

作业5 实验报告

1 集群环境运行

1.1 准备集群环境

```
1 ~$ sudo docker start h01 h02 h03 h04 h05
2 ~$ sudo docker exec -it h01 /bin/bash
```

将节点全部start，实际只用运行主节点h01

```
1 root@h01:/usr/local/hadoop/sbin# ./start-all.sh
```

开始运行

1.2 将文件导入至hdfs

```
1 ~$ sudo docker cp ~/bigdata_workspace/analyst_ratings.csv e5c3902c27e5:/tmp
2 ~$ sudo docker cp ~/bigdata_workspace/stop-word-list.txt e5c3902c27e5:/tmp
```

使用 `~$ sudo docker cp <需要传输的文件路径> <容器id>:<传输文件在容器中的存放位置>` 将文件从本机暂存至容器h01的根目录下的tmp中（其中容器id由 `sudo docker ps` 查看）

```
1 root@h01:/usr/local/hadoop# ./bin/hdfs dfs -put /tmp/analyst_ratings.csv /input/
2 root@h01:/usr/local/hadoop# ./bin/hdfs dfs -put /tmp/stop-word-list.txt /input/
```

将暂存于tmp下的文件们移动至hdfs下的input中

通过 `./bin/hdfs dfs -ls /input` 查看结果

```
root@h01:/usr/local/hadoop# ./bin/hdfs dfs -ls /input
Found 2 items
-rw-r--r--  2 root supergroup  52462980 2024-10-23 10:02 /input/analyst_rating
s.csv
-rw-r--r--  2 root supergroup    2231 2024-10-23 10:03 /input/stop-word-list
.txt
```

1.3 导入代码

在本地使用maven管理java文件

```
1 ~/bigdata_workspace/hadoop_job$ mvn clean package
```

在终端下运行命令

`clean` 在进行打包之前，先确保没有旧的编译文件

`package` 将项目打包成 `.jar` 文件，放在 `target` 目录下

```
1 | ~$ sudo docker cp ~/bigdata_workspace/hadoop_job/target/hadoop_job-1.0-SNAPSHOT.jar e5c3902c27e5:/usr/local/hadoop
```

将包含Java代码的打包好的.jar传入容器

通过 `root@h01:/usr/local/hadoop# jar tf hadoop_job-1.0-SNAPSHOT.jar` 查看.jar下的文件

```
root@h01:/usr/local/hadoop# jar tf hadoop_job-1.0-SNAPSHOT.jar
META-INF/MANIFEST.MF
META-INF/
com/
com/example/
META-INF/maven/
META-INF/maven/com.example/
META-INF/maven/com.example/hadoop_job/
com/example/StockCount.class
com/example/StockCount$StockReducer.class
com/example/StockCount$StockMapper.class
com/example/App.class
com/example/HighFrequencyWords.class
com/example/HighFrequencyWords$WordMapper.class
META-INF/maven/com.example/hadoop_job/pom.xml
META-INF/maven/com.example/hadoop_job/pom.properties
com/example/HighFrequencyWords$WordReducer.class
```

1.4 运行MapReduce代码

任务1

```
1 | #运行任务1
2 | root@h01:/usr/local/hadoop# ./bin/hadoop jar hadoop_job-1.0-SNAPSHOT.jar
com.example.StockCount /input/analyst_ratings.csv /output/SCout
```

```
root@h01:/usr/local/hadoop# ./bin/hadoop jar hadoop_job-1.0-SNAPSHOT.jar com.example.StockCount /input/analyst_ratings.csv /output/SCout
2024-10-23 19:03:49,771 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at h01/172.18.0.2:8032
2024-10-23 19:03:50,471 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2024-10-23 19:03:50,494 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1729703406329_0010
2024-10-23 19:03:50,846 INFO input.FileInputFormat: Total input files to process : 1
2024-10-23 19:03:51,319 INFO mapreduce.JobSubmitter: number of splits:1
2024-10-23 19:03:51,490 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1729703406329_0010
2024-10-23 19:03:51,490 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-10-23 19:03:51,928 INFO conf.Configuration: resource-types.xml not found
2024-10-23 19:03:51,929 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-10-23 19:03:52,064 INFO impl.YarnClientImpl: Submitted application application_1729703406329_0010
2024-10-23 19:03:52,153 INFO mapreduce.Job: The url to track the job: http://h01:8088/proxy/application_1729703406329_0010/
2024-10-23 19:03:52,155 INFO mapreduce.Job: Running job: job_1729703406329_0010
2024-10-23 19:03:59,442 INFO mapreduce.Job: Job job_1729703406329_0010 running in uber mode : false
2024-10-23 19:03:59,445 INFO mapreduce.Job:  map 0% reduce 0%
2024-10-23 19:04:06,570 INFO mapreduce.Job:  map 100% reduce 0%
```

