注：

代码在shiyan2/demo1/src/main/java/com/example　任务和代码的对应关系看commit message

输出结果在shiyan2/demo1/output

（第一次运行时没有终端截图，全部写完后再重跑的，故运行截图和github上的日期不大一致。）

1 每日资金流入流出统计

要求：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。

注：每笔交易的资金流入和流出量分别由字段 total_purchase_amt 和 total_redeem_amt 表示。请注意处理数据中的缺失值，将其视为零交易。

输出格式：< 日期 > TAB < 资金流入量 >,< 资金流出量 >

1.1 设计思路

直观想法就是按照日期，对所有用户的total_purchase_amt 和 total_redeem_amt 分别进行加总。

- Mapper
    1. 逐行处理数据，使用split方法切片，提取字段日期、资金流入和资金流出。
    2. 排除表头行并检查字段数量是否足够。
    3. 处理缺失值，将其视为零交易。
    4. 将字符串转换为数字，确保没有无效数据，忽略含有无效数字的行。
    5. 将日期作为key，将资金流入和流出拼接成字符串作为value，并写入context。
- Reducer
    1. 初始化资金流入和流出为0。
    2. 遍历所有的value，逐一累加资金流入和流出。
    3. 输出该日期（key）和对应的资金流入、流出金额。
- main

  设置MapReduce作业的配置：指定作业名称；配置Mapper和Reducer类；设置输出键值类型；指定输入、输出文件路径（args[0]和 args[1]）；启动作业并等待作业完成。

1.2 程序运行结果

- 终端输出结果

  运行MapReduce结果

```
root@h01:/usr/local/hadoop# ./bin/hadoop jar demo1-1.0-SNAPSHOT.jar com.example.
AmtStatistics /input/user_balance_table.csv /output/amt_statistics
2024-11-06 09:24:54,873 INFO client.DefaultNoHARMFailoverProxyProvider: Connecti
ng to ResourceManager at h01/172.18.0.2:8032
2024-11-06 09:24:55,653 WARN mapreduce.JobResourceUploader: Hadoop command-line
option parsing not performed. Implement the Tool interface and execute your appl
ication with ToolRunner to remedy this.
2024-11-06 09:24:55,701 INFO mapreduce.JobResourceUploader: Disabling Erasure Co
ding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1730884828760_0001
2024-11-06 09:24:57,204 INFO input.FileInputFormat: Total input files to process
 : 1
2024-11-06 09:24:58,167 INFO mapreduce.JobSubmitter: number of splits:2
2024-11-06 09:24:58,434 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1730884828760_0001
2024-11-06 09:24:58,435 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-11-06 09:24:58,898 INFO conf.Configuration: resource-types.xml not found
2024-11-06 09:24:58,900 INFO resource.ResourceUtils: Unable to find 'resource-ty
pes.xml'.
2024-11-06 09:25:00,369 INFO impl.YarnClientImpl: Submitted application applicat
ion_1730884828760_0001
2024-11-06 09:25:00,462 INFO mapreduce.Job: The url to track the job: http://h01
:8088/proxy/application_1730884828760_0001/
```

root@h01: /usr/local/hadoop

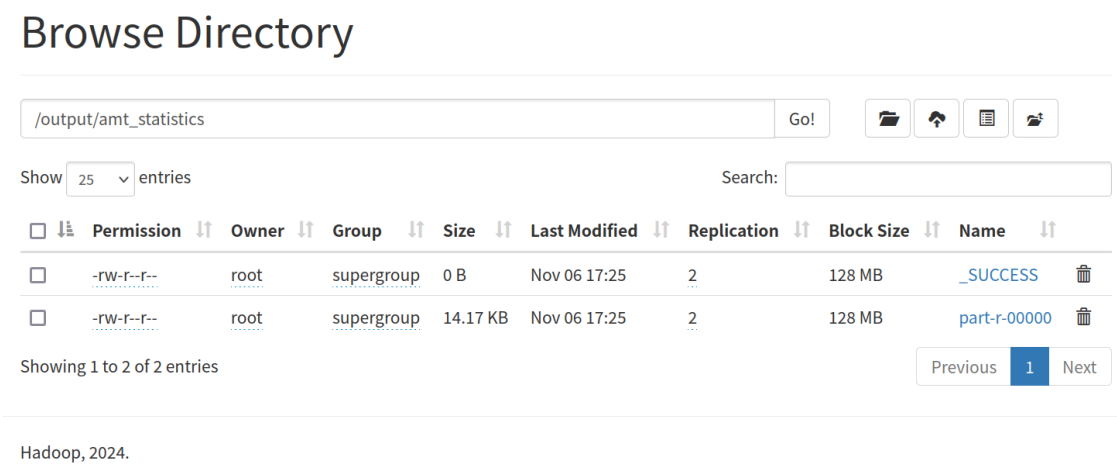
```
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=5724
                CPU time spent (ms)=28320
                Physical memory (bytes) snapshot=1472032768
                Virtual memory (bytes) snapshot=7789596672
                Total committed heap usage (bytes)=1548746752
                Peak Map Physical memory (bytes)=613449728
                Peak Map Virtual memory (bytes)=2596200448
                Peak Reduce Physical memory (bytes)=254500864
                Peak Reduce Virtual memory (bytes)=2602209280
        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=157765297
        File Output Format Counters
                Bytes Written=14515
root@h01:/usr/local/hadoop#
```

