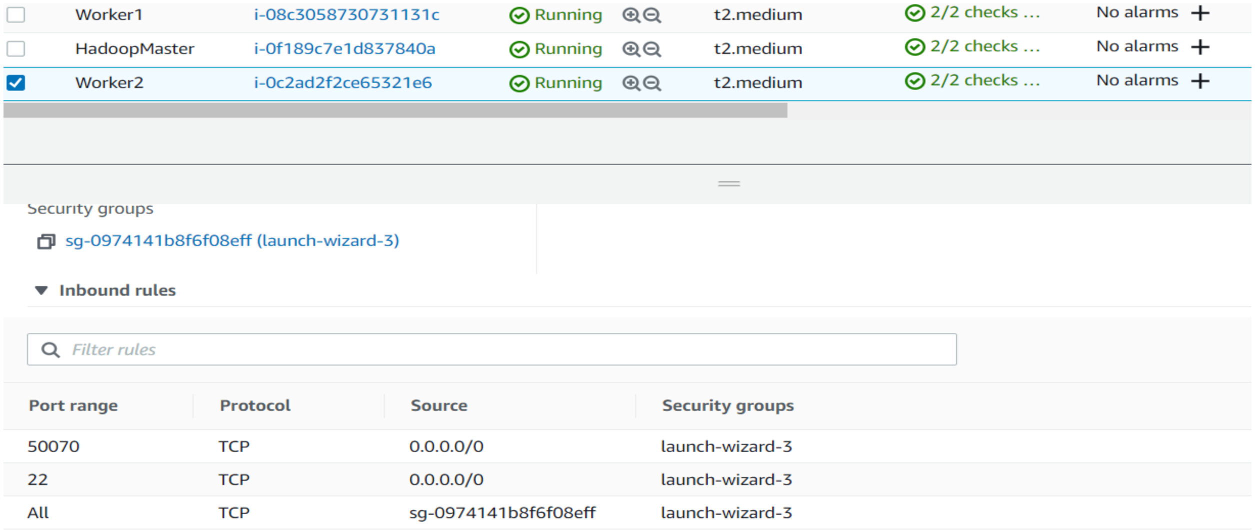
Azat Dovgeldiyev

CSC 555

Midterm

Part 1 Multi Node Cluster.



☐	Worker1	i-08c3058730731131c	✓ Running	t2.medium	✓ 2/2 checks ...	No alarms +
☐	HadoopMaster	i-0f189c7e1d837840a	✓ Running	t2.medium	✓ 2/2 checks ...	No alarms +
☑	Worker2	i-0c2ad2f2ce65321e6	✓ Running	t2.medium	✓ 2/2 checks ...	No alarms +

Security groups

sg-0974141b8f6f08eff (launch-wizard-3)

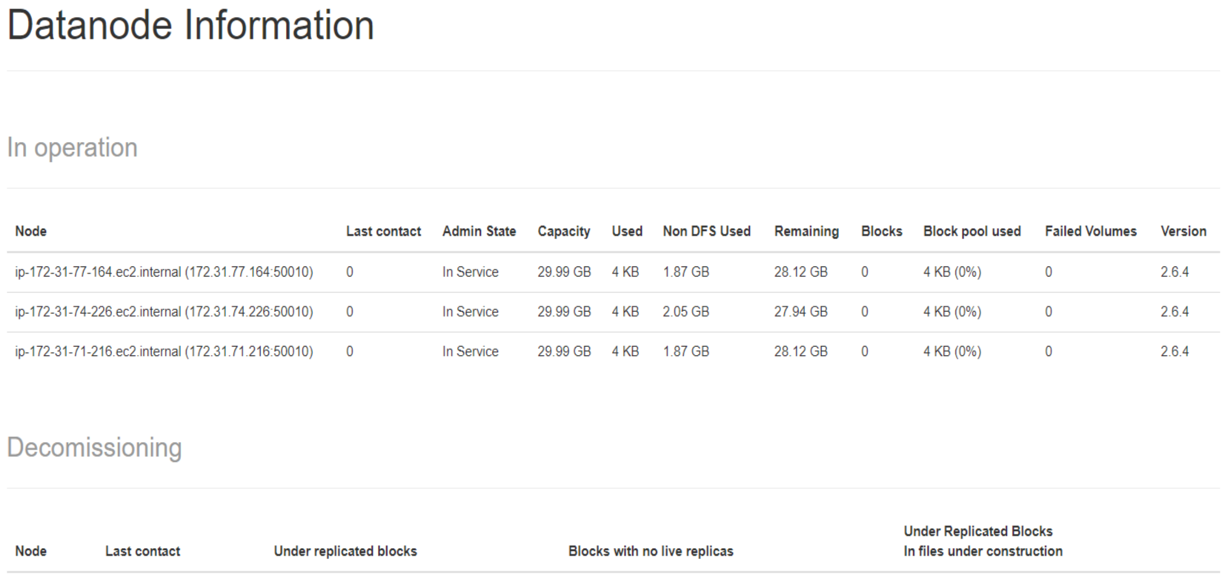
▼ Inbound rules

Filter rules

Port range	Protocol	Source	Security groups
50070	TCP	0.0.0.0/0	launch-wizard-3
22	TCP	0.0.0.0/0	launch-wizard-3
All	TCP	sg-0974141b8f6f08eff	launch-wizard-3

You should verify that the cluster is running by pointing your browser to the link below.

