第五次作业

221275037刘紫艺

一、准备工作

1. 先启动节点，将analyst_ratings.csv和stop-word-list.txt从本地传输到h01的tmp中
2. 在h01的bin/hdfs中新建input和output文件夹
3. 将tmp中的两个文件移动到input

二、代码运行

1. 先将java文件从本地传输到docker的hadoop文件夹中
2. 输入 `/usr/local/hadoop/bin/badoop classpath` 找到classpath的路径

   输入 `javac -classpath "/usr/local/hadoop/etc/hadoop:/usr/local/hadoop/share/hadoop/common/lib/*:/usr/local/hadoop/share/hadoop/common/*:/usr/local/hadoop/share/hadoop/hdfs:/usr/local/hadoop/share/hadoop/hdfs/lib/*:/usr/local/hadoop/share/hadoop/hdfs/*:/usr/local/hadoop/share/hadoop/mapreduce/*:/usr/local/hadoop/share/hadoop/yarn:/usr/local/hadoop/share/hadoop/yarn/lib/*:/usr/local/hadoop/share/hadoop/yarn/*" StockCodeCount.java` 生成class文件

   报错：

```
StockCodeCount.java:16: error: unmappable character for encoding ASCII
    // Mapper ???
               ^
StockCodeCount.java:16: error: unmappable character for encoding ASCII
    // Mapper ???
                ^
StockCodeCount.java:16: error: unmappable character for encoding ASCII
    // Mapper ???
                 ^
StockCodeCount.java:31: error: unmappable character for encoding ASCII
    // Reducer ???
               ^
StockCodeCount.java:31: error: unmappable character for encoding ASCII
    // Reducer ???
                ^
StockCodeCount.java:31: error: unmappable character for encoding ASCII
    // Reducer ???
                 ^
StockCodeCount.java:34: error: unmappable character for encoding ASCII
        // ?????? TreeMap ??????????????????????????????????????????????????
           ^
StockCodeCount.java:34: error: unmappable character for encoding ASCII
        // ?????? TreeMap ??????????????????????????????????????????????????
            ^
StockCodeCount.java:34: error: unmappable character for encoding ASCII
        // ?????? TreeMap ??????????????????????????????????????????????????
             ^
StockCodeCount.java:34: error: unmappable character for encoding ASCII
        // ?????? TreeMap ??????????????????????????????????????????????????
              ^
```

   改：指定编码方式

```
root@h01:/usr/local/hadoop# javac -encoding UTF-8 -classpath "/usr/local/hadoop/
etc/hadoop:/usr/local/hadoop/share/hadoop/common/lib/*:/usr/local/hadoop/share/h
adoop/common/*:/usr/local/hadoop/share/hadoop/hdfs:/usr/local/hadoop/share/hadoo
p/hdfs/lib/*:/usr/local/hadoop/share/hadoop/hdfs/*:/usr/local/hadoop/share/hadoo
p/mapreduce/*:/usr/local/hadoop/share/hadoop/yarn:/usr/local/hadoop/share/hadoop
/yarn/lib/*:/usr/local/hadoop/share/hadoop/yarn/*" StockCodeCount.java
```

3. 生成jar文件

```
root@h01:/usr/local/hadoop# jar -cvf StockCodeCount.jar StockCodeCount*.class
added manifest
adding: StockCodeCount$StockMapper.class(in = 1824) (out= 769)(deflated 57%)
adding: StockCodeCount$StockReducer.class(in = 3859) (out= 1607)(deflated 58%)
adding: StockCodeCount.class(in = 1463) (out= 788)(deflated 46%)
```

4. 运行 将结果放入output中新建的hm51文件夹

```
root@h01:/usr/local/hadoop# ./bin/hadoop jar StockCodeCount.jar StockCodeCount /
input/analyst_ratings.csv /output/hm51
```