打印输出文件结果

root@h01: /usr/local/hadoop

```
        File Output Format Counters
                Bytes Written=14515
root@h01:/usr/local/hadoop# ./bin/hadoop fs -cat /output/amt_statistics/part-r-0
0000
20130701        3.2488348E7,5525022.0
20130702        2.903739E7,2554548.0
20130703        2.727077E7,5953867.0
20130704        1.8321185E7,6410729.0
20130705        1.1648749E7,2763587.0
20130706        3.6751272E7,1616635.0
20130707        8962232.0,3982735.0
20130708        5.7258266E7,8347729.0
20130709        2.6798941E7,3473059.0
20130710        3.0696506E7,2597169.0
20130711        4.4075197E7,3508800.0
20130712        3.4183904E7,8492573.0
20130713        1.5164717E7,3482829.0
20130714        2.2615303E7,2784107.0
20130715        4.8128555E7,1.3107943E7
20130716        5.0622847E7,1.1864981E7
20130717        2.9015682E7,1.0911513E7
20130718        2.4234505E7,1.1765356E7
20130719        3.3680124E7,9244769.0
20130720        2.0439079E7,4601143.0
```

- web页面截图



1.3 程序分析

进一步对性能、扩展性等方面存在的不足和可能改进之处进行分析。

1.3.1 Reducer的资源消耗

不足：Reducer直接累加每个key对应的所有value，可能导致内存和CPU开销较高。

改进分析：在Reducer中引入缓存或批量处理机制，对于超大数据集可以更灵活地管理资源分配。此外，若数据倾斜（某些日期的资金流入流出记录较多），可以考虑进行数据分区处理（Partitioner），确保数据均匀分布到不同的Reducer。

1.3.2 数据格式的依赖

不足：当前代码假定输入数据的字段顺序和数量固定，一旦数据格式变化，代码可能无法适应。

改进分析：引入配置文件或参数化机制，将字段索引设置为外部参数，提升代码对数据格式变化的适应性。

2 星期交易量统计

要求：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。

输出格式： TAB < 平均资金流入量 >,< 平均资金流出量 >

2.1 设计思路

直观想法是根据任务一的输出，把日期转换为weekday，生成键值对，然后在Reducer进行求均值。

- Mapper
    1. 逐行处理数据，使用split方法切片，提取日期、资金流入和资金流出。
    2. 将日期转换成对应的星期几。
    3. 将日期作为key，将资金流入和流出作为value，并写入context。
- Reducer
    1. 维护两个HashMap，存储每个weekday的交易金额总以及交易次数。对于每个weekday（key），累加所有资金流入和流出金额，并记录总次数。

2. 在cleanup方法中，对所有weekday的资金流入和流出金额进行平均值计算，并根据平均资金流入金额进行降序排序。
3. 调整输出格式为 TAB < 平均资金流入量 >,< 平均资金流出量 >

- main

  设置MapReduce作业的配置：指定作业名称；配置Mapper和Reducer类；设置输出键值类型；指定输入、输出文件路径（args[0]和 args[1]）；启动作业并等待作业完成。