http://[insert-the-public-ip-of-master]:50070/



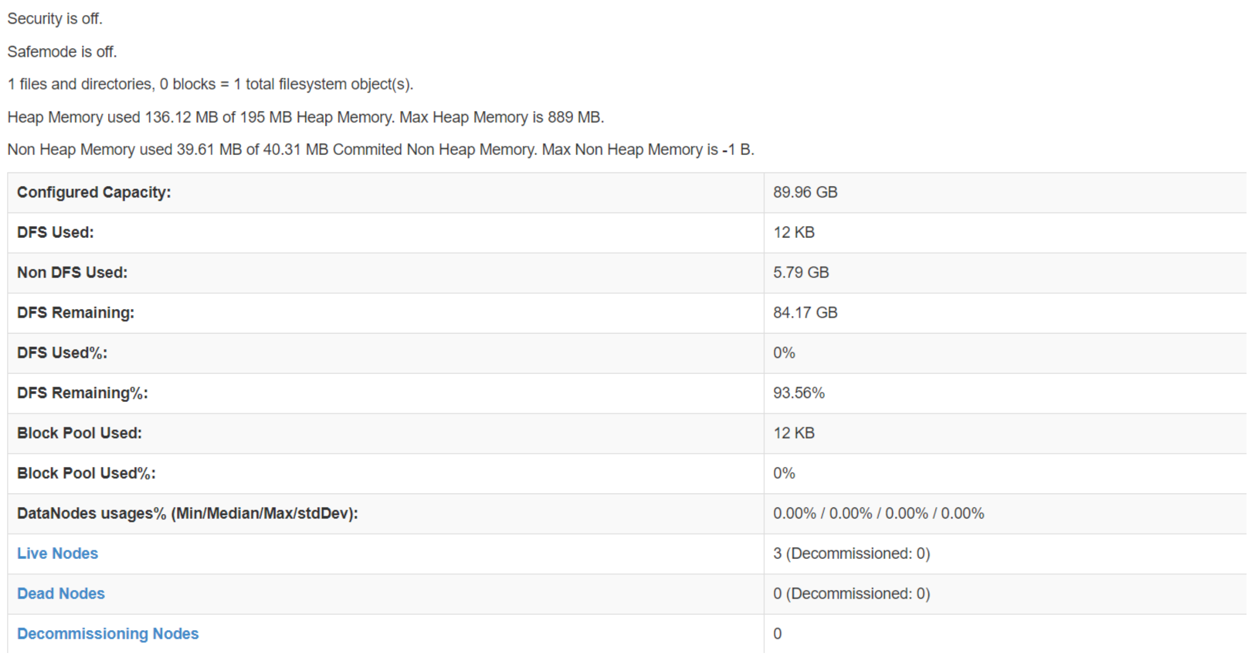
Datanode Information

In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
ip-172-31-77-164.ec2.internal (172.31.77.164:50010)	0	In Service	29.99 GB	4 KB	1.87 GB	28.12 GB	0	4 KB (0%)	0	2.6.4
ip-172-31-74-226.ec2.internal (172.31.74.226:50010)	0	In Service	29.99 GB	4 KB	2.05 GB	27.94 GB	0	4 KB (0%)	0	2.6.4
ip-172-31-71-216.ec2.internal (172.31.71.216:50010)	0	In Service	29.99 GB	4 KB	1.87 GB	28.12 GB	0	4 KB (0%)	0	2.6.4

Decomissioning

Node	Last contact	Under replicated blocks	Blocks with no live replicas	Under Replicated Blocks In files under construction



Security is off.

Safemode is off.

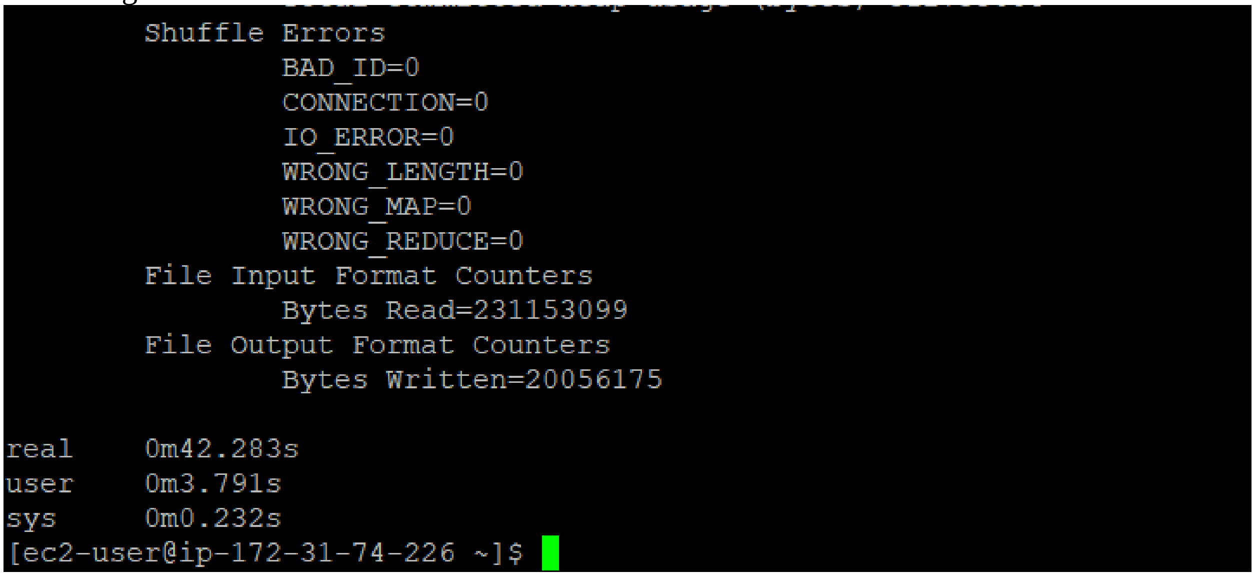
1 files and directories, 0 blocks = 1 total filesystem object(s).

Heap Memory used 136.12 MB of 195 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 39.61 MB of 40.31 MB Commited Non Heap Memory. Max Non Heap Memory is -1 B.

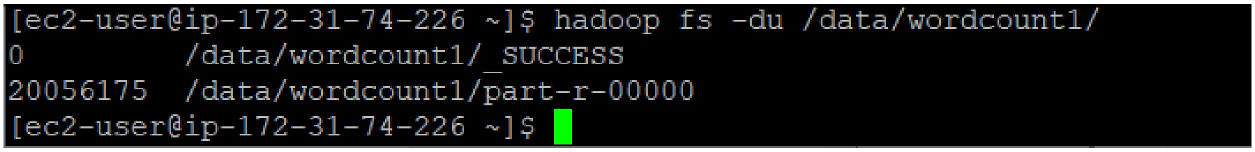
Configured Capacity:	89.96 GB
DFS Used:	12 KB
Non DFS Used:	5.79 GB
DFS Remaining:	84.17 GB
DFS Used%:	0%
DFS Remaining%:	93.56%
Block Pool Used:	12 KB
Block Pool Used%:	0%
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	3 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0

Repeat the steps for wordcount using bioproject.xml from Assignment 1 and submit screenshots of running it.

