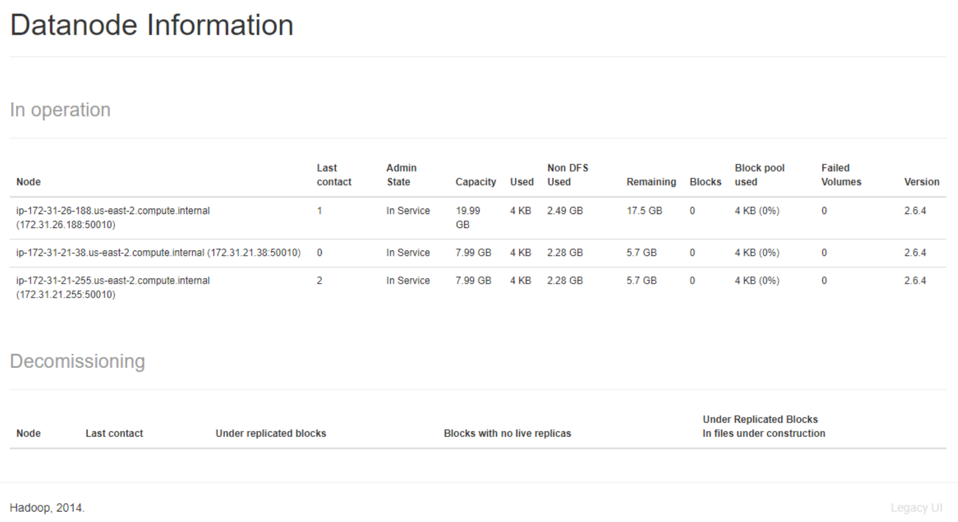
Robert Kaszubski

CSC 555 Project Phase 1

Midterm

Part 1)

I set up my cluster:



Datanode Information

In operation

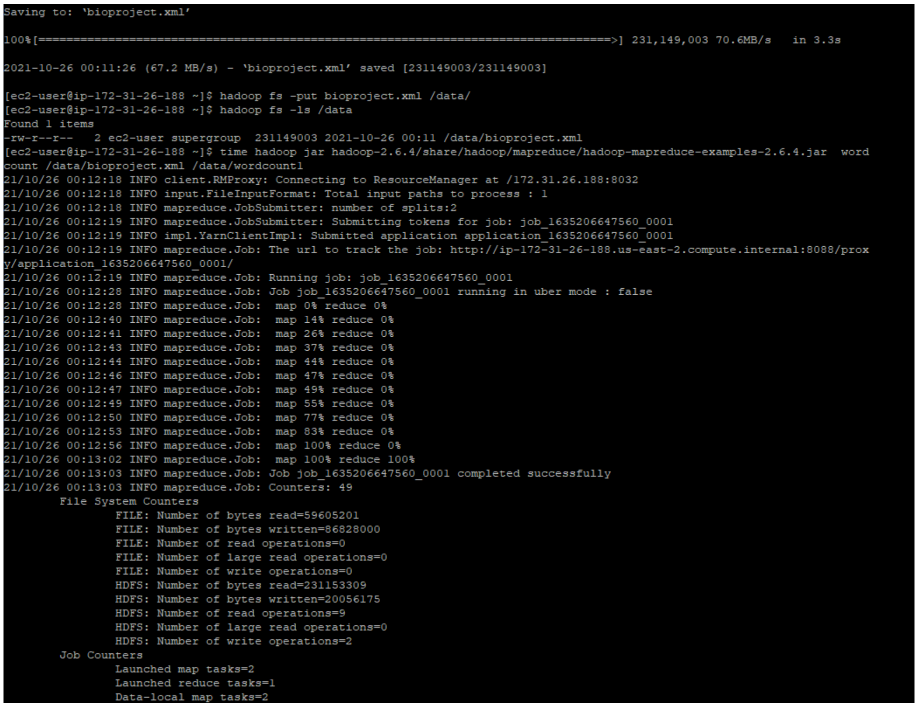
Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
ip-172-31-26-188.us-east-2.compute.internal (172.31.26.188:50010)	1	In Service	19.99 GB	4 KB	2.49 GB	17.5 GB	0	4 KB (0%)	0	2.6.4
ip-172-31-21-38.us-east-2.compute.internal (172.31.21.38:50010)	0	In Service	7.99 GB	4 KB	2.28 GB	5.7 GB	0	4 KB (0%)	0	2.6.4
ip-172-31-21-255.us-east-2.compute.internal (172.31.21.255:50010)	2	In Service	7.99 GB	4 KB	2.28 GB	5.7 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

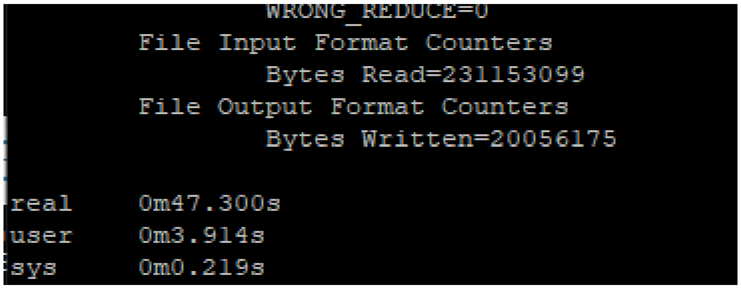
Hadoop, 2014.

Running wordcount:



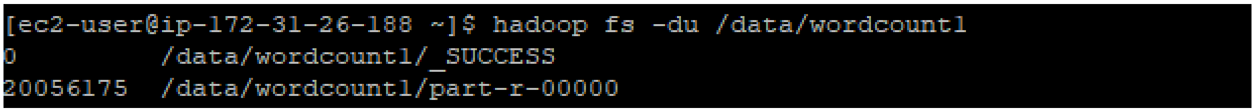
```
Saving to: 'bioproject.xml'

100%[===================================================================================>] 231,149,003 70.6MB/s   in 3.3s

2021-10-26 00:11:26 (67.2 MB/s) - 'bioproject.xml' saved [231149003/231149003]

[ec2-user@ip-172-31-26-188 ~]$ hadoop fs -put bioproject.xml /data/
[ec2-user@ip-172-31-26-188 ~]$ hadoop fs -ls /data
Found 1 items
-rw-r--r--   2 ec2-user supergroup  231149003 2021-10-26 00:11 /data/bioproject.xml
[ec2-user@ip-172-31-26-188 ~]$ time hadoop jar hadoop-2.6.4/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.4.jar  word
count /data/bioproject.xml /data/wordcount1
21/10/26 00:12:18 INFO client.RMProxy: Connecting to ResourceManager at /172.31.26.188:8032
21/10/26 00:12:18 INFO input.FileInputFormat: Total input paths to process : 1
21/10/26 00:12:18 INFO mapreduce.JobSubmitter: number of splits:2
21/10/26 00:12:19 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1635206647560_0001
21/10/26 00:12:19 INFO impl.YarnClientImpl: Submitted application application_1635206647560_0001
21/10/26 00:12:19 INFO mapreduce.Job: The url to track the job: http://ip-172-31-26-188.us-east-2.compute.internal:8088/prox
y/application_1635206647560_0001/
21/10/26 00:12:19 INFO mapreduce.Job: Running job: job_1635206647560_0001
21/10/26 00:12:28 INFO mapreduce.Job: Job job_1635206647560_0001 running in uber mode : false
21/10/26 00:12:28 INFO mapreduce.Job:  map 0% reduce 0%
21/10/26 00:12:40 INFO mapreduce.Job:  map 14% reduce 0%
21/10/26 00:12:41 INFO mapreduce.Job:  map 26% reduce 0%
21/10/26 00:12:43 INFO mapreduce.Job:  map 37% reduce 0%
21/10/26 00:12:44 INFO mapreduce.Job:  map 44% reduce 0%
21/10/26 00:12:46 INFO mapreduce.Job:  map 47% reduce 0%
21/10/26 00:12:47 INFO mapreduce.Job:  map 49% reduce 0%
21/10/26 00:12:49 INFO mapreduce.Job:  map 55% reduce 0%
21/10/26 00:12:50 INFO mapreduce.Job:  map 77% reduce 0%
21/10/26 00:12:53 INFO mapreduce.Job:  map 83% reduce 0%
21/10/26 00:12:56 INFO mapreduce.Job:  map 100% reduce 0%
21/10/26 00:13:02 INFO mapreduce.Job:  map 100% reduce 100%
21/10/26 00:13:03 INFO mapreduce.Job: Job job_1635206647560_0001 completed successfully
21/10/26 00:13:03 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=59605201
                FILE: Number of bytes written=86828000
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=231153309
                HDFS: Number of bytes written=20056175
                HDFS: Number of read operations=9
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
```