5. 查看结果并输出

```
root@h01:/usr/local/hadoop# ./bin/hadoop fs -cat /output/hm51/part-r-00000
1：MS,  726
2：MRK,         704
3：QQQ,         693
4：BABA,        689
5：EWU,         681
6：GILD,        663
7：JNJ,         663
8：MU,  659
9：NVDA,        655
10：VZ,         648
11：KO,         643
12：QCOM,       636
13：M,  635
14：NFLX,       635
15：EBAY,       621
16：DAL,        605
17：WFC,        582
18：BBRY,       581
19：ORCL,       575
20：FDX,        573
21：BMY,        563
22：AA,         561
23：JCP,        559
24：EWP,        553
25：NOK,        532
26：EWJ,        526
27：GLD,        513
28：EWI,        510
29：LMT,        509
30：CHK,        508
31：GPRO,       508
32：HD,         506
```

6. 将结果传回本机

任务一

设计思路

1. Map 阶段：

   读取 CSV 文件的每一行，提取第四列的股票代码。

   为每个股票代码输出一次计数1。

2. Reduce 阶段：

   接收 Map 阶段输出的股票代码及其计数。

   汇总每个股票代码的总出现次数。

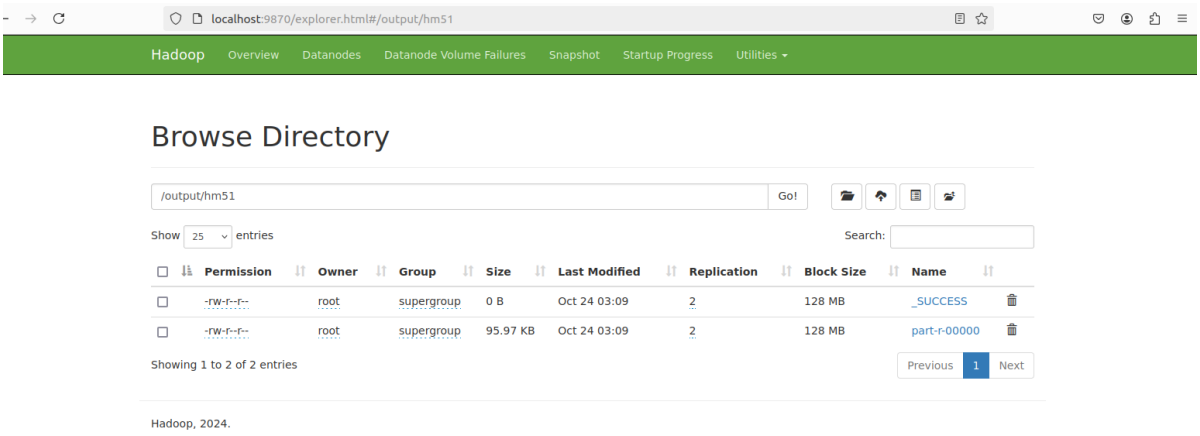
使用 `TreeMap` 按照出现次数降序排序股票代码。

最终输出带有排名的股票代码和其出现次数。

运行结果

```
root@h01:/usr/local/hadoop# ./bin/hadoop jar StockCodeCount.jar StockCodeCount /
input/analyst_ratings.csv /output/hm51
2024-10-23 19:08:33,459 INFO client.DefaultNoHARMFailoverProxyProvider: Connecti
ng to ResourceManager at h01/172.18.0.2:8032
2024-10-23 19:08:34,848 WARN mapreduce.JobResourceUploader: Hadoop command-line
option parsing not performed. Implement the Tool interface and execute your appl
ication with ToolRunner to remedy this.
2024-10-23 19:08:34,902 INFO mapreduce.JobResourceUploader: Disabling Erasure Co
ding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1729703252844_0010
2024-10-23 19:08:35,627 INFO input.FileInputFormat: Total input files to process
 : 1
2024-10-23 19:08:35,903 INFO mapreduce.JobSubmitter: number of splits:1
2024-10-23 19:08:36,273 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1729703252844_0010
2024-10-23 19:08:36,274 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-10-23 19:08:36,986 INFO conf.Configuration: resource-types.xml not found
2024-10-23 19:08:36,988 INFO resource.ResourceUtils: Unable to find 'resource-ty
pes.xml'.
2024-10-23 19:08:37,262 INFO impl.YarnClientImpl: Submitted application applicat
ion_1729703252844_0010