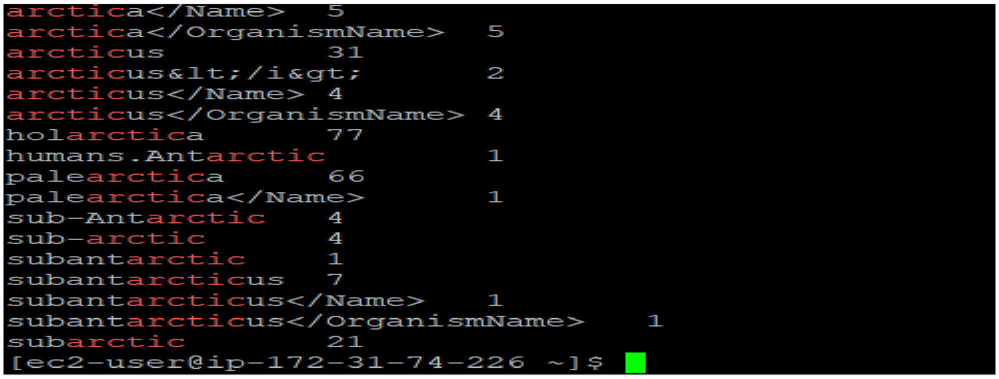


```
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=231153099
        File Output Format Counters
                Bytes Written=20056175

real    0m42.283s
user    0m3.791s
sys     0m0.232s
[ec2-user@ip-172-31-74-226 ~]$
```

Size of wordcount



```
[ec2-user@ip-172-31-74-226 ~]$ hadoop fs -du /data/wordcount1/
0          /data/wordcount1/_SUCCESS
20056175   /data/wordcount1/part-r-00000
[ec2-user@ip-172-31-74-226 ~]$
```

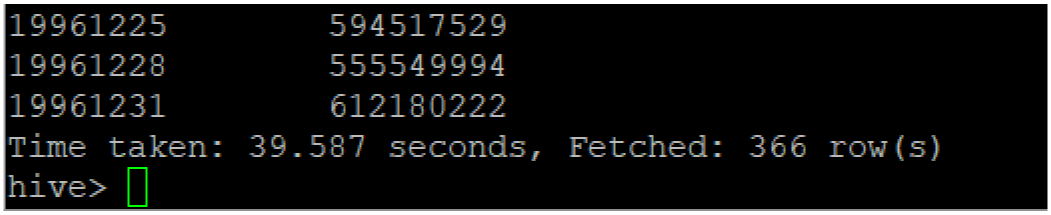


```
arctica</Name>   5
arctica</OrganismName>   5
arcticus             31
arcticus&lt;/i&gt;       2
arcticus</Name>  4
arcticus</OrganismName>  4
holarctica           77
humans.Antarctic         1
palearctica          66
palearctica</Name>       1
sub-Antarctic        4
sub-arctic           4
subantarctic         1
subantarcticus       7
subantarcticus</Name>    1
subantarcticus</OrganismName>    1
subarctic            21
[ec2-user@ip-172-31-74-226 ~]$
```

Comparing to the first assignment where time completed was about 1 minute and 14 seconds, this time it is faster with time completed for 42 seconds, and the reason for that one node is slower than working with 3 nodes.

Part 2: Hive

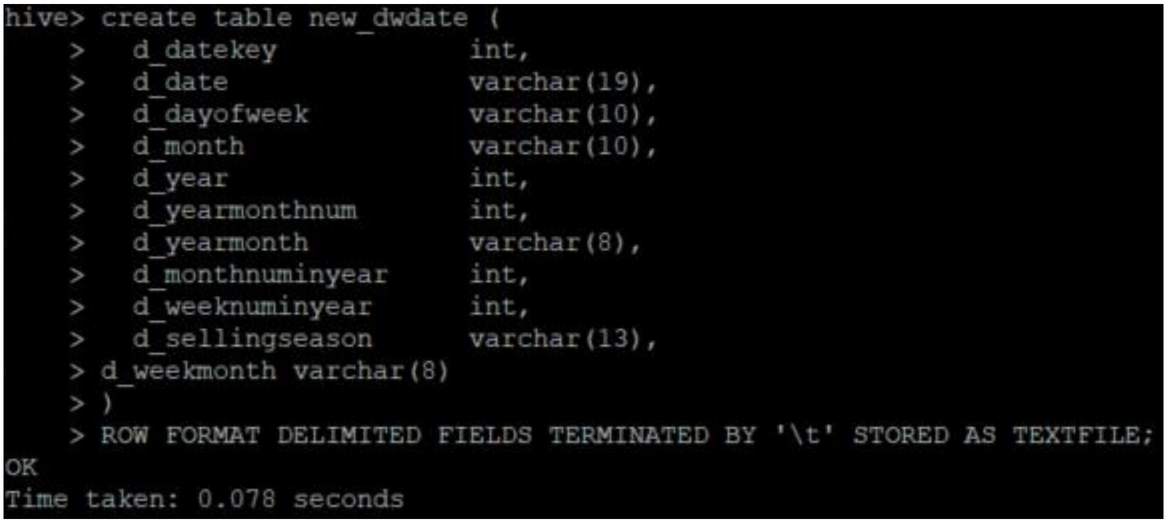
1) Run the following query in Hive and report the time it takes to execute:



```
19961225        594517529
19961228        555549994
19961231        612180222
Time taken: 39.587 seconds, Fetched: 366 row(s)
hive>
```

2) Perform the following transform operation using SELECT TRANSFORM on the dwdate table by creating a new table. The new dwdate table will combine d_daynuminweek, d_daynuminmonth, and d_daynuminyear into a single column in the new table. You should also eliminate of the last 4 columns (d_lastdayinweekfl, d_lastdayinmonthfl, d_holidayfl, and d_weekdayfl). The final table will have 6 fewer columns than the original table because you merge 3 columns into 1 and remove 4 columns.