Time:



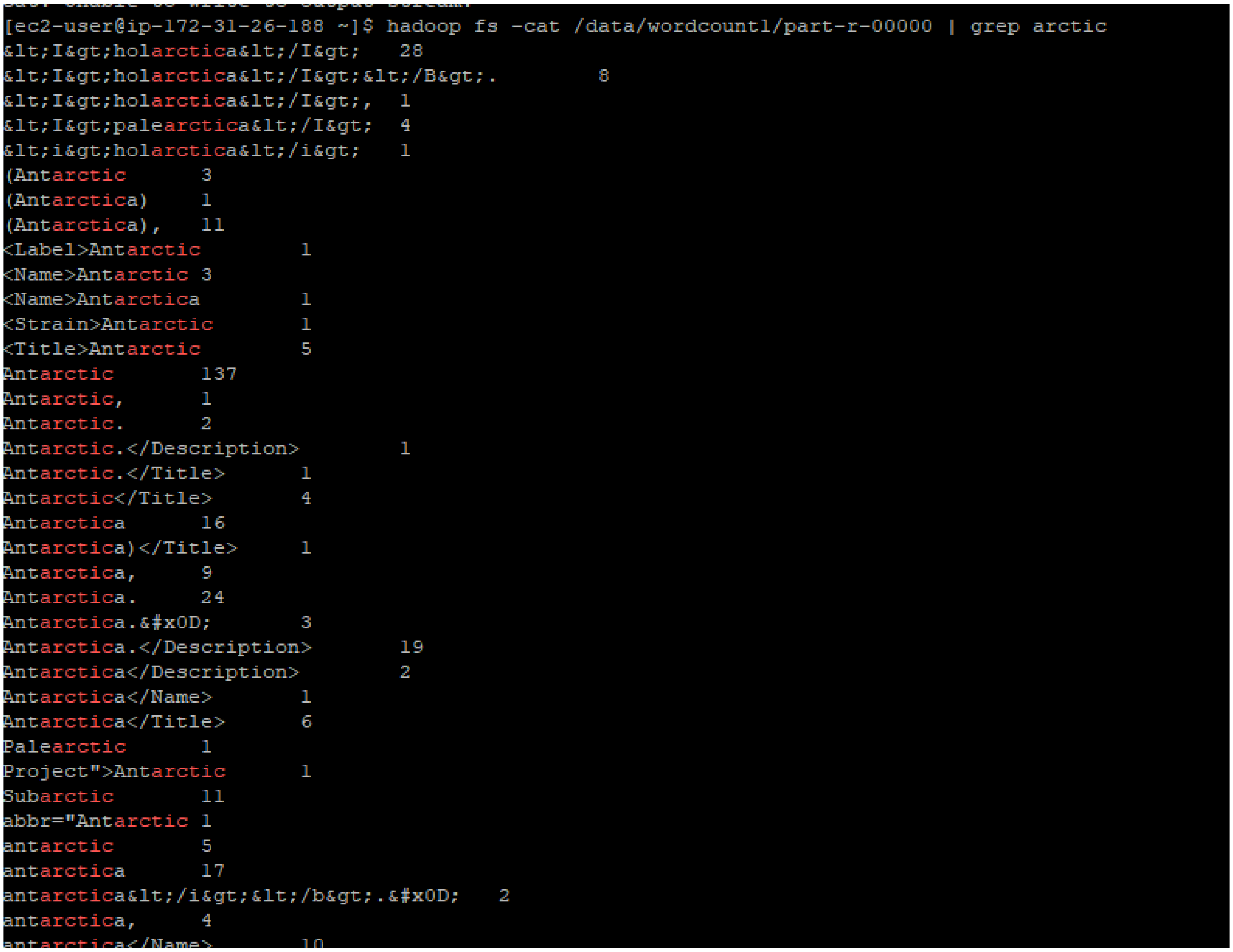
```
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=231153099
        File Output Format Counters
                Bytes Written=20056175

real    0m47.300s
user    0m3.914s
sys     0m0.219s
```

Successful output:



```
[ec2-user@ip-172-31-26-188 ~]$ hadoop fs -du /data/wordcount1
0          /data/wordcount1/_SUCCESS
20056175   /data/wordcount1/part-r-00000
```

Running Grep Arctic as in Assignment 2:



```
[ec2-user@ip-172-31-26-188 ~]$ hadoop fs -cat /data/wordcount1/part-r-00000 | grep arctic
&lt;I&gt;holarctica&lt;/I&gt;   28
&lt;I&gt;holarctica&lt;/I&gt;&lt;/B&gt;.        8
&lt;I&gt;holarctica&lt;/I&gt;,  1
&lt;I&gt;palearctica&lt;/I&gt;  4
&lt;i&gt;holarctica&lt;/i&gt;   1
(Antarctic      3
(Antarctica)    1
(Antarctica),   11
<Label>Antarctic        1
<Name>Antarctic 3
<Name>Antarctica        1
<Strain>Antarctic       1
<Title>Antarctic        5
Antarctic       137
Antarctic,      1
Antarctic.      2
Antarctic.</Description>        1
Antarctic.</Title>      1
Antarctic</Title>       4
Antarctica      16
Antarctica)</Title>     1
Antarctica,     9
Antarctica.     24
Antarctica.&#x0D;       3
Antarctica.</Description>       19
Antarctica</Description>        2
Antarctica</Name>       1
Antarctica</Title>      6
Palearctic      1
Project">Antarctic      1
Subarctic       11
abbr="Antarctic 1
antarctic       5
antarctica      17
antarctica&lt;/i&gt;&lt;/b&gt;.&#x0D;   2
antarctica,     4
antarctica</Name>       10
```

Looking back at assignment 2, running wordcount using the single node originally finished in 1 minute and 13.386 seconds. Running it now using the cluster setup, resulted in a run time of 47.3 seconds. So, it was about 26 seconds faster than before. I would have expected it to be even faster given that it was running on a three-node cluster rather than a single node, yet it wasn’t even twice as fast. However, we do have to consider that there were more things that may have slowed down the process a bit such as the way the blocks were spread out of distributed among the nodes. It is also likely that the speed of the network and connecting to different nodes played a part. However, the overall speed is still much faster than it was before.