- 1 root@h01:/usr/local/hadoop# ./bin/hadoop fs -ls /output/wCout
- 2 #查看输出结果
- 3 root@h01:/usr/local/hadoop# ./bin/hadoop fs -cat /output/SCout/part-r-00000

```
root@h01:/usr/local/hadoop# ./bin/hadoop fs -ls /output/WCout
Found 2 items
-rw-r--r--  2 root supergroup          0 2024-10-23 18:28 /output/WCout/_SUCCESS
-rw-r--r--  2 root supergroup    1904 2024-10-23 18:28 /output/WCout/part-r-000000
root@h01:/usr/local/hadoop# ./bin/hadoop fs -cat /output/SCout/part-r-000000
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
```

Browse Directory

展示web界面如下:

localhost:9870/explorer.html#/output/SCout

HadoopOverviewDatanodesDatanode Volume FailuresSnapshotStartup ProgressUtilities

Browse Directory

/output/SCout

Go!

Show25entries

Search:

<input type="checkbox"/>	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
<input type="checkbox"/>	-rw-r--r--	root	supergroup	0 B	Oct 24 03:04	2	128 MB	_SUCCESS	<div></div>
<input type="checkbox"/>	-rw-r--r--	root	supergroup	90.37 KB	Oct 24 03:04	2	128 MB	part-r-000000	<div></div>

Showing 1 to 2 of 2 entries

Previous

1

Next

Hadoop, 2024.

任务2

与任务1类似，仅展示结果

```
root@h01:/usr/local/hadoop# ./bin/hadoop jar hadoop_job-1.0-SNAPSHOT.jar com.example.WordCount /input/analyst_ratings.csv /output/WCout /input/stop-word-list.txt
2024-10-23 17:47:11,484 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at h01/172.18.0.2:8032
2024-10-23 17:47:11,890 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2024-10-23 17:47:11,909 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1729703406329_0004
2024-10-23 17:47:12,278 INFO input.FileInputFormat: Total input files to process : 1
2024-10-23 17:47:12,491 INFO mapreduce.JobSubmitter: number of splits:1
2024-10-23 17:47:12,657 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1729703406329_0004
2024-10-23 17:47:12,657 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-10-23 17:47:12,843 INFO conf.Configuration: resource-types.xml not found
2024-10-23 17:47:12,844 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-10-23 17:47:12,925 INFO impl.YarnClientImpl: Submitted application application_1729703406329_0004
2024-10-23 17:47:12,963 INFO mapreduce.Job: The url to track the job: http://h01:8088/proxy/application_1729703406329_0004/
2024-10-23 17:47:12,964 INFO mapreduce.Job: Running job: job_1729703406329_0004
2024-10-23 17:47:19,076 INFO mapreduce.Job: Job job_1729703406329_0004 running in uber mode : false
```

```

root@h01:/usr/local/hadoop# ./bin/hadoop fs -cat /output/WCout/part-r-00000
1 : s , 76104
2 : stocks , 54702
3 : q , 48952
4 : market , 39551
5 : eps , 38003
6 : vs , 36796
7 : m , 36600
8 : shares , 36291
9 : reports , 33653
10 : update , 31535
11 : est , 30384
12 : earnings , 27821
13 : benzinga , 25589
14 : week , 23346
15 : mid , 21332
16 : trading , 20512
17 : buy , 20040
18 : upgrades , 19499
19 : maintains , 16435
20 : downgrades , 16098
21 : higher , 15627
22 : day , 15184
23 : new , 15072

```

Browse Directory

localhost:9870/explorer.html#/output/WCout
110%

Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities

Browse Directory

Show entries
Search:

<input type="checkbox"/>	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
<input type="checkbox"/>	-rw-r--r--	root	supergroup	0 B	Oct 24 02:28	2	128 MB	_SUCCESS	
<input type="checkbox"/>	-rw-r--r--	root	supergroup	1.86 KB	Oct 24 02:28	2	128 MB	part-r-00000	

Showing 1 to 2 of 2 entries

Previous
1
Next

Hadoop, 2024.