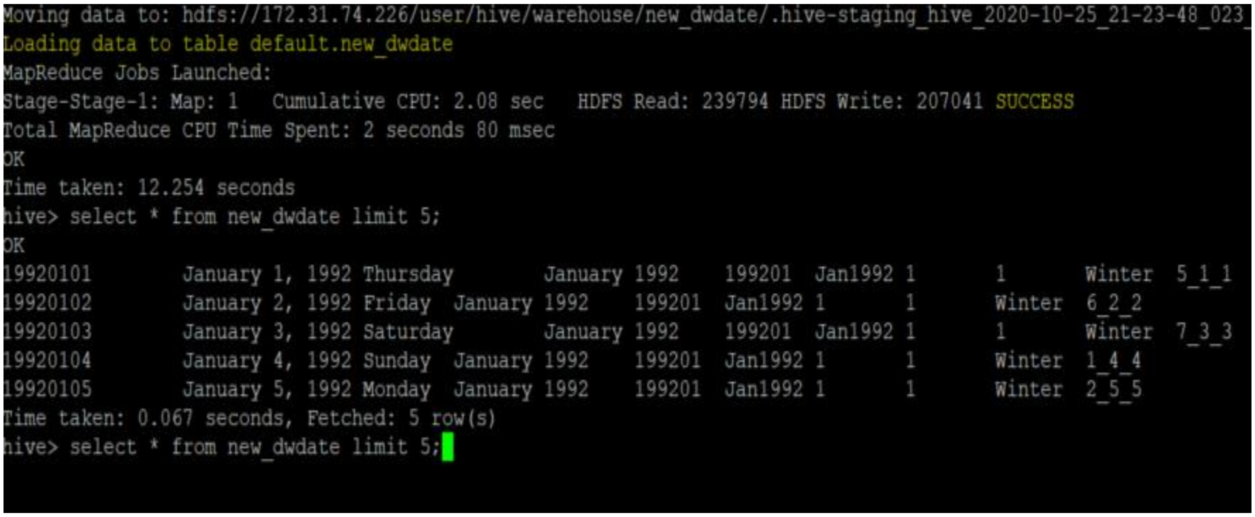
I created a new table , last column d_weekmonth will combine 3 columns:



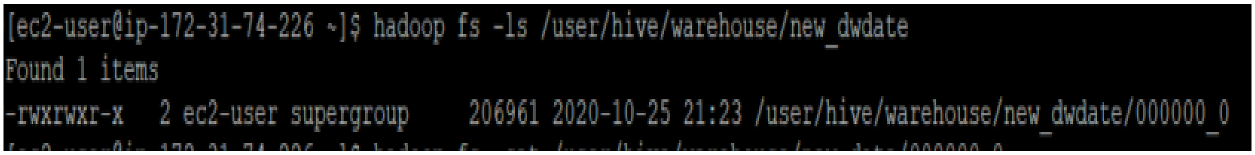
```
hive> create table new_dwdate (
    >   d_datekey               int,
    >   d_date                  varchar(19),
    >   d_dayofweek             varchar(10),
    >   d_month                 varchar(10),
    >   d_year                  int,
    >   d_yearmonthnum          int,
    >   d_yearmonth             varchar(8),
    >   d_monthnuminyear        int,
    >   d_weeknuminyear         int,
    >   d_sellingseason         varchar(13),
    > d_weekmonth varchar(8)
    > )
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' STORED AS TEXTFILE;
OK
Time taken: 0.078 seconds
```

INSERT OVERWRITE TABLE new_dwdate SELECT TRANSFORM (d_datekey, d_date, d_dayofweek, d_month, d_year, d_yearmonthnum,d_yearmonth,d_daynuminweek, d_daynuminmonth, d_daynuminyear, d_monthnuminyear, d_weeknuminyear, d_sellingseason, d_lastdayinweekfl, d_lastdayinmonthfl, d_holidayfl, d_weekdayfl) USING 'python pdate1.py' AS (d_datekey, d_date, d_dayofweek, d_month, d_year, d_yearmonthnum,d_yearmonth, d_monthnuminyear, d_weeknuminyear, d_sellingseason, d_weekmonth) FROM dwdate;

TESTING;