2.2 程序运行结果

- 终端输出结果

  运行MapReduce结果

```
root@h01:/usr/local/hadoop# ./bin/hadoop jar demo1-1.0-SNAPSHOT.jar com.example.
Task2 /output/amt_statistics/part-r-00000 /output/weekly_avg
2024-11-06 09:38:13,243 INFO client.DefaultNoHARMFailoverProxyProvider: Connecti
ng to ResourceManager at h01/172.18.0.2:8032
2024-11-06 09:38:13,947 WARN mapreduce.JobResourceUploader: Hadoop command-line
option parsing not performed. Implement the Tool interface and execute your appl
ication with ToolRunner to remedy this.
2024-11-06 09:38:13,974 INFO mapreduce.JobResourceUploader: Disabling Erasure Co
ding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1730884828760_0002
2024-11-06 09:38:14,445 INFO input.FileInputFormat: Total input files to process
 : 1
2024-11-06 09:38:14,629 INFO mapreduce.JobSubmitter: number of splits:1
2024-11-06 09:38:14,825 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1730884828760_0002
2024-11-06 09:38:14,825 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-11-06 09:38:15,063 INFO conf.Configuration: resource-types.xml not found
2024-11-06 09:38:15,063 INFO resource.ResourceUtils: Unable to find 'resource-ty
pes.xml'.
2024-11-06 09:38:15,235 INFO impl.YarnClientImpl: Submitted application applicat
ion_1730884828760_0002
2024-11-06 09:38:15,346 INFO mapreduce.Job: The url to track the job: http://h01
:8088/proxy/application_1730884828760_0002/
2024-11-06 09:38:15,348 INFO mapreduce.Job: Running job: job_1730884828760_0002
2024-11-06 09:38:23,547 INFO mapreduce.Job: Job job_1730884828760_0002 running i
n uber mode : false
2024-11-06 09:38:23,548 INFO mapreduce.Job:  map 0% reduce 0%
```

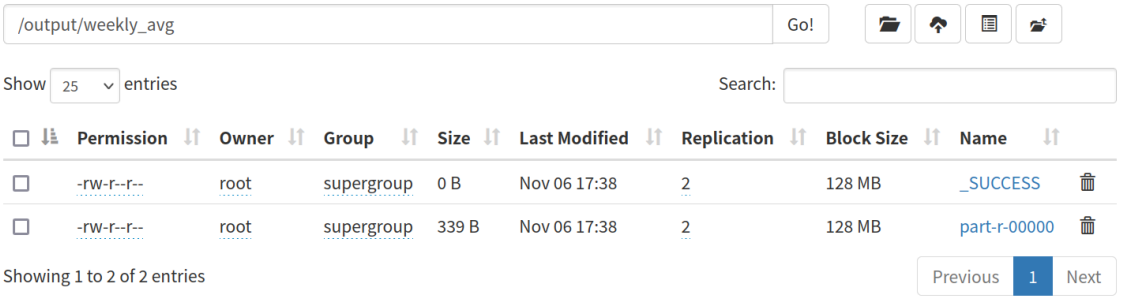
```
root@h01: /usr/local/hadoop
                Reduce shuffle bytes=15009
                Reduce input records=427
                Reduce output records=7
                Spilled Records=854
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=125
                CPU time spent (ms)=1520
                Physical memory (bytes) snapshot=521101312
                Virtual memory (bytes) snapshot=5196173312
                Total committed heap usage (bytes)=632291328
                Peak Map Physical memory (bytes)=266424320
                Peak Map Virtual memory (bytes)=2593673216
                Peak Reduce Physical memory (bytes)=254676992
                Peak Reduce Virtual memory (bytes)=2602500096
        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=14515
        File Output Format Counters
                Bytes Written=339
root@h01:/usr/local/hadoop#
```

  打印输出文件结果

```
root@h01:/usr/local/hadoop# ./bin/hadoop fs -cat /output/weekly_avg/part-r-00000

Tuesday 2.6358205886885247E8,1.9176914462295082E8
Monday  2.6030581E8,2.174638654918033E8
Wednesday       2.5416260783606556E8,1.946394465081967E8
Thursday        2.3642559403278688E8,1.764666748852459E8
Friday  1.9940792306557378E8,1.6646796019672132E8
Sunday  1.5591455193442622E8,1.3242720506557377E8
Saturday        1.4808806829508197E8,1.1286894208196722E8
```

- web页面截图



2.3 程序分析

进一步对性能、扩展性等方面存在的不足和可能改进之处进行分析。

2.3.1 Mapper和Reducer之间的数据传输量

不足：Mapper和Reducer之间的数据传输量较大，当数据集过大时，会影响作业效率。

改进分析：可以添加一个Combiner，以在Map端先行聚合相同key的value，减少Mapper和Reducer之间的数据传输量，提高MapReduce作业的效率。

