排列。

思路：设计三个类，ActiveDaysDriver、ActiveDaysMapper、ActiveDaysReducer、UserActiveDaysKey。

ActiveDaysDriver 和 FundFlowMapper 主要思路相同。

ActiveDaysMapper 首先读取 csv 文件，每一行转换成 String 类型后切分为字段；第二步根据直接购买金额和赎回金额来判断用户是否活跃，如果活跃就以用户 ID 为键，one 为值，作为键值对传输给 reducer。

ActiveDaysReducer 负责把相同用户 ID 的活跃天数逐一累加，并把用户 ID 和活跃天数作为键值对存入列表 userList 里；接着根据 sort 函数降序排序后写入 context。

UserActiveDaysKey 设计了构造函数以及一些 get 函数，主要功能和 FlowWritable 类似。

4. 任务四

题目：从其他的表中自行选取研究对象，通过 MapReduce（或其他工具），根据统计结果（也即类似于上面三个任务的结果）阐述某一因素对用户交易行为的影响。

思路：我选取了 mfd_day_share_interest.csv 文件中的 mfd_daily_yield（万份收益）、任务一的结果（申购金额和赎回金额）以及日期三个字段来统计不同的万份收益对用户申购金额和赎回金额的影响。根据题目设计三个类，InterestFlowDriver、InterestFlowMapper、InterestFlowReducer。

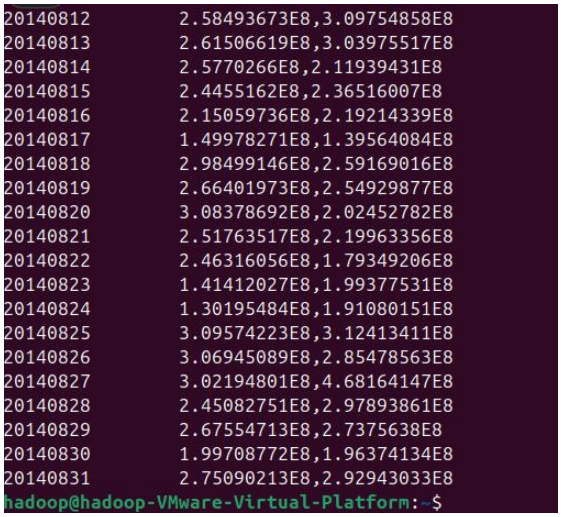
InterestFlowDriver 功能和上面的主函数类似。

InterestFlowMapper 首先读取 output_1（任务一的结果），获取日期、申购金额和赎回金额，存入 Map（dateToFlowMap）里；第二步，读取 csv 文件，获取日期和万份收益，利用 dateToFlowMap.get(date)函数获得该日期对应的申购金额和赎回金额；第三步，构建一个 level 函数，利用万份收益的最大值和最小值将该区间按顺序划分为十个区间，等级为 0-9；最后把万份收益对应的等级作为键，申购金额和赎回金额作为值，传输给 reducer。

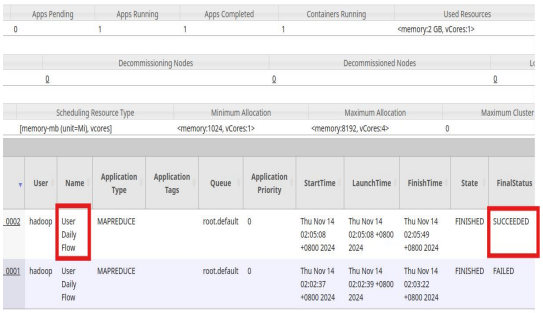
InterestFlowReducer 的操作和任务二类似，先读取数据、累加每个等级的次数、总申购金额和赎回金额并存入数组；计算平均资金流入和流出，最后写入 context。

二、程序运行结果

任务 1：输出了所有用户每日的资金流入与流出情况，输出情况按照日期顺序排列。具体输出请见 output_1。



任务一部分输出



网页查看任务一，执行成功

任务 2：输出了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。具体输出请见 output_2。

```
hadoop@hadoop-VMware-Virtual-Platform:~$ hdfs dfs -cat output/*
Sat.    148088068,112868942
Sun.    155914551,132427205
Fri.    199407923,166467960
Thur.   236425594,176466674
Wed.    254162607,194639446
Mon.    260305810,217463865
Tues.   263582058,191769144
hadoop@hadoop-VMware-Virtual-Platform:~$
```

任务二输出

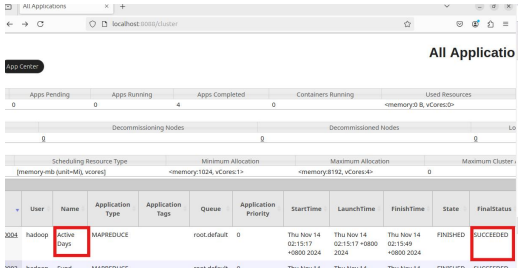
	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus
0003	hadoop	Fund Flow Analysis	MAPREDUCE		root.default	0	Thu Nov 14 02:10:29 +0800 2024	Thu Nov 14 02:10:30 +0800 2024	Thu Nov 14 02:10:55 +0800 2024	FINISHED	SUCCEEDED