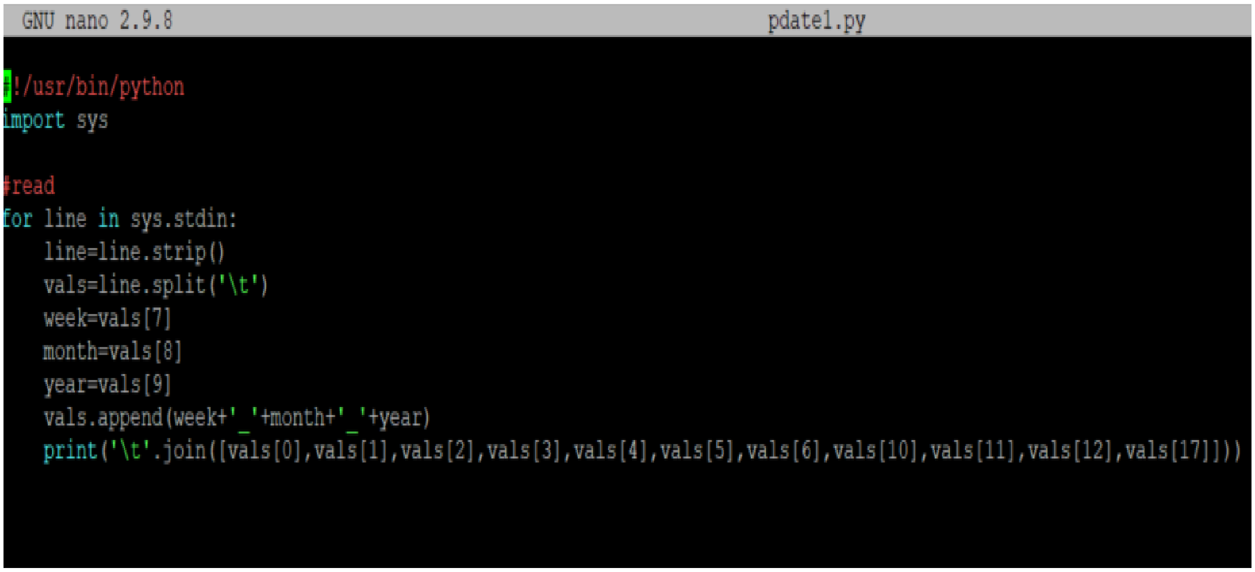


```
Moving data to: hdfs://172.31.74.226/user/hive/warehouse/new_dwdate/.hive-staging_hive_2020-10-25_21-23-48_023_
Loading data to table default.new_dwdate
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 2.08 sec   HDFS Read: 239794 HDFS Write: 207041 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 80 msec
OK
Time taken: 12.254 seconds
hive> select * from new_dwdate limit 5;
OK
19920101        January 1, 1992 Thursday        January 1992    199201  Jan1992 1       1       Winter  5_1_1
19920102        January 2, 1992 Friday  January 1992    199201  Jan1992 1       1       Winter  6_2_2
19920103        January 3, 1992 Saturday        January 1992    199201  Jan1992 1       1       Winter  7_3_3
19920104        January 4, 1992 Sunday  January 1992    199201  Jan1992 1       1       Winter  1_4_4
19920105        January 5, 1992 Monday  January 1992    199201  Jan1992 1       1       Winter  2_5_5
Time taken: 0.067 seconds, Fetched: 5 row(s)
hive> select * from new_dwdate limit 5;
```



```
[ec2-user@ip-172-31-74-226 ~]$ hadoop fs -ls /user/hive/warehouse/new_dwdate
Found 1 items
-rwxrwxr-x   2 ec2-user supergroup     206961 2020-10-25 21:23 /user/hive/warehouse/new_dwdate/000000_0
```

Python code :



```
  GNU nano 2.9.8                                    pdate1.py

#!/usr/bin/python
import sys

#read
for line in sys.stdin:
    line=line.strip()
    vals=line.split('\t')
    week=vals[7]
    month=vals[8]
    year=vals[9]
    vals.append(week+'_'+month+'_'+year)
    print('\t'.join([vals[0],vals[1],vals[2],vals[3],vals[4],vals[5],vals[6],vals[10],vals[11],vals[12],vals[17]]))
```

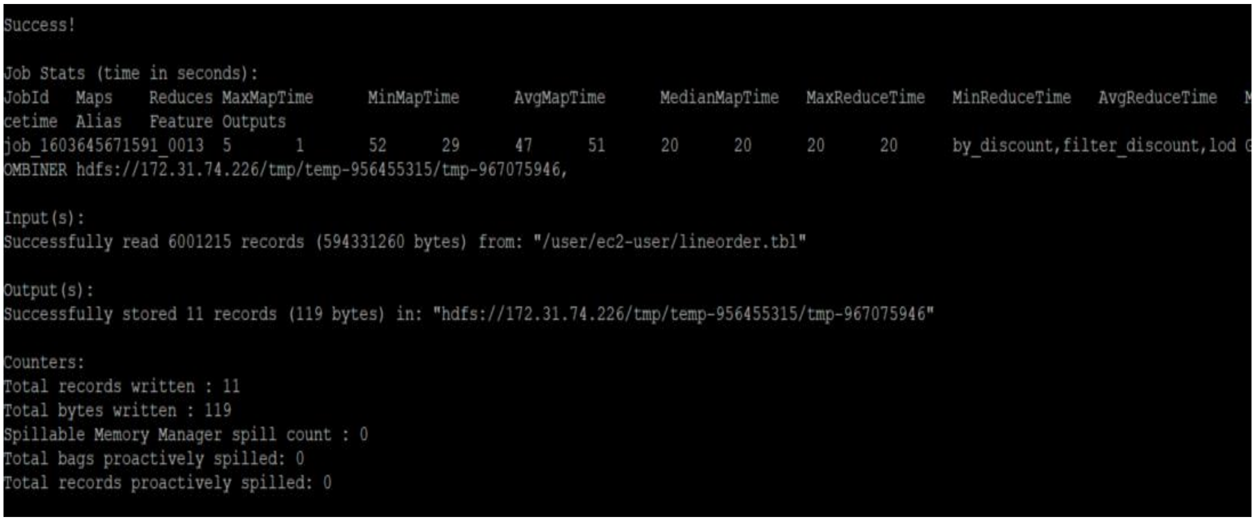
```
#!/usr/bin/python
import sys

#read
for line in sys.stdin:
   line=line.strip()
   vals=line.split('\t')
   week=vals[7]
   month=vals[8]
   year=vals[9]
   vals.append(week+'_'+month+'_'+year)

print('\t'.join([vals[0],vals[1],vals[2],vals[3],vals[4],vals[5],vals[6],vals[10],vals[11],vals[12],vals[17
]]))
```