2.3.2 排序阶段的开销

不足：当前代码在`cleanup()`方法中排序所有数据，时间复杂度较高。这对于较小数据集是可以的，但如果每个键对应的数据量较大，排序过程可能耗时太长。

改进分析：

- 采用二级MapReduce任务：在第一个Reducer中输出后，将结果作为输入，第二个MapReduce任务完成排序。
- 另一个替代方案是使用外部排序，将输出结果保存到文件后再用工具进行排序。

3 用户活跃度分析

要求：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。当用户当日有直接购买（direct_purchase_amt 字段大于 0 ）或赎回行为（total_redeem_amt字段大于 0 ）时，则该用户当天活跃。

输出格式： < 用户 ID > TAB < 活跃天数 >

3.1 设计思路

最直观的想法是先初步对每一行判断用户当天是否活跃，若是输出用户id和活跃天数（初始化为1）的键值对，再对重复出现的用户id进行累加。最后做排序和格式的修改。

- Mapper
    1. 逐行处理数据，使用split方法切片，提取用户id及直接购买 direct_purchase_amt 和赎回行为 total_redeem_amt 字段。

2. 判断用户的 direct_purchase_amt 或 total_redeem_amt是否大于0，即用户当日是否有直接购买或赎回行为，确定某用户当天是否活跃。
3. 若用户当天活跃，则输出键值对 < 用户id, 1>，表示该用户活跃的天数为1。

- Reducer
  1. 维护一个 ArrayList，用于存储每个用户的id及其活跃天数。
  2. 对每个用户id的活跃天数进行累加。
  3. 在 cleanup 方法中，对所有用户的活跃天数进行降序排序。
  4. 调整输出格式为 < 用户 ID> TAB < 活跃天数 >

