

بسم الله الرحمن الرحيم

گروه 8

تکلیف hadoop map reduce

اعضا:

امیر ارسلان یوری 9830253

محمد مهدی برقی 9818453

الف) کد نوشته شده به زبان پایتون در پیوست این داکيومنت قرار داده شده است.

ب) مقدار reducer به صورت دیفالت در hadoop مقدار 1 را دارد و پارامتر shuffled_map مقدار برابر تعداد map * تعداد reducer را دارد. که همانطور که در خروجی ملاحظه می‌کنید shuffled_map برابر 1 است و این به این دلیل است که با توجه به سایز فایل که برابر 64 مگابایت می‌باشد، همچنین بلاک‌سایز که برابر 128 مگابایت می‌باشد (با استفاده از دستور زیر بلاک سایز را بر حسب مگابایت محاسبه کردیم)

```
echo $((`hadoop fs -stat %o /tmpFile.txt` / (1024*1024) ))
```

بنابراین تعداد مپر ما برابرگرد شده‌ی 64/128 یا همان یک خواهد بود.
همینطور این قضیه را با مقدار Shuffled Maps که برابر 1 بود نیز مطابقت دادیم؛ پس مقدار Shuffled Maps با این عدد تصدیق می‌شود.

خروجی اجرای کد:

```
hadoop@master:~/mew$ hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.3.4.jar -file main.py -mapper mapper.py -
reducer reducer.py -input /tmpFile.txt -output /output-kest
2023-01-27 20:47:20,646 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [main.py] [/tmp/streamjob3932667401473970253.jar tmpDir=null]
2023-01-27 20:47:21,572 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2023-01-27 20:47:21,701 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2023-01-27 20:47:21,701 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2023-01-27 20:47:21,714 WARN impl.MetricsSystemImpl: JobTracker metrics system already initialized!
2023-01-27 20:47:21,906 INFO mapred.FileInputFormat: Total input files to process : 1
2023-01-27 20:47:21,956 INFO mapreduce.JobSubmitter: number of splits:1
2023-01-27 20:47:22,065 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local689182232_0001
2023-01-27 20:47:22,065 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-01-27 20:47:22,224 INFO mapred.LocalDistributedCacheManager: Localized file:/home/hadoop/mew/main.py as file:/tmp/hadoop-hadoop/
mapred/local/job_local689182232_0001_11b1f47e-7ed3-4a64-854a-14d90f11e344/main.py
2023-01-27 20:47:22,331 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2023-01-27 20:47:22,333 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2023-01-27 20:47:22,334 INFO mapreduce.Job: Running job: job_local689182232_0001
2023-01-27 20:47:22,336 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapred.FileOutputCommitter
2023-01-27 20:47:22,342 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2023-01-27 20:47:22,342 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:f
alse, ignore cleanup failures: false
2023-01-27 20:47:22,416 INFO mapred.LocalJobRunner: Waiting for map tasks
2023-01-27 20:47:22,420 INFO mapred.LocalJobRunner: Starting task: attempt_local689182232_0001_m_0000000_0
2023-01-27 20:47:22,447 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2023-01-27 20:47:22,447 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:f
alse, ignore cleanup failures: false
2023-01-27 20:47:22,464 INFO mapred.Task: Using ResourceCalculatorProcessTree : [ ]
2023-01-27 20:47:22,469 INFO mapred.MapTask: Processing split: hdfs://master:9000/tmpFile.txt:0+67185420
2023-01-27 20:47:22,491 INFO mapred.MapTask: numReduceTasks: 1
2023-01-27 20:47:22,592 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2023-01-27 20:47:22,592 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2023-01-27 20:47:22,592 INFO mapred.MapTask: soft limit at 83886080
2023-01-27 20:47:22,592 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2023-01-27 20:47:22,592 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2023-01-27 20:47:22,595 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2023-01-27 20:47:22,604 INFO streaming.PipeMapRed: PipeMapRed exec [/home/hadoop/mew/./mapper.py]
2023-01-27 20:47:22,608 INFO Configuration.deprecation: mapred.work.output.dir is deprecated. Instead, use mapreduce.task.output.dir

HDFS: Number of bytes read=134370840
HDFS: Number of bytes written=16757
HDFS: Number of read operations=15
HDFS: Number of large read operations=0
HDFS: Number of write operations=4
HDFS: Number of bytes read erasure-coded=0

Map-Reduce Framework
  Map input records=28
  Map output records=9915200
  Map output bytes=87015794
  Map output materialized bytes=106846200
  Input split bytes=82
  Combine input records=0
  Combine output records=0
  Reduce input groups=1102
  Reduce shuffle bytes=106846200
  Reduce input records=9915200
  Reduce output records=1102
  Spilled Records=29745600
  Shuffled Maps =1
  Failed Shuffles=0
  Merged Map outputs=1
  GC time elapsed (ms)=30
  Total committed heap usage (bytes)=837812224

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=67185420
File Output Format Counters
  Bytes Written=16757
2023-01-27 20:48:00,416 INFO streaming.StreamJob: Output directory: /output-kest
hadoop@master:~/mew$ hdfs dfsadmin -report
```