Part 2

1)

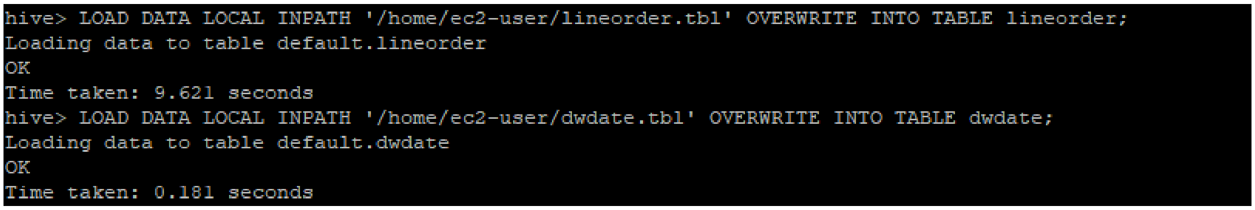
Building Tables:

```
create table 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_daynuminweek     int,

  d_daynuminmonth    int,

  d_daynuminyear     int,

  d_monthnuminyear   int,

  d_weeknuminyear    int,

  d_sellingseason    varchar(13),

  d_lastdayinweekfl  varchar(1),

  d_lastdayinmonthfl varchar(1),

  d_holidayfl        varchar(1),

  d_weekdayfl        varchar(1)

) ROW FORMAT DELIMITED FIELDS

TERMINATED BY '|' STORED AS TEXTFILE;
```

```
create table lineorder (
  lo_orderkey          int,
  lo_linenumber        int,
  lo_custkey           int,
  lo_partkey           int,
  lo_suppkey           int,
  lo_orderdate         int,
  lo_orderpriority     varchar(15),
  lo_shippriority      varchar(1),
  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          varchar(10)
) ROW FORMAT DELIMITED FIELDS
TERMINATED BY '|' STORED AS TEXTFILE;
```

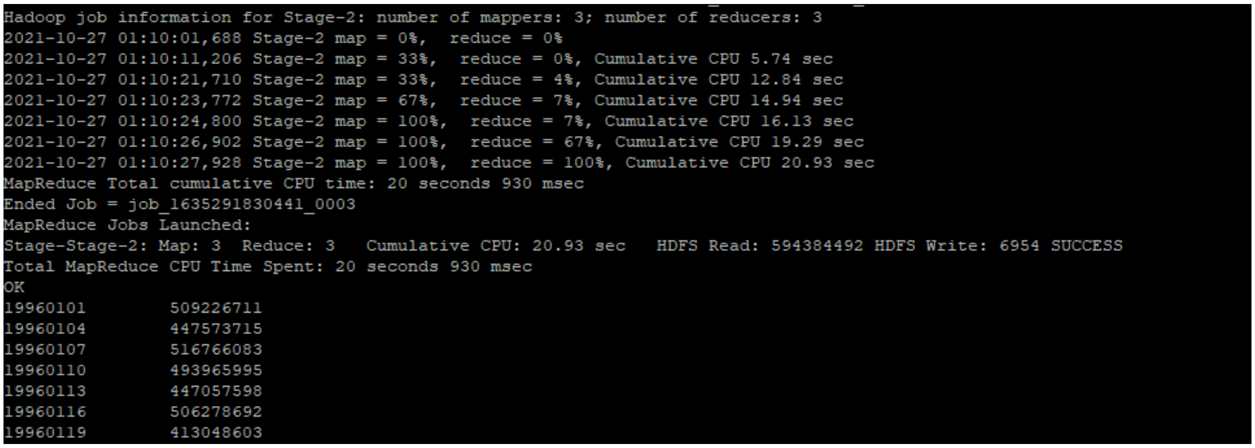
Import data:

```
LOAD DATA LOCAL INPATH '/home/ec2-user/dwdate.tbl' OVERWRITE INTO TABLE dwdate;
LOAD DATA LOCAL INPATH '/home/ec2-user/lineorder.tbl' OVERWRITE INTO TABLE lineorder;
```

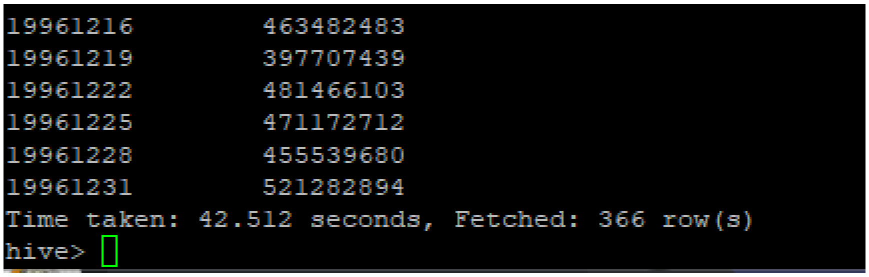


```
hive> LOAD DATA LOCAL INPATH '/home/ec2-user/lineorder.tbl' OVERWRITE INTO TABLE lineorder;
Loading data to table default.lineorder
OK
Time taken: 9.621 seconds
hive> LOAD DATA LOCAL INPATH '/home/ec2-user/dwdate.tbl' OVERWRITE INTO TABLE dwdate;
Loading data to table default.dwdate
OK
Time taken: 0.181 seconds
```

Running Query:



```
Hadoop job information for Stage-2: number of mappers: 3; number of reducers: 3
2021-10-27 01:10:01,688 Stage-2 map = 0%,  reduce = 0%
2021-10-27 01:10:11,206 Stage-2 map = 33%,  reduce = 0%, Cumulative CPU 5.74 sec
2021-10-27 01:10:21,710 Stage-2 map = 33%,  reduce = 4%, Cumulative CPU 12.84 sec
2021-10-27 01:10:23,772 Stage-2 map = 67%,  reduce = 7%, Cumulative CPU 14.94 sec
2021-10-27 01:10:24,800 Stage-2 map = 100%,  reduce = 7%, Cumulative CPU 16.13 sec
2021-10-27 01:10:26,902 Stage-2 map = 100%,  reduce = 67%, Cumulative CPU 19.29 sec
2021-10-27 01:10:27,928 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 20.93 sec
MapReduce Total cumulative CPU time: 20 seconds 930 msec
Ended Job = job_1635291830441_0003
MapReduce Jobs Launched:
Stage-Stage-2: Map: 3  Reduce: 3   Cumulative CPU: 20.93 sec   HDFS Read: 594384492 HDFS Write: 6954 SUCCESS
Total MapReduce CPU Time Spent: 20 seconds 930 msec
OK
19960101        509226711
19960104        447573715
19960107        516766083
19960110        493965995
19960113        447057598
19960116        506278692
19960119        413048603
```