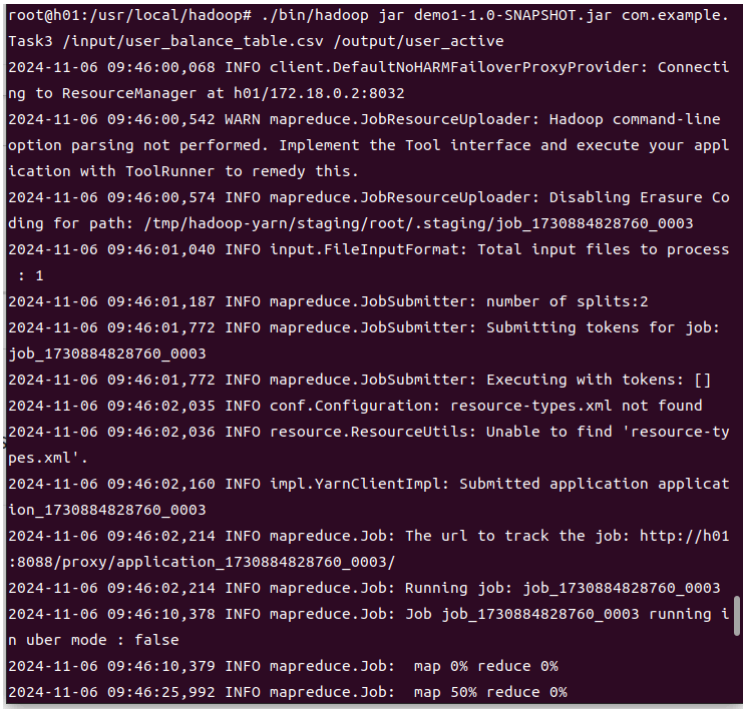
- main

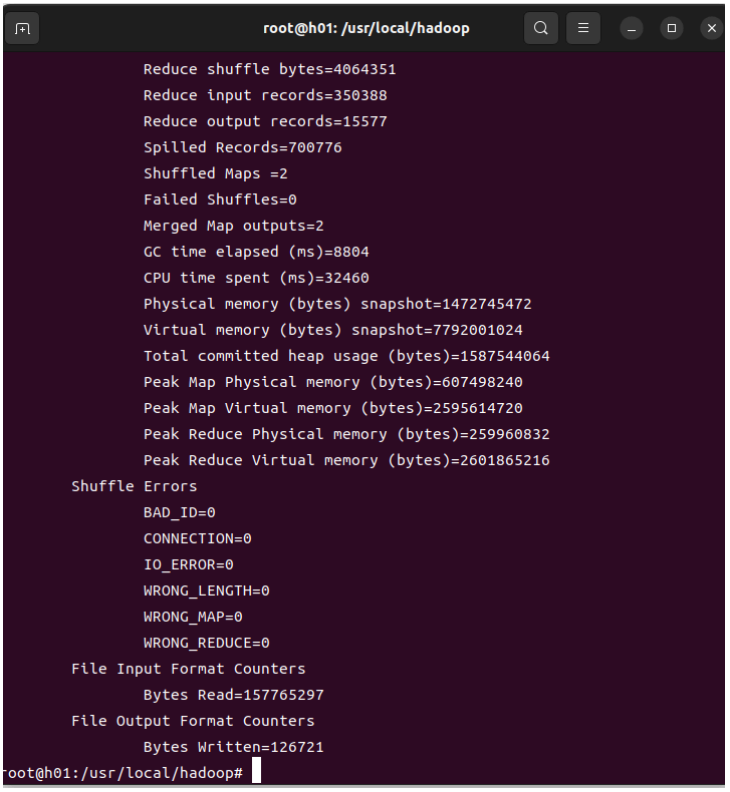
  设置MapReduce作业的配置：指定作业名称；配置Mapper和Reducer类；设置输出键值类型；指定输入、输出文件路径（args[0]和 args[1]）；启动作业并等待作业完成。

3.2 程序运行结果

- 终端输出结果

  运行MapReduce结果

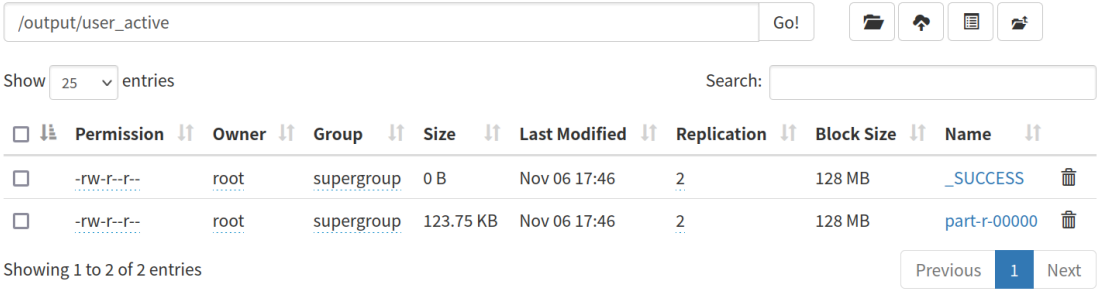




打印输出文件结果

```
root@h01:/usr/local/hadoop# ./bin/hadoop fs -cat /output/user_active/part-r-0000
0 | head -n 100
7629    384
11818   359
21723   334
19140   332
24378   315
26395   297
25147   295
27719   293
20515   291
5016    287
27751   285
14472   280
25951   280
2521    277
13435   268
5284    262
26554   260
4561    260
24259   257
7848    251
18521   249
24474   249
7320    240
12105   236
```

- web页面截图



3.3 程序分析

进一步对性能、扩展性等方面存在的不足和可能改进之处进行分析。

3.3.1 排序阶段的开销

不足：在Reducer的cleanup方法中，将所有用户的活跃天数存入一个List再排序。这种操作会在内存中存储大量数据，消耗大量内存，且无法处理大规模数据。

改进分析：

- 采用二级MapReduce任务：在第一个Reducer中输出<用户ID, 活跃天数>后，将结果作为输入，第二个MapReduce任务完成排序。
- 另一个替代方案是使用外部排序，将输出结果保存到文件后再用工具进行排序。

3.3.2 代码的可扩展性与数据类型优化

不足：在数据类型选择上，`IntWritable`虽然适合活跃天数的存储，但如果天数超过了`Integer`的范围，可能会导致溢出。

改进分析：若数据集更大，可以考虑使用`LongWritable`这种更大的数据类型，以应对数据增长。

4 交易行为影响因素分析

要求：用户的交易行为（如：余额宝或银行卡的购买或赎回，用户的消费情况等）受到很多因素的影响。例如：用户特性（参考用户信息表 user_profile_table ），当前利率（参考支付宝收益率 mfd_day_share_interest 以及银利行率表 mfd_bank_shibor ）。 在上面的三个任务中，我们重点研究了 user_balance_table 表中的数据。现在，请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。（即使你的结论是某一因素对用户的交易行为没有显著影响，这样的结果也是完全 OK 的。本次实验重点关注的是使用 MapReduce 进行统计的过程。）