2024-10-23 19:08:37,610 INFO mapreduce.Job: The url to track the job: http://h01
:8088/proxy/application_1729703252844_0010/
2024-10-23 19:08:37,613 INFO mapreduce.Job: Running job: job_1729703252844_0010
2024-10-23 19:08:54,240 INFO mapreduce.Job: Job job_1729703252844_0010 running i
n uber mode : false
2024-10-23 19:08:54,245 INFO mapreduce.Job:  map 0% reduce 0%
2024-10-23 19:09:08,746 INFO mapreduce.Job:  map 100% reduce 0%
2024-10-23 19:09:18,935 INFO mapreduce.Job:  map 100% reduce 100%
2024-10-23 19:09:19,963 INFO mapreduce.Job: Job job_1729703252844_0010 completed
 successfully
2024-10-23 19:09:20,300 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=3042970
                FILE: Number of bytes written=6703449
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
```

```
root@h01:/usr/local/hadoop# ./bin/hadoop fs -cat /output/hm51/part-r-00000
1：MS,  726
2：MRK,         704
3：QQQ,         693
4：BABA,        689
5：EWU,         681
6：GILD,        663
7：JNJ,         663
8：MU,  659
9：NVDA,        655
10：VZ,         648
11：KO,         643
12：QCOM,       636
13：M,  635
14：NFLX,       635
15：EBAY,       621
16：DAL,        605
17：WFC,        582
18：BBRY,       581
19：ORCL,       575
20：FDX,        573
21：BMY,        563
22：AA,         561
23：JCP,        559
24：EWP,        553
25：NOK,        532
26：EWJ,        526
27：GLD,        513
28：EWI,        510
29：LMT,        509
30：CHK,        508
31：GPRO,       508
32：HD,         506
33：TWX,        506
34：GPS,        502
35：P,  501
36：MCD,        494
37：AGN,        485
38：GRPN,       477
39：LLY,        474
40：AZN,        471
```



不足和可改进之处

此代码是将所有数据最终汇总到一个 `Reducer` 中处理，对于大规模数据集，单个 `Reducer` 的内存和处理能力有限，难以扩展

可以通过增加 Reducer 的数量来减轻单个 Reducer 的负担，例如通过 `job.setNumReduceTasks(n)` 来设置多个 Reducer

任务二

设计思路

1.Mapper类

`WordMapper`类负责将文本转换成键值对，键是单词，值是计数1，并且用停用词列表来过滤无意义的常见单词

`map()`方法每处理一行数据，并去除标点，然后将符合条件的单词发送到`context.write()`中，发送到下一步。

2.Reducer类

`WordReducer`类负责将相同的单词聚合，并计算出每个单词的总次数。

在`cleanup()`中，代码使用了`TreeMap`来对单词根据出现次数从高到低进行排序。

最后输出频率最高的100个单词。

3.Job配置

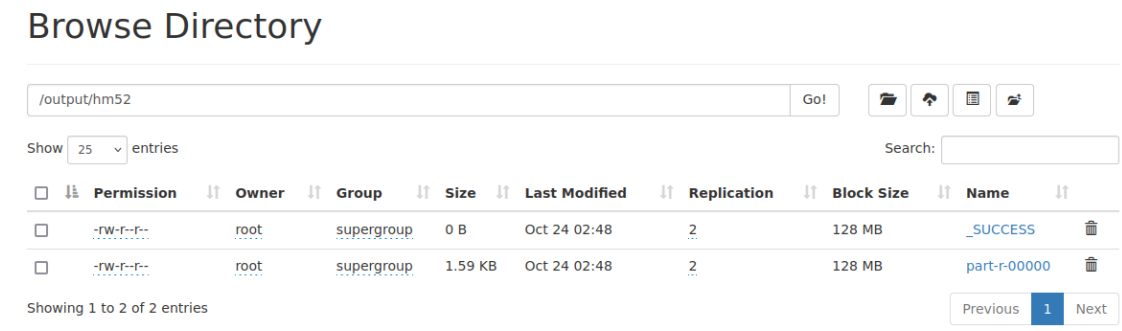
`Job`类用于配置和运行整个MapReduce作业，包括指定输入、输出路径，Mapper和Reducer类的设置。