(ج) مطابق زیر ده کپی از فایل ایجاد کردیم (9 تا کپی با خود فایل میشه ده تا)

```
hadoop@master:~/mew/copy$ ls
tmpFile.txt
hadoop@master:~/mew/copy$ for i in {1..9}; do cp tmpFile.txt tmpFile$i.txt; done
hadoop@master:~/mew/copy$ ls
tmpFile1.txt  tmpFile3.txt  tmpFile5.txt  tmpFile7.txt  tmpFile9.txt
tmpFile2.txt  tmpFile4.txt  tmpFile6.txt  tmpFile8.txt  tmpFile.txt
hadoop@master:~/mew/copy$
```



```
hadoop@master:~/mew$ hdfs dfs -ls /
Found 15 items
drwxr-xr-x - hadoop supergroup 0 2023-01-27 17:45 /input-mr
drwxr-xr-x - hadoop supergroup 0 2022-12-23 00:35 /input_spark
drwxr-xr-x - hadoop supergroup 0 2023-01-27 19:58 /output
-rw-r--r-- 1 hadoop supergroup 67185420 2023-01-27 21:36 /tmpFile1.txt
-rw-r--r-- 1 hadoop supergroup 67185420 2023-01-27 18:52 /tmpFile10.txt
-rw-r--r-- 1 hadoop supergroup 67185420 2023-01-27 21:36 /tmpFile2.txt
-rw-r--r-- 1 hadoop supergroup 67185420 2023-01-27 21:36 /tmpFile3.txt
-rw-r--r-- 1 hadoop supergroup 67185420 2023-01-27 21:37 /tmpFile4.txt
-rw-r--r-- 1 hadoop supergroup 67185420 2023-01-27 21:37 /tmpFile5.txt
-rw-r--r-- 1 hadoop supergroup 67185420 2023-01-27 21:37 /tmpFile6.txt
-rw-r--r-- 1 hadoop supergroup 67185420 2023-01-27 21:37 /tmpFile7.txt
-rw-r--r-- 1 hadoop supergroup 67185420 2023-01-27 21:37 /tmpFile8.txt
-rw-r--r-- 1 hadoop supergroup 67185420 2023-01-27 21:37 /tmpFile9.txt
drwxr-xr-x - hadoop supergroup 0 2023-01-27 17:45 /user
drwxr-xr-x - hadoop supergroup 0 2023-01-27 01:03 /wcp
hadoop@master:~/mew$ for i in {1..10}; do hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.3.4.jar -file main.py -mapper mapper.py -reducer reducer.py -input /tmpFile$i.txt -output /output$i &; done
```



بعد هم با فرمت عکس زیر اجراشون کردیم:


```
hadoop@master:~/mew$ for i in {1..10}; do
> hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.3.4.jar -file main.py -mapper mapper.py -reducer reducer.py -input /tmpFile$i.txt -output /output$i &
> done
> done
[1] 112836
[2] 112837
[3] 112838
[4] 112839
[5] 112840
[6] 112841
[7] 112842
[8] 112843
[9] 112844
[10] 112848
```

برای هر یک مثل بخش قبل یک میپر ایجاد شده است که مجموعاً 10 میپر و 10 رییدیوسر اجرا شده است. و این نشان می‌دهد که به ازای هر فایل که کوچک‌تر از block size باشد یک mapper و یک reducer ایجاد می‌گردد.