研究内容：

尝试研究用户星座对用户活跃天数的影响，通过将用户的星座信息与其活跃天数关联，计算每个星座的平均活跃天数，旨在研究星座特性对用户活跃行为的潜在影响。

这种分析在实际应用中具有一定意义：例如，用户的活跃度可能与其个性、兴趣或生活习惯相关，星座是个性特征的参考之一。企业可以借助此类数据，为不同特征的用户定制服务或产品推荐策略，以提升用户体验和满意度。

4.1 设计思路

直观想法是对于先将每个用户id和活跃天数的对应关系转化成星座和活跃天数的对应关系，再对相同星座的活跃天数进行累加，最后求均值。得到12个星座用户的平均活跃天数

输出格式为 < 星座名 > TAB < 平均活跃天数 >

- Mapper

  负责处理两个输入文件：用户活跃天数文件和用户信息文件。

  1. 在setup方法中，读取用户信息文件，跳过表头并提取用户ID和星座信息，建立用户ID与星座的映射（userConstellationMap）。
  2. 在map方法中，读取用户活跃天数文件，逐行处理数据，读取用户ID和活跃天数。
  3. 检查该用户ID是否存在于用户ID与星座的映射（userConstellationMap）中，如果用存在，将星座名称作为key，活跃天数作为value，写入context。

- Reducer

1. 接收来自Mapper的输出<星座名, 活跃天数>键值对，并进行汇总。
2. 对每个星座的活跃天数进行累加，并计数每个星座下的用户数量。
3. 计算每个星座的平均活跃天数
4. 调整输出格式为 < 星座名 > TAB < 平均活跃天数 >

- main

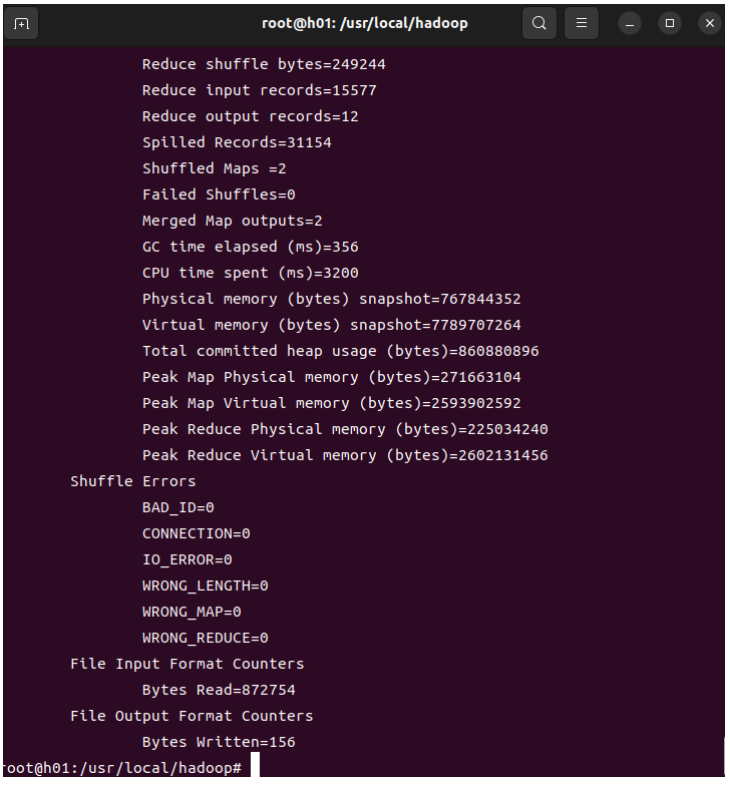
  设置MapReduce作业的配置：指定作业名称；配置Mapper和Reducer类；设置输出键值类型；指定输入、输出文件路径（args[0]和 args[1]）；启动作业并等待作业完成。