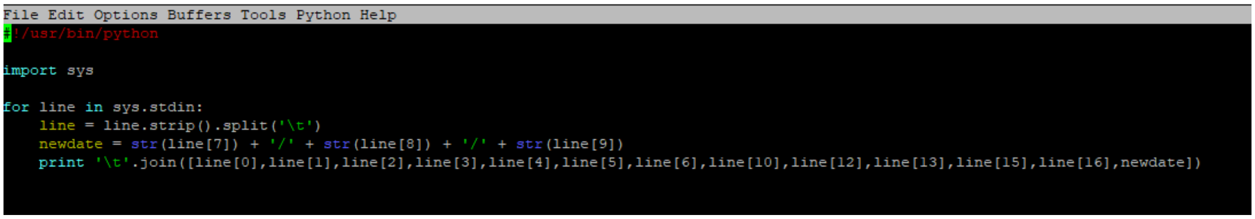


```
19961216        463482483
19961219        397707439
19961222        481466103
19961225        471172712
19961228        455539680
19961231        521282894
Time taken: 42.512 seconds, Fetched: 366 row(s)
hive> 
```

The query took 42.512 seconds to run

2)

Python Code:



```
#!/usr/bin/python

import sys

for line in sys.stdin:
    line = line.strip().split('\t')
    newdate = str(line[7]) + '/' + str(line[8]) + '/' + str(line[9])
    print '\t'.join([line[0],line[1],line[2],line[3],line[4],line[5],line[6],line[10],line[12],line[13],line[15],line[16],newdate])
```

ADD FILE /home/ec2-user/part2b.py

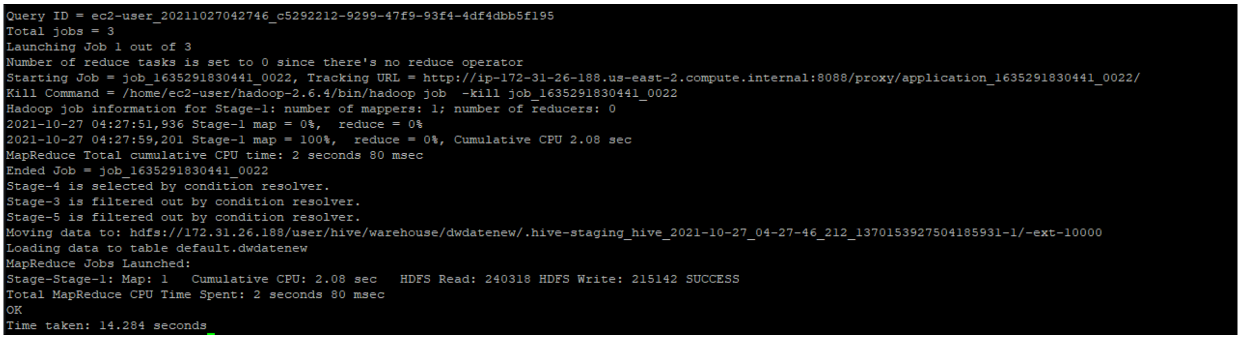
New Table Schema:

create table dwdatenew (

```
 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_sellingseason     varchar(13),
 d_lastdayinweekfl   varchar(1),
 d_holidayfl         varchar(1),
 d_weekdayfl         varchar(1),
 d_daynuminweekmonthyear          varchar(10)
) ROW FORMAT DELIMITED FIELDS
TERMINATED BY '\t' STORED AS TEXTFILE;
```

COMMAND:

```
INSERT OVERWRITE TABLE dwdatenew 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 part2b.py' AS (d_datekey, d_date,
d_dayofweek, d_month, d_year, d_yearmonthnum, d_yearmonth, d_monthnuminyear, d_sellingseason,
d_lastdayinweekfl, d_holidayfl, d_weekdayfl, d_daynuminweekmonthyear) FROM dwdate;
```

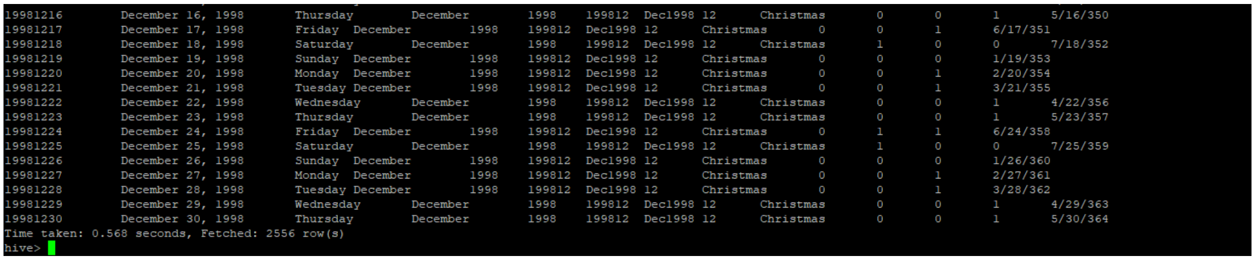


```
Query ID = ec2-user_20211027042746_c5292212-9299-47f9-93f4-4df4dbb5f195
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1635291830441_0022, Tracking URL = http://ip-172-31-26-188.us-east-2.compute.internal:8088/proxy/application_1635291830441_0022/
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job  -kill job_1635291830441_0022
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2021-10-27 04:27:51,936 Stage-1 map = 0%,  reduce = 0%
2021-10-27 04:27:59,201 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.08 sec
MapReduce Total cumulative CPU time: 2 seconds 80 msec
Ended Job = job_1635291830441_0022
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://172.31.26.188/user/hive/warehouse/dwdatenew/.hive-staging_hive_2021-10-27_04-27-46_212_1370153927504185931-1/-ext-10000
Loading data to table default.dwdatenew
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 2.08 sec   HDFS Read: 240318 HDFS Write: 215142 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 80 msec
OK
Time taken: 14.284 seconds
```

Took 14.284 seconds to run

Sample results: new column is added to the end, other columns removed – 13 total columns now vs 17 before