(د) مطابق تصویر زیر فایل 640 مگابایتی را ایجاد کردیم:

```
hadoop@master: ~/mew/copy X + v
hadoop@master:~/mew/copy$ cat tmpFile{1..10}.txt >> tmpBigFile.txt
hadoop@master:~/mew/copy$ ls
tmpBigFile.txt tmpFile1.txt tmpFile3.txt tmpFile5.txt tmpFile7.txt tmpFile9.txt
tmpFile10.txt tmpFile2.txt tmpFile4.txt tmpFile6.txt tmpFile8.txt
hadoop@master:~/mew/copy$ history | grep copy
1628 hdfs dfs -copyFromLocal ./tmpFile.txt group8-wordcount
1629 hdfs dfs -copyFromLocal ./tmpFile.txt /group8-wordcount
1699 hdfs dfs -copyFromLocal ./tmpFile.txt /word_count_in_python
1733 hdfs dfs -copyFromLocal ./tmpFile.txt /wcp
1819 hdfs dfs -copyFromLocal tmpFile.txt /
1845 ls copy/
1859 mkdir copy
1860 cp ../hadoop-group8/tmpFile.txt copy/
1861 cd copy/
1880 ls copy/
1897 cd copy/
1906 history | grep copy
hadoop@master:~/mew/copy$ hdfs dfs -copyFromLocal tmpBigFile.txt /tmpBigFile.txt
```

(ز) هر بخش به توضیح دانی... و توابع هر حالت مختلف سایر فایل سیستم را نگاه می‌کنی
که رویای نیست (و این نگاه سایر را هم در قسمت به پوشه‌های (کتابخانه) می‌نویسد)

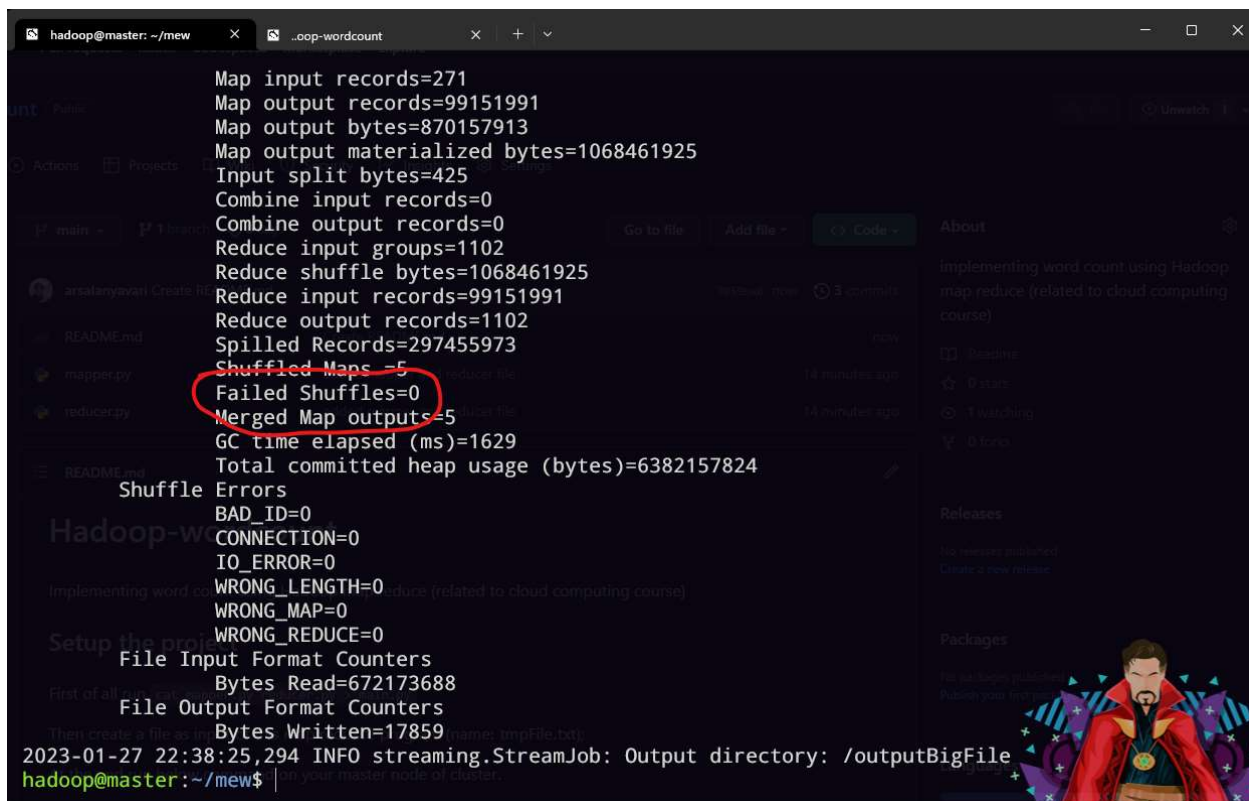


طبق این تصویر و کد اجرا شده می‌توانیم به نتیجه برسیم که با توجه به اینکه همچنان تعداد reducer های ما برابر 1 می‌باشد پس تعداد 5، mapper داریم (600 تقسیم بر 128 که به بالا گرد می‌شود). که این بدلیل این است که فایل ما مقداری بزرگتر از block size که برابر 128 MB بود، دارد.

زیرا طبق بررسی‌های ما در hadoop به ازای هر chunk از فایل یک mapper ساخته می‌شود.

و به صورت کلی این قانون برقرار است که به ازای هر فایل کوچکتر از block size یک mapper و به اندازه تعیین شده reducer ایجاد می‌گردد.

و اگر یک فایل سائیزی بزرگ تر block size داشته باشد به ازای هر بلاک یک mapper جدید ایجاد و برای کل آن فایل به تعداد تعیین شده در کانفیگ (به صورت پیش‌فرض 1 عدد) reducer وجود دارد.



```
hadoop@master: ~/mew
oop-wordcount

Map input records=271
Map output records=99151991
Map output bytes=870157913
Map output materialized bytes=1068461925
Input split bytes=425
Combine input records=0
Combine output records=0
Reduce input groups=1102
Reduce shuffle bytes=1068461925
Reduce input records=99151991
Reduce output records=1102
Spilled Records=297455973
Shuffled Maps=5
Failed Shuffles=0
Merged Map outputs=5
GC time elapsed (ms)=1629
Total committed heap usage (bytes)=6382157824

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=672173688
File Output Format Counters
Bytes Written=17859
2023-01-27 22:38:25,294 INFO streaming.StreamJob: Output directory: /outputBigFile
hadoop@master:~/mew$
```


ه) همچنان با توجه به مقدار دیفالت reducer ما تنها یک reducer داریم و با توجه به shuffled_map که با توجه به مقدار $\text{reducer} = 1$ نشان دهنده تعداد mapper ها است. ما برای یک فایل 300 MB با توجه به $\text{block size} = 128\text{MB}$ داریم: سقف 300 تقسیم بر 128 که برابر 3 است و دقیقا منطبق بر تعداد shuffled_map پس بنابراین ما در این سوال 3 mapper داریم.

و) با استفاده از این کامند می‌توان تعداد mapper ها را تغییر داد که طبق توضیحات داده شده در ورژن جدید hadoop این تغییر در تعداد mapper ها در نظر گرفته نشده و طبق فرمول قبلی تعداد محاسبه می‌گردد دستور بدین صورت است:

```
hadoop jar /path/to/hadoop-streaming.jar -D mapred.map.tasks=10 -mapper my_mapper.py -reducer my_reducer.py -input /input/path -output /output/path
```

و برای تغییر reducer ها از این دستور استفاده می‌گردد که بر روی تعداد reducer ها تاثیرگذار است:

```
hadoop jar /path/to/hadoop-streaming.jar -D mapred.reduce.tasks=5 -mapper my_mapper.py -reducer my_reducer.py -input /input/path -output /output/path
```



با توجه به اینکه ما از کد پایتون استفاده کرده‌ایم کدی تحت عنوان درایور نداریم اما یک کد پایتون برای اجرای دستور مناسب نوشتیم که به پیوست ارسال شده است.

ز) این سوال به صورت کامل در بخش د توضیح داده شده است.
و نکته قابل توجه در رابطه با تعداد mapper ها این است که طبق ویدیو ضبط شده در ورژن‌های جدید hadoop تغییر دستی تعداد mapper در نظر گرفته نمی‌شود و با توجه به فرمول ذکر شده تعداد mapper ها مشخص می‌گردد.