Part 3: Pig

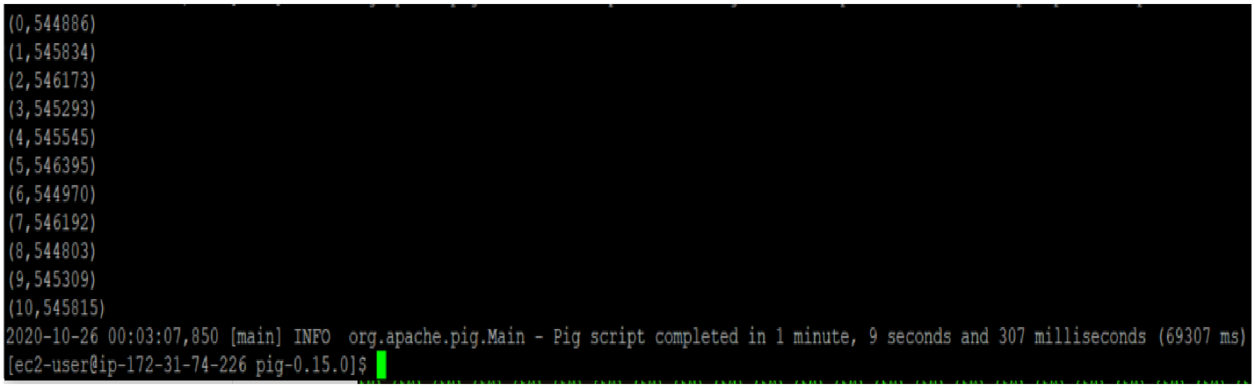
```
SELECT lo_discount, COUNT(lo_extendedprice)
FROM lineorder
GROUP BY lo_discount;
```

```
lod = LOAD '/user/ec2-user/lineorder.tbl' USING PigStorage('|') AS(lo_orderkey:int,lo_linenumber:int,
lo_custkey:int, lo_partkey:int, lo_suppkey:int, lo_orderdate:int, lo_orderpriority:chararray,
lo_shippriority:chararray, lo_quantity:int, lo_extendedprice:int, lo_ordertotalprice:int, lo_discount:int,
lo_revenue:int,lo_supplycost:int, lo_tax:int, lo_commitdate:int, lo_shipmode:chararray);
```

```
by_discount = group lod by lo_discount;
filter_discount = FOREACH by_discount GENERATE group as
lo_discount,COUNT(lod.lo_extendedprice);
dump filter_discount;
```



```
Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime   M
cetime  Alias   Feature Outputs
job_1603645671591_0013  5       1       52      29      47      51      20      20      20      20      by_discount,filter_discount,lod C
OMBINER hdfs://172.31.74.226/tmp/temp-956455315/tmp-967075946,

Input(s):
Successfully read 6001215 records (594331260 bytes) from: "/user/ec2-user/lineorder.tbl"

Output(s):
Successfully stored 11 records (119 bytes) in: "hdfs://172.31.74.226/tmp/temp-956455315/tmp-967075946"

Counters:
Total records written : 11
Total bytes written : 119
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
```

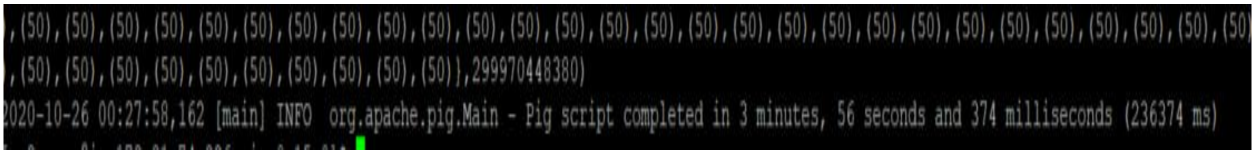


```
(0,544886)
(1,545834)
(2,546173)
(3,545293)
(4,545545)
(5,546395)
(6,544970)
(7,546192)
(8,544803)
(9,545309)
(10,545815)
2020-10-26 00:03:07,850 [main] INFO  org.apache.pig.Main - Pig script completed in 1 minute, 9 seconds and 307 milliseconds (69307 ms)
[ec2-user@ip-172-31-74-226 pig-0.15.0]$
```

Created nano pig_script.pig file wrote scripts then saved and run with **bin/pig -f pig_script.pig**

Completed in : 1 min, 9 secs and 307 msecs (69307ms)

SELECT lo_quantity, SUM(lo_revenue)
FROM lineorder
WHERE lo_discount > 6
GROUP BY lo_quantity;
lod = LOAD '/user/ec2-user/lineorder.tbl' USING PigStorage('|')
AS(lo_orderkey:int,lo_linenumber:int, lo_custkey:int, lo_partkey:int, lo_suppkey:int,$
filter_data= FILTER lod BY lo_discount > 6;
group_quantity = GROUP filter_data by lo_quantity;
filtered = foreach group_quantity generate filter_data.lo_quantity,
SUM(filter_data.lo_revenue);
dump filtered;



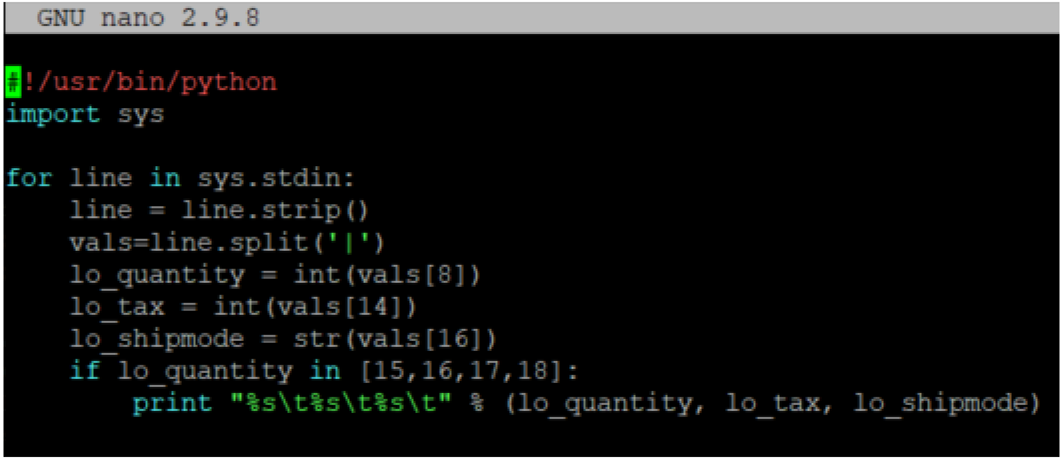
```
,(50),(50),(50),(50),(50),(50),(50),(50),(50),(50),(50),(50),(50),(50),(50),(50),(50),(50),(50),(50),(50),(50),(50),(50),(50),(50),(50),(50),(50
,(50),(50),(50),(50),(50),(50),(50),(50),(50),(50)},299970448380)
2020-10-26 00:27:58,162 [main] INFO  org.apache.pig.Main - Pig script completed in 3 minutes, 56 seconds and 374 milliseconds (236374 ms)
```

Completed in 3 mins, 56 secs and 374 msecs (236374ms)