New column is called d_daynuminweekmonthyear and takes the format of day num in week/month/year.



```
19981216        December 16, 1998       Thursday        December        1998    199812  Dec1998 12      Christmas       0       0       1       5/16/350
19981217        December 17, 1998       Friday  December        1998    199812  Dec1998 12      Christmas       0       0       1       6/17/351
19981218        December 18, 1998       Saturday        December        1998    199812  Dec1998 12      Christmas       1       0       0       7/18/352
19981219        December 19, 1998       Sunday  December        1998    199812  Dec1998 12      Christmas       0       0       0       1/19/353
19981220        December 20, 1998       Monday  December        1998    199812  Dec1998 12      Christmas       0       0       1       2/20/354
19981221        December 21, 1998       Tuesday December        1998    199812  Dec1998 12      Christmas       0       0       1       3/21/355
19981222        December 22, 1998       Wednesday       December        1998    199812  Dec1998 12      Christmas       0       0       1       4/22/356
19981223        December 23, 1998       Thursday        December        1998    199812  Dec1998 12      Christmas       0       0       1       5/23/357
19981224        December 24, 1998       Friday  December        1998    199812  Dec1998 12      Christmas       0       1       1       6/24/358
19981225        December 25, 1998       Saturday        December        1998    199812  Dec1998 12      Christmas       1       0       0       7/25/359
19981226        December 26, 1998       Sunday  December        1998    199812  Dec1998 12      Christmas       0       0       0       1/26/360
19981227        December 27, 1998       Monday  December        1998    199812  Dec1998 12      Christmas       0       0       1       2/27/361
19981228        December 28, 1998       Tuesday December        1998    199812  Dec1998 12      Christmas       0       0       1       3/28/362
19981229        December 29, 1998       Wednesday       December        1998    199812  Dec1998 12      Christmas       0       0       1       4/29/363
19981230        December 30, 1998       Thursday        December        1998    199812  Dec1998 12      Christmas       0       0       1       5/30/364
Time taken: 0.568 seconds, Fetched: 2556 row(s)
hive> 
```

Part 3

lineorder = 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);

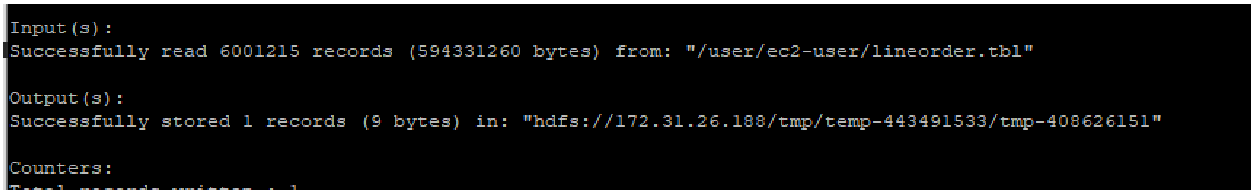
Testing:

lineorderG = GROUP lineorder ALL;

Count = FOREACH lineorderG GENERATE COUNT(lineorder);

DUMP Count;

Data was loaded and pig works!



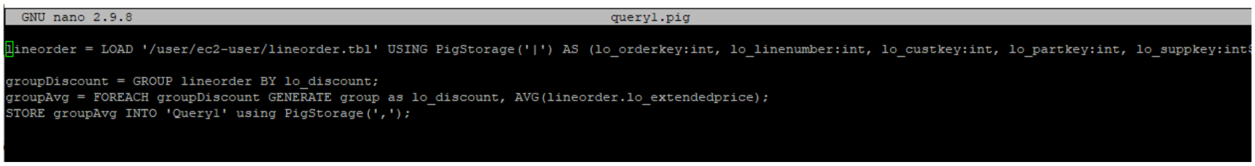
```
Input(s):
Successfully read 6001215 records (594331260 bytes) from: "/user/ec2-user/lineorder.tbl"

Output(s):
Successfully stored 1 records (9 bytes) in: "hdfs://172.31.26.188/tmp/temp-443491533/tmp-408626151"

Counters:
Total records written : 1
```

Query 1:

groupDiscount = GROUP lineorder BY lo_discount;

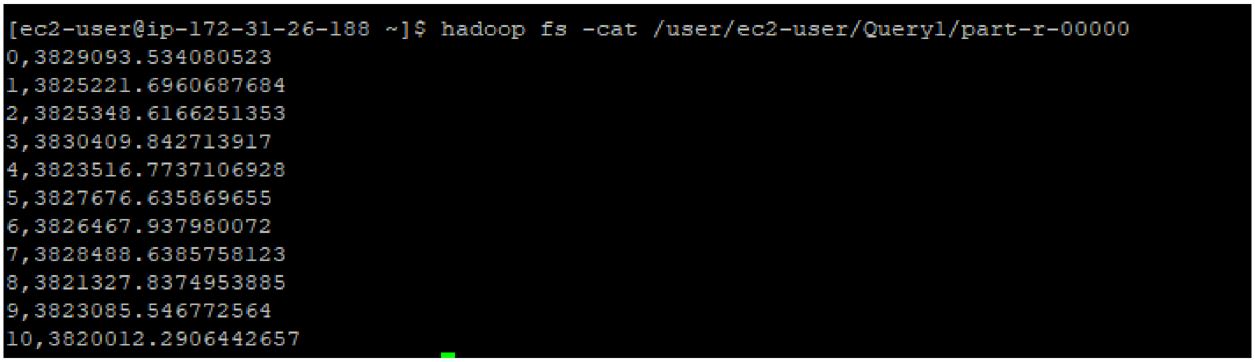
groupAvg = FOREACH groupDiscount GENERATE group as lo_discount, AVG(lineorder.lo_extendedprice);

STORE groupAvg INTO 'Query1' using PigStorage(', ');



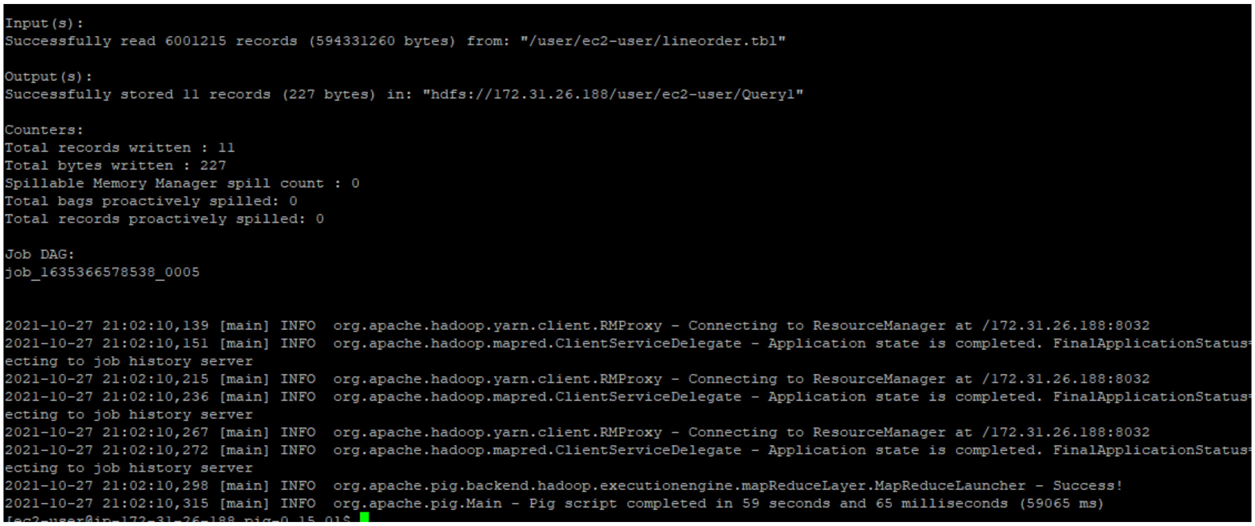
```
  GNU nano 2.9.8                                                       query1.pig

lineorder = LOAD '/user/ec2-user/lineorder.tbl' USING PigStorage('|') AS (lo_orderkey:int, lo_linenumber:int, lo_custkey:int, lo_partkey:int, lo_suppkey:int$

groupDiscount = GROUP lineorder BY lo_discount;
groupAvg = FOREACH groupDiscount GENERATE group as lo_discount, AVG(lineorder.lo_extendedprice);
STORE groupAvg INTO 'Query1' using PigStorage(',');
```