4.2 程序运行结果

- 终端输出结果

  运行MapReduce结果

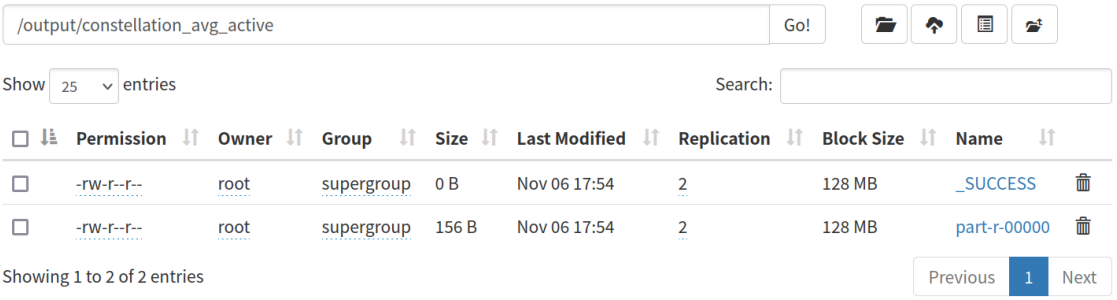
```
root@h01:/usr/local/hadoop# ./bin/hadoop jar demo1-1.0-SNAPSHOT.jar com.example.
Task4 /output/user_active/part-r-00000 /input/user_profile_table.csv /output/con
stellation_avg_active
2024-11-06 09:53:56,248 INFO client.DefaultNoHARMFailoverProxyProvider: Connecti
ng to ResourceManager at h01/172.18.0.2:8032
2024-11-06 09:53:56,876 WARN mapreduce.JobResourceUploader: Hadoop command-line
option parsing not performed. Implement the Tool interface and execute your appl
ication with ToolRunner to remedy this.
2024-11-06 09:53:56,925 INFO mapreduce.JobResourceUploader: Disabling Erasure Co
ding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1730884828760_0004
2024-11-06 09:53:57,480 INFO input.FileInputFormat: Total input files to process
 : 2
2024-11-06 09:53:57,686 INFO mapreduce.JobSubmitter: number of splits:2
2024-11-06 09:53:58,030 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1730884828760_0004
2024-11-06 09:53:58,031 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-11-06 09:53:58,586 INFO conf.Configuration: resource-types.xml not found
2024-11-06 09:53:58,587 INFO resource.ResourceUtils: Unable to find 'resource-ty
pes.xml'.
2024-11-06 09:53:59,625 INFO impl.YarnClientImpl: Submitted application applicat
ion_1730884828760_0004
2024-11-06 09:54:00,008 INFO mapreduce.Job: The url to track the job: http://h01
:8088/proxy/application_1730884828760_0004/
2024-11-06 09:54:00,010 INFO mapreduce.Job: Running job: job_1730884828760_0004
2024-11-06 09:54:11,441 INFO mapreduce.Job: Job job_1730884828760_0004 running i
n uber mode : false
2024-11-06 09:54:11,442 INFO mapreduce.Job:  map 0% reduce 0%
2024-11-06 09:54:18,591 INFO mapreduce.Job:  map 100% reduce 0%
```



打印输出文件结果

```
root@h01:/usr/local/hadoop# ./bin/hadoop fs -cat /output/constellation_avg_activ
e/part-r-00000
双子座  23
双鱼座  22
处女座  21
天秤座  23
天蝎座  22
射手座  21
巨蟹座  21
摩羯座  22
水瓶座  23
狮子座  21
白羊座  22
金牛座  23
root@h01:/usr/local/hadoop#
```

- web页面截图



4.3 程序分析

4.3.1 Mapper内存消耗

不足：`userConstellationMap`将用户ID和星座存储在内存中，当用户数非常大时，内存占用可能会显著增加，导致性能下降甚至内存不足。

改进分析：考虑通过外部存储服务（如HBase或DynamoDB）来存储和查询用户星座信息，避免占用Mapper的内存资源。

4.3.2 代码扩展性

不足：当前程序将星座作为唯一的分组依据，且星座的格式没有验证。若后续需求变更，需要根据更多字段（例如性别、年龄段）进行分析，代码可能需要较大改动。

改进分析：可以增加字段配置的灵活性，通过配置文件或作业参数指定分组依据，增强程序的通用性。同时在读取数据时可增加字段的格式校验，确保数据一致性。

4.4 研究结论

从输出结果来看，十二星座的用户活跃天数都在21-23的区间内，差异不大。故得出结论：星座这一用户特性对于用户活跃天数并没有显著的影响。