网页查看任务二，执行成功

任务 3：输出了每个用户的活跃天数，并按照活跃天数降序排列。具体输出请见 output_3。

```
24043   18
24521   18
24825   18
24976   18
25217   18
25224   18
25338   18
25350   18
25380   18
25437   18
25913   18
2601    18
26203   18
26468   18
2647    18
26710   18
26763   18
26822   18
26919   18
26961   18
27490   18
27764   18
2777    18
```

任务三部分输出

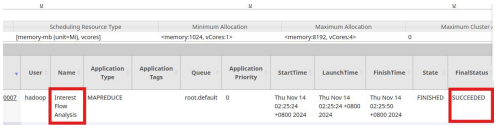


网页查看任务三，执行成功

任务 4：输出了不同的万份收益等级中用户不同的申购金额和赎回金额。大致情况申购金额和赎回金额随万份收益等级增加而增加，少部分等级不符合这个规律。具体输出请见 output_4。

```
hadoop@hadoop-VMware-Virtual-Platform:~$ hdfs dfs -cat output/*
Level 0 (1.10 - 1.17)   243335456.62    264034226.30
Level 1 (1.17 - 1.24)   80956489.34     75661329.34
Level 2 (1.24 - 1.32)   157617852.47    128230458.75
Level 3 (1.32 - 1.39)   179841994.31    139083070.84
Level 4 (1.39 - 1.46)   224274301.80    186042709.73
Level 5 (1.46 - 1.53)   259067294.55    231014439.38
Level 6 (1.53 - 1.61)   411843574.27    257909305.73
Level 7 (1.61 - 1.68)   371093839.59    185647109.81
Level 8 (1.68 - 1.75)   458039434.26    243966976.63
Level 9 (1.75 - 1.82)   303157175.89    145942169.56
hadoop@hadoop-VMware-Virtual-Platform:~$
```

任务四输出

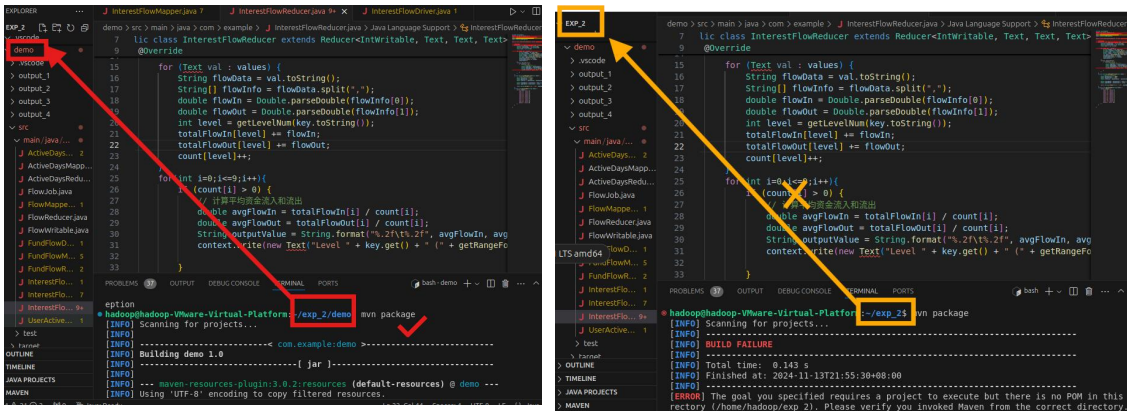


网页查看任务四，执行成功

三、遇到错误和解决方法

1.问题：在 vscode 中无法对 maven 项目进行打包，会出现 pom.xml 无法识别的报错。

原因：发生错误时打包命令的目录是作业文件夹的目录，而不是文件夹下项目的目录。以下面的截图为例，发生错误时打包命令所在的目录是 hm_5，只需要 cd demo 进入项目，再进行打包就可以（即使此时有关 pom.xml 里导入包的代码还是会标红，但是不影响打包和 hadoop 运行了）。



2.问题：任务 2 在运行项目之后不是干净利落的每一行对应星期几，一共七行，而是有很多个星期一、星期二等等。

原因：没有真正理解 mapreduce 的工作原理。出错的时候是在 reducer 类里修改了键的，在 reducer 类里修改键之后会出现错误，不再像正常那样统计每个键对应的值了。如果要修改，最好在 mapper 里修改完成后再传输给 reducer。

3.问题：任务 4 在运行项目时发现读取的文件和写入的路径不符。

原因：在 main 函数里设置了接收的三个命令行参数，args[0] 是输入路径，args[1] 是输出路径，args[2]是多出来的输入路径。但是在写代码的时候把 args[] 里面的数写错了，造成后面运行也有问题。