Output:



```
[ec2-user@ip-172-31-26-188 ~]$ hadoop fs -cat /user/ec2-user/Query1/part-r-00000
0,3829093.534080523
1,3825221.6960687684
2,3825348.6166251353
3,3830409.842713917
4,3823516.7737106928
5,3827676.635869655
6,3826467.937980072
7,3828488.6385758123
8,3821327.8374953885
9,3823085.546772564
10,3820012.2906442657
```

Running as Script:



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

Output(s):
Successfully stored 11 records (227 bytes) in: "hdfs://172.31.26.188/user/ec2-user/Query1"

Counters:
Total records written : 11
Total bytes written : 227
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1635366578538_0005


2021-10-27 21:02:10,139 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /172.31.26.188:8032
2021-10-27 21:02:10,151 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=
ecting to job history server
2021-10-27 21:02:10,215 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /172.31.26.188:8032
2021-10-27 21:02:10,236 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=
ecting to job history server
2021-10-27 21:02:10,267 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /172.31.26.188:8032
2021-10-27 21:02:10,272 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=
ecting to job history server
2021-10-27 21:02:10,298 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2021-10-27 21:02:10,315 [main] INFO  org.apache.pig.Main - Pig script completed in 59 seconds and 65 milliseconds (59065 ms)
[ec2-user@ip-172-31-26-188 pig-0.15.0]$
```

Runtime for **Query 1 is 59 seconds and 65 milliseconds**

Query 2:

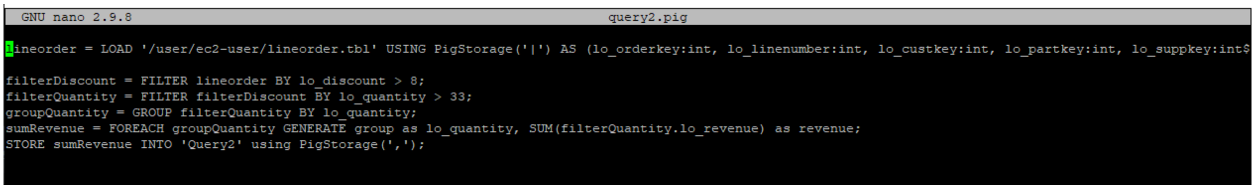
filterDiscount = FILTER lineorder BY lo_discount > 8;

filterQuantity = FILTER filterDiscount BY lo_quantity > 33;

groupQuantity = GROUP filterQuantity BY lo_quantity;

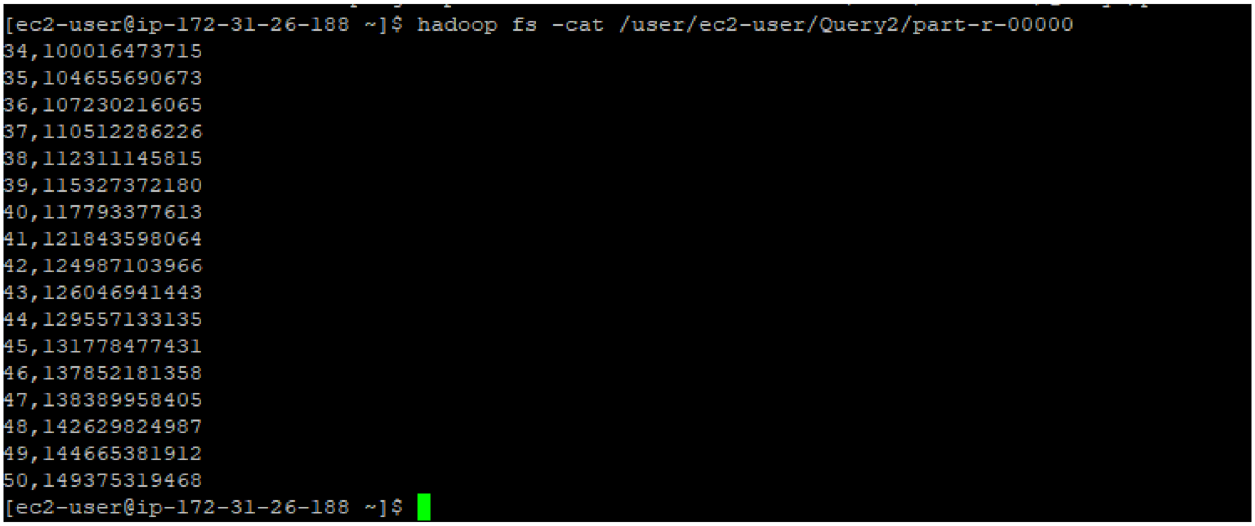
sumRevenue = FOREACH groupQuantity GENERATE group as lo_quantity,
SUM(filterQuantity.lo_revenue) as revenue;

STORE sumRevenue INTO 'Query2' using PigStorage(',');



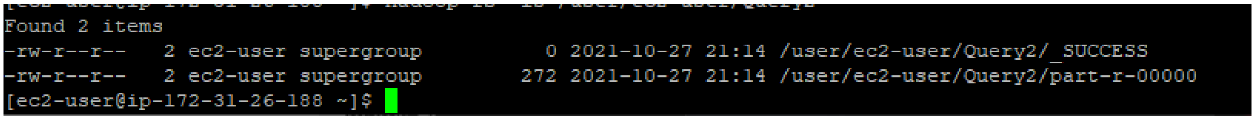
```
  GNU nano 2.9.8                                   query2.pig

lineorder = LOAD '/user/ec2-user/lineorder.tbl' USING PigStorage('|') AS (lo_orderkey:int, lo_linenumber:int, lo_custkey:int, lo_partkey:int, lo_suppkey:int$

filterDiscount = FILTER lineorder BY lo_discount > 8;
filterQuantity = FILTER filterDiscount BY lo_quantity > 33;
groupQuantity = GROUP filterQuantity BY lo_quantity;
sumRevenue = FOREACH groupQuantity GENERATE group as lo_quantity, SUM(filterQuantity.lo_revenue) as revenue;
STORE sumRevenue INTO 'Query2' using PigStorage(',');
```

Output:



```
[ec2-user@ip-172-31-26-188 ~]$ hadoop fs -cat /user/ec2-user/Query2/part-r-00000
34,100016473715
35,104655690673
36,107230216065
37,110512286226
38,112311145815
39,115327372180
40,117793377613
41,121843598064
42,124987103966
43,126046941443
44,129557133135
45,131778477431
46,137852181358
47,138389958405
48,142629824987
49,144665381912
50,149375319468
[ec2-user@ip-172-31-26-188 ~]$
```

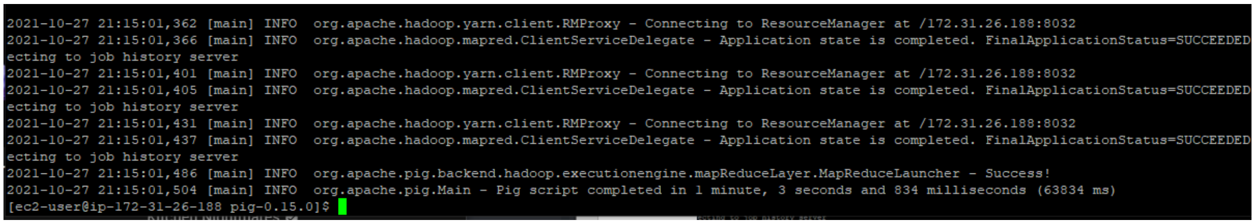
File Size:



```
Found 2 items
-rw-r--r--   2 ec2-user supergroup          0 2021-10-27 21:14 /user/ec2-user/Query2/_SUCCESS
-rw-r--r--   2 ec2-user supergroup        272 2021-10-27 21:14 /user/ec2-user/Query2/part-r-00000
[ec2-user@ip-172-31-26-188 ~]$
```

File size is **272 bytes**

Running as script:



```
2021-10-27 21:15:01,362 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /172.31.26.188:8032
2021-10-27 21:15:01,366 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED
ecting to job history server
2021-10-27 21:15:01,401 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /172.31.26.188:8032
2021-10-27 21:15:01,405 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED
ecting to job history server
2021-10-27 21:15:01,431 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /172.31.26.188:8032
2021-10-27 21:15:01,437 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED
ecting to job history server
2021-10-27 21:15:01,486 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2021-10-27 21:15:01,504 [main] INFO  org.apache.pig.Main - Pig script completed in 1 minute, 3 seconds and 834 milliseconds (63834 ms)
[ec2-user@ip-172-31-26-188 pig-0.15.0]$
```

Query 2 took **1 minute, 3 seconds, and 834 milliseconds** to run

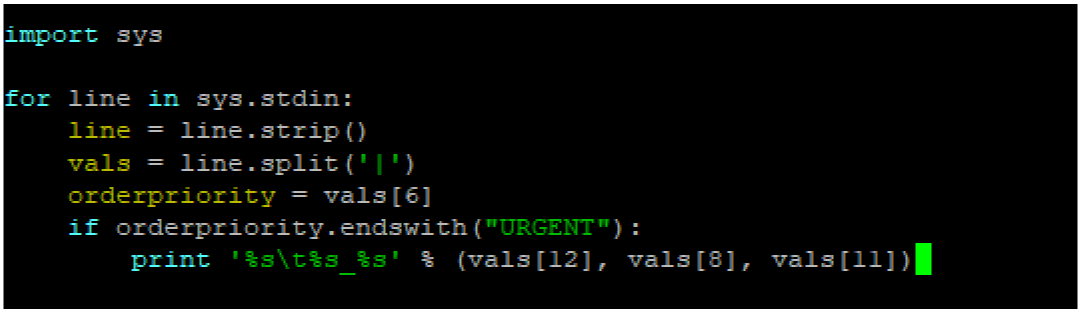
Part 4

SUBQUERY:

We are first trying to implement the subquery:

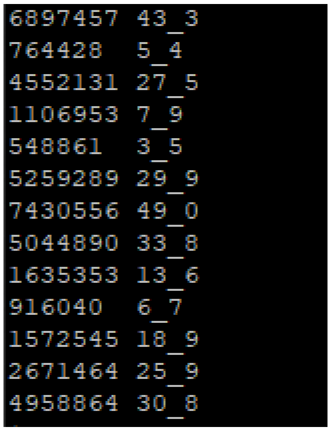
```
SELECT lo_revenue, MAX(lo_quantity) as lo_quantity,
MAX(lo_discount) as lo_discount
FROM lineorder
WHERE lo_orderpriority LIKE '%URGENT'
GROUP BY lo_revenue)
```

MyMapper1.py:



```
import sys

for line in sys.stdin:
    line = line.strip()
    vals = line.split('|')
    orderpriority = vals[6]
    if orderpriority.endswith("URGENT"):
        print '%s\t%s_%s' % (vals[12], vals[8], vals[11])
```

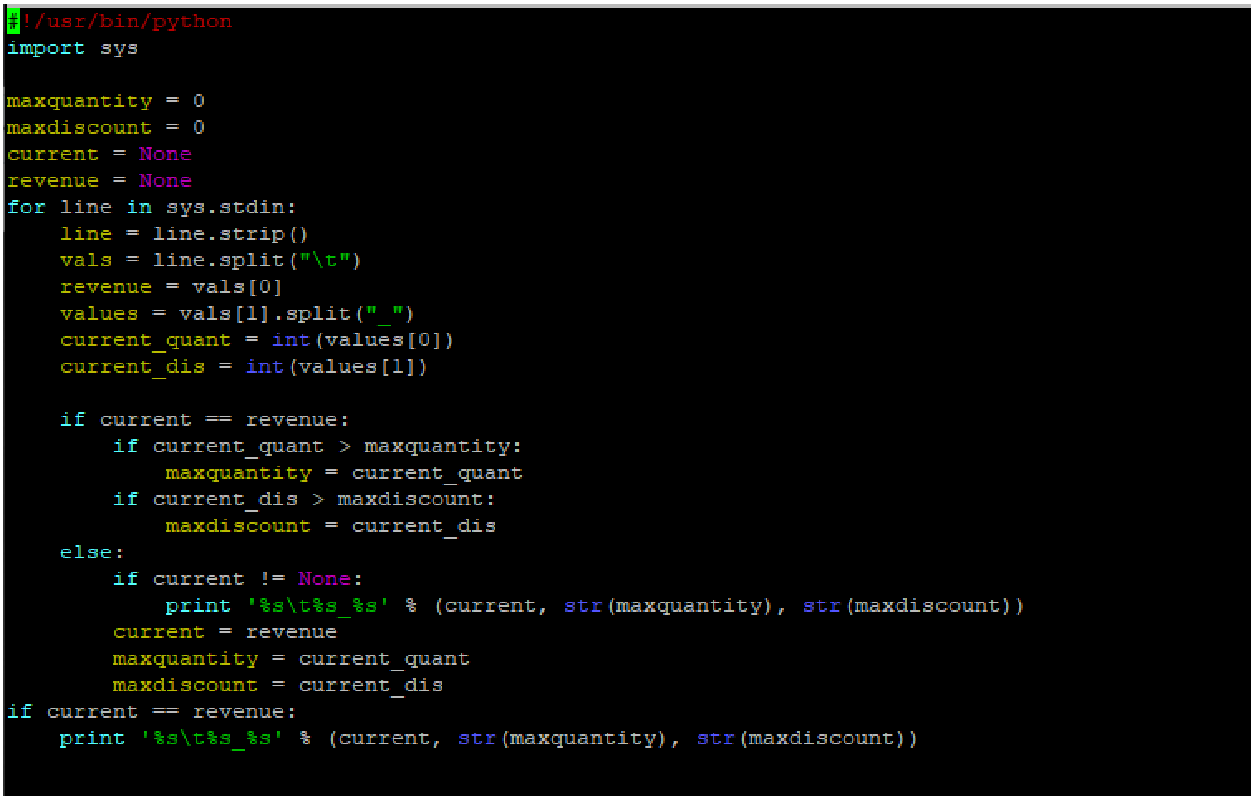
Sample Output of MyMapper1 (cat lineorder.tbl | python MyMapper1)



```
6897457 43_3
764428  5_4
4552131 27_5
1106953 7_9
548861  3_5
5259289 29_9
7430556 49_0
5044890 33_8
1635353 13_6
916040  6_7
1572545 18_9
2671464 25_9
4958864 30_8
```

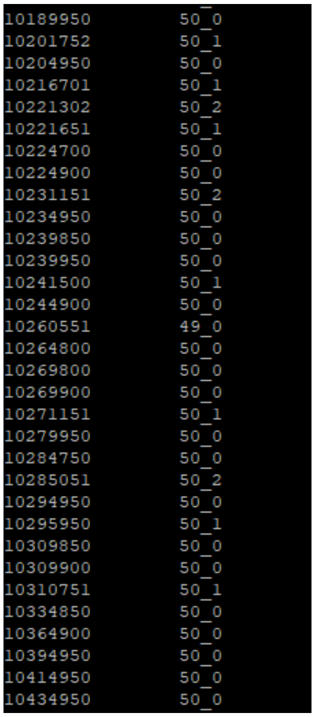
Key is lo_revenue, Value is lo_quantity _ lo_discount

MyReducer1.py:



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

maxquantity = 0
maxdiscount = 0
current = None
revenue = None
for line in sys.stdin:
    line = line.strip()
    vals = line.split("\t")
    revenue = vals[0]
    values = vals[1].split("_")
    current_quant = int(values[0])
    current_dis = int(values[1])

    if current == revenue:
        if current_quant > maxquantity:
            maxquantity = current_quant
        if current_dis > maxdiscount:
            maxdiscount = current_dis
    else:
        if current != None:
            print '%s\t%s_%s' % (current, str(maxquantity), str(maxdiscount))
        current = revenue
        maxquantity = current_quant
        maxdiscount = current_dis
if current == revenue:
    print '%s\t%s_%s' % (current, str(maxquantity), str(maxdiscount))
```

Sample Output of MyReducer1 (cat lineorder.tbl | python MyMapper1.py | sort -n | python MyReducer1.py):



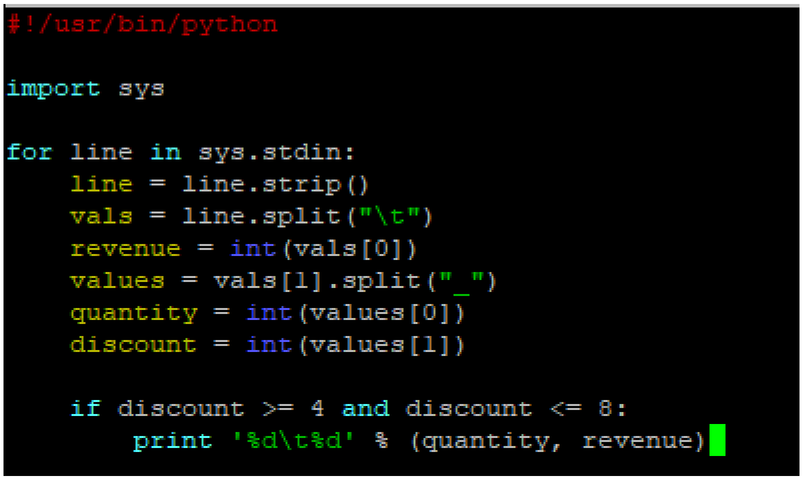
```
10189950        50_0
10201752        50_1
10204950        50_0
10216701        50_1
10221302        50_2
10221651        50_1
10224700        50_0
10224900        50_0
10231151        50_2
10234950        50_0
10239850        50_0
10239950        50_0
10241500        50_1
10244900        50_0
10260551        49_0
10264800        50_0
10269800        50_0
10269900        50_0
10271151        50_1
10279950        50_0
10284750        50_0
10285051        50_2
10294950        50_0
10295950        50_1
10309850        50_0
10309900        50_0
10310751        50_1
10334850        50_0
10364900        50_0
10394950        50_0
10414950        50_0
10434950        50_0
```

Key is lo_revenue, Value is max(lo_quantity)_max(lo_discount)

Now with the subquery complete we can take this data and begin our second MapReduce operation:

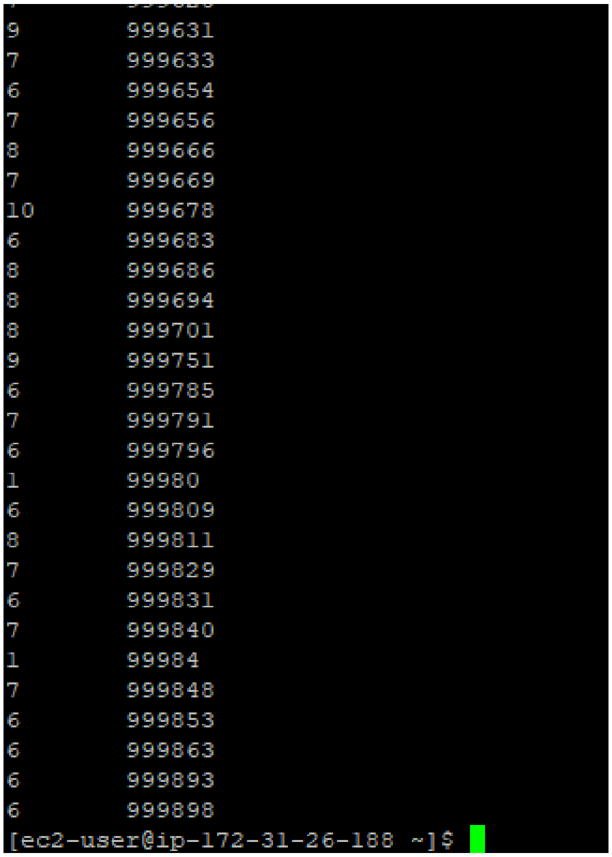
Main Query:

MyMapper2.py:



```
#!/usr/bin/python

import sys

for line in sys.stdin:
    line = line.strip()
    vals = line.split("\t")
    revenue = int(vals[0])
    values = vals[1].split("_")
    quantity = int(values[0])
    discount = int(values[1])

    if discount >= 4 and discount <= 8:
        print '%d\t%d' % (quantity, revenue)
```