Part 4: Hadoop Streaming

```
SELECT lo_shipmode, STDDEV(lo_tax)
FROM lineorder
WHERE lo_quantity BETWEEN 15 AND 18
GROUP BY lo_shipmode;
```

myMapper.py

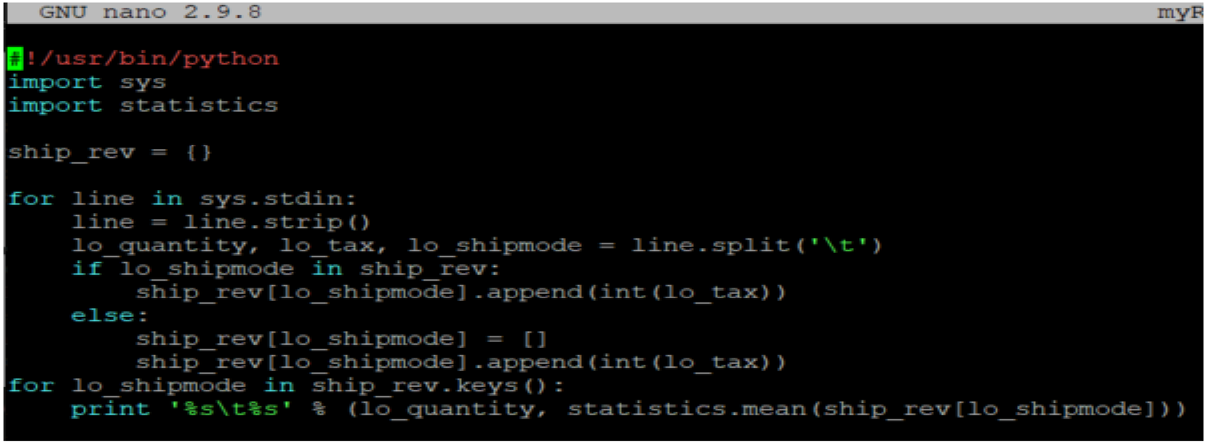


```
#!/usr/bin/python
import sys

for line in sys.stdin:
    line = line.strip()
    vals=line.split('|')
    lo_quantity = int(vals[8])
    lo_tax = int(vals[14])
    lo_shipmode = str(vals[16])
    if lo_quantity in [15,16,17,18]:
        print "%s\t%s\t%s\t" % (lo_quantity, lo_tax, lo_shipmode)
```

```
#!/usr/bin/python
import sys

for line in sys.stdin:
    line = line.strip()
    vals=line.split('|')
    lo_quantity = int(vals[8])
    lo_tax = int(vals[14])
    lo_shipmode = str(vals[16])
    if lo_quantity in [15,16,17,18]:
        print "%s\t%s\t%s\t" % (lo_quantity, lo_tax, lo_shipmode)
```

myReducer.py

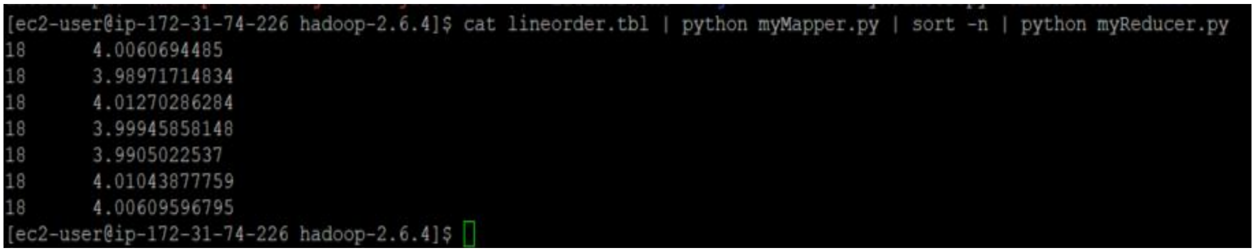
```
#!/usr/bin/python
import sys
import statistics
```



```
#!/usr/bin/python
import sys
import statistics

ship_rev = {}

for line in sys.stdin:
    line = line.strip()
    lo_quantity, lo_tax, lo_shipmode = line.split('\t')
    if lo_shipmode in ship_rev:
        ship_rev[lo_shipmode].append(int(lo_tax))
    else:
        ship_rev[lo_shipmode] = []
        ship_rev[lo_shipmode].append(int(lo_tax))
for lo_shipmode in ship_rev.keys():
    print '%s\t%s' % (lo_quantity, statistics.mean(ship_rev[lo_shipmode]))
```

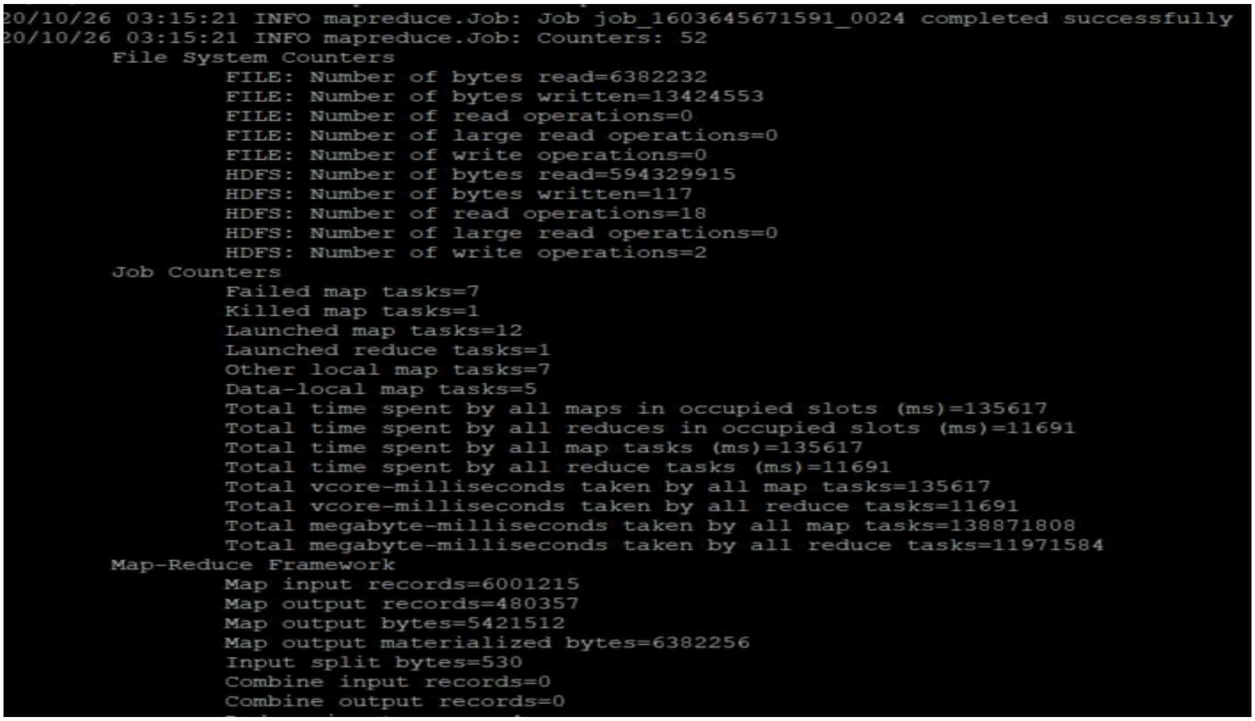
```
ship_rev = {}

for line in sys.stdin:
    line = line.strip()
    lo_quantity, lo_tax, lo_shipmode = line.split('\t')
    if lo_shipmode in ship_rev:
        ship_rev[lo_shipmode].append(int(lo_tax))
    else:
        ship_rev[lo_shipmode] = []
        ship_rev[lo_shipmode].append(int(lo_tax))
for lo_shipmode in ship_rev.keys():
    print '%s\t%s' % (lo_quantity, statistics.mean(ship_rev[lo_shipmode]))
```