`FileInputFormat`和`FileOutputFormat`用于定义数据的输入和输出路径。

运行结果

```
root@h01:/usr/local/hadoop# ./bin/hadoop jar Top100WordsCount.jar Top100WordsCou
nt /input/analyst_ratings.csv /output/hm52 /input/stop-word-list.txt
2024-10-23 18:47:42,408 INFO client.DefaultNoHARMFailoverProxyProvider: Connecti
ng to ResourceManager at h01/172.18.0.2:8032
2024-10-23 18:47:43,488 WARN mapreduce.JobResourceUploader: Hadoop command-line
option parsing not performed. Implement the Tool interface and execute your appl
ication with ToolRunner to remedy this.
2024-10-23 18:47:43,531 INFO mapreduce.JobResourceUploader: Disabling Erasure Co
ding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1729703252844_0009
2024-10-23 18:47:44,236 INFO input.FileInputFormat: Total input files to process
 : 1
2024-10-23 18:47:44,492 INFO mapreduce.JobSubmitter: number of splits:1
2024-10-23 18:47:44,828 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1729703252844_0009
2024-10-23 18:47:44,829 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-10-23 18:47:45,396 INFO conf.Configuration: resource-types.xml not found
2024-10-23 18:47:45,399 INFO resource.ResourceUtils: Unable to find 'resource-ty
pes.xml'.
2024-10-23 18:47:45,658 INFO impl.YarnClientImpl: Submitted application applicat
ion_1729703252844_0009
2024-10-23 18:47:45,802 INFO mapreduce.Job: The url to track the job: http://h01
:8088/proxy/application_1729703252844_0009/
2024-10-23 18:47:45,804 INFO mapreduce.Job: Running job: job_1729703252844_0009
2024-10-23 18:47:58,334 INFO mapreduce.Job: Job job_1729703252844_0009 running i
n uber mode : false
2024-10-23 18:47:58,337 INFO mapreduce.Job:  map 0% reduce 0%
2024-10-23 18:48:17,796 INFO mapreduce.Job:  map 57% reduce 0%
2024-10-23 18:48:23,946 INFO mapreduce.Job:  map 67% reduce 0%
2024-10-23 18:48:29,066 INFO mapreduce.Job:  map 100% reduce 0%
2024-10-23 18:48:42,399 INFO mapreduce.Job:  map 100% reduce 100%
2024-10-23 18:48:43,431 INFO mapreduce.Job: Job job_1729703252844_0009 completed
 successfully
2024-10-23 18:48:43,724 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=113975936
                FILE: Number of bytes written=171581764
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=52465317
                HDFS: Number of bytes written=1624
```

```
1: m    64962
2: stocks       56170
3: vs   55993
4: q    54680
5: est  48362
6: eps  45028
7: shares       39586
8: reports      37355
9: update       31728
10: market      31453
11: earnings    30036
12: sales       27626
13: pt  25113
14: week        23504
15: announces   23104
16: price       22382
17: buy 22147
18: trading     21452
19: downgrades  21426
20: benzingas   20101
21: b   20005
22: raises      19852
23: upgrades    19703
24: target      18989
25: maintains   17993
26: new 16641
27: higher      16625
28: session     15664
29: says        14955
30: moving      14586
31: stock       13639
32: premarket   13560
33: sees        13275
34: estimate    13272
35: midday      13030
36: energy      12427
37: initiates   12128
```



不足和可改进之处

- 问题：`WordReducer`类会将所有单词及其出现次数存入内存中的`TreeMap`进行排序，如果单词数非常多，可能会导致内存不足
- 改进建议：可以使用更高效的算法如Top-K排序算法。比如使用最小堆来维持前100个最大值，可以避免将所有单词都放入内存中排序。