1.5输出结果传回主机

```

1 #暂存到tmp下
2 root@h01:/usr/local/hadoop# ./bin/hdfs dfs -get /output/WCout/part-r-00000 /tmp
3 #从h01移动到主机的bigdata_workspace文件夹下
4 sudo docker cp e5c3902c27e5:/tmp/part-r-00000 ~/bigdata_workspace

```

3 设计思路与改进

任务一

思路

代码目的是从 CSV 文件中统计上市公司股票代码（“stock” 列）的出现次数，并按出现次数从大到小输出。主要包含三个部分：Mapper、Reducer 和 Driver 类。

1. Mapper 类（StockMapper）：

- 继承自 `Mapper`，输入为每一行数据。
- 将每一行按逗号分割成列，并检查是否存在有效的股票代码。
- 对于每个有效的股票代码，输出键（股票代码）和值（1），以便在 Reducer 中进行统计。

2. Reducer 类（StockReducer）：

- 继承自 `Reducer`，将 Mapper 输出的股票代码聚合。
- 使用 `HashMap occurrencesMap` 记录每个股票代码的出现次数。
- 在 `cleanup` 方法中，对结果进行排序，并按格式输出排名、股票代码和出现次数。

3. Driver 类（main）：

- 设置作业的配置，包括 Mapper 和 Reducer 的类，以及输入输出路径。
- 启动 MapReduce 作业。

改进

1. 性能不足

- 排序效率：
 - 目前使用 `List` 进行排序，在数据量较大时可能会很慢。考虑使用更高效的数据结构，比如 `PriorityQueue`，来优化排序操作。
- 内存使用：
 - 在 Reducer 中使用了一个 `HashMap` 来存储所有股票代码及其出现次数，可能导致内存使用量过大，尤其在处理大规模数据集时。考虑使用更为紧凑的存储结构，但暂时还不知道可以如何改进。

2. 扩展性不足

- 硬编码列索引：
 - 获取股票代码时，`columns.length == 4 && !columns[3].equals("stock")` 假设数据有固定的列数，这降低了代码的灵活性。可以使用配置文件或参数化输入来动态指定要处理的列索引。
- 缺乏容错处理：
 - 代码缺少对输入数据的异常情况（如格式不正确、缺失值等）进行处理。可以适当增加异常性处理。

任务二

思路

代码用于从上市公司热点新闻标题数据集中统计高频单词。具体步骤包括读取停词列表、处理标题、统计单词出现次数，并输出出现频率最高的前 100 个单词。主要包含以下几个部分：

1. Mapper 类（wordMapper）：

- 在 `setup` 方法中，读取停词文件，将其加载到内存中的集合 `stopWords` 中，以便在后续的单词统计中进行过滤。
- 在 `map` 方法中，从输入的每一行中提取“headline”列，将其转换为小写并清理标点符号，分割成单词。
- 对于每个不在停词列表中的单词，输出该单词和计数（1）。

2. Reducer 类（wordReducer）：

- 在 `reduce` 方法中，接收相同单词的计数，将其累加并存储到 `wordCountMap` 中。
- 在 `cleanup` 方法中，按出现次数对 `wordCountMap` 进行排序，并输出前 100 个高频单词及其出现次数。

3. 主类（main 方法）：

- 配置和设置 MapReduce 作业，指定输入输出路径，并提交作业。

改进

1. 性能不足

• 内存使用：

- 当前实现将所有单词及其计数存储在内存中，如果数据集非常大，可能导致内存溢出。可以考虑使用外部排序或数据库存储，以处理大量数据。

• 停词加载效率：

- 停词列表在每个 Mapper 实例中加载，如果有多个 Mapper 实例，会造成多次重复读取。可以考虑使用分布式缓存（Distributed Cache）将停词文件传递给所有 Mapper，从而提高效率。

2. 扩展性不足

• 硬编码路径：

- 停词文件路径被硬编码为 `hdfs://h01:9000/input/stop-word-list.txt`，这降低了代码的灵活性。可以将其改为通过命令行参数传递，使程序更灵活。