Testing

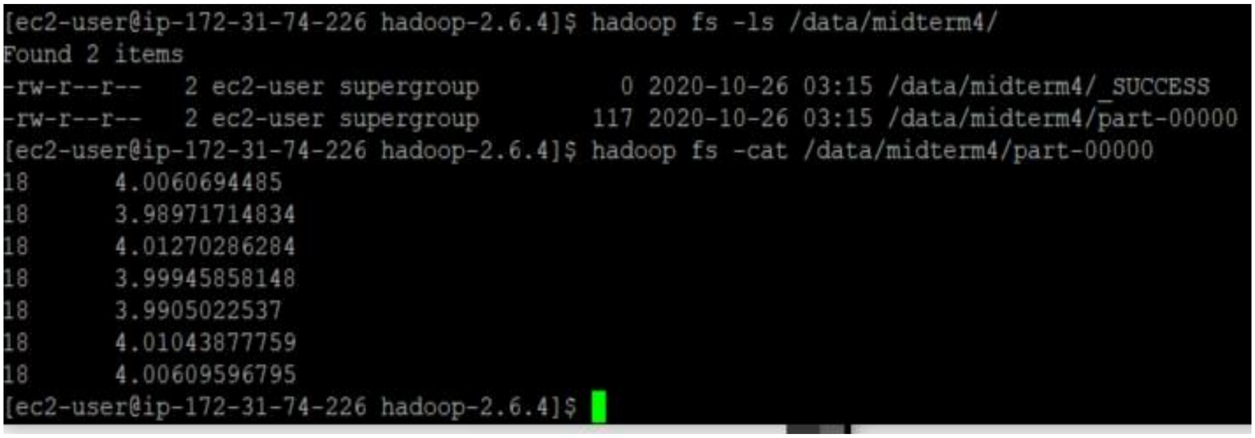


```
[ec2-user@ip-172-31-74-226 hadoop-2.6.4]$ cat lineorder.tbl | python myMapper.py | sort -n | python myReducer.py
18      4.0060694485
18      3.98971714834
18      4.01270286284
18      3.99945858148
18      3.9905022537
18      4.01043877759
18      4.00609596795
[ec2-user@ip-172-31-74-226 hadoop-2.6.4]$
```

Output
hadoop jar hadoop-streaming-2.6.4.jar -input /user/ec2-user/part4 -output /data/midterm4 -mapper myMapper.py -reducer myReducer.py -file myReducer.py -file myMapper.py



```
20/10/26 03:15:21 INFO mapreduce.Job: Job job_1603645671591_0024 completed successfully
20/10/26 03:15:21 INFO mapreduce.Job: Counters: 52
        File System Counters
                FILE: Number of bytes read=6382232
                FILE: Number of bytes written=13424553
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=594329915
                HDFS: Number of bytes written=117
                HDFS: Number of read operations=18
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Failed map tasks=7
                Killed map tasks=1
                Launched map tasks=12
                Launched reduce tasks=1
                Other local map tasks=7
                Data-local map tasks=5
                Total time spent by all maps in occupied slots (ms)=135617
                Total time spent by all reduces in occupied slots (ms)=11691
                Total time spent by all map tasks (ms)=135617
                Total time spent by all reduce tasks (ms)=11691
                Total vcore-milliseconds taken by all map tasks=135617
                Total vcore-milliseconds taken by all reduce tasks=11691
                Total megabyte-milliseconds taken by all map tasks=138871808
                Total megabyte-milliseconds taken by all reduce tasks=11971584
        Map-Reduce Framework
                Map input records=6001215
                Map output records=480357
                Map output bytes=5421512
                Map output materialized bytes=6382256
                Input split bytes=530
                Combine input records=0
                Combine output records=0
```



```
[ec2-user@ip-172-31-74-226 hadoop-2.6.4]$ hadoop fs -ls /data/midterm4/
Found 2 items
-rw-r--r--   2 ec2-user supergroup          0 2020-10-26 03:15 /data/midterm4/_SUCCESS
-rw-r--r--   2 ec2-user supergroup        117 2020-10-26 03:15 /data/midterm4/part-00000
[ec2-user@ip-172-31-74-226 hadoop-2.6.4]$ hadoop fs -cat /data/midterm4/part-00000
18      4.0060694485
18      3.98971714834
18      4.01270286284
18      3.99945858148
18      3.9905022537
18      4.01043877759
18      4.00609596795
[ec2-user@ip-172-31-74-226 hadoop-2.6.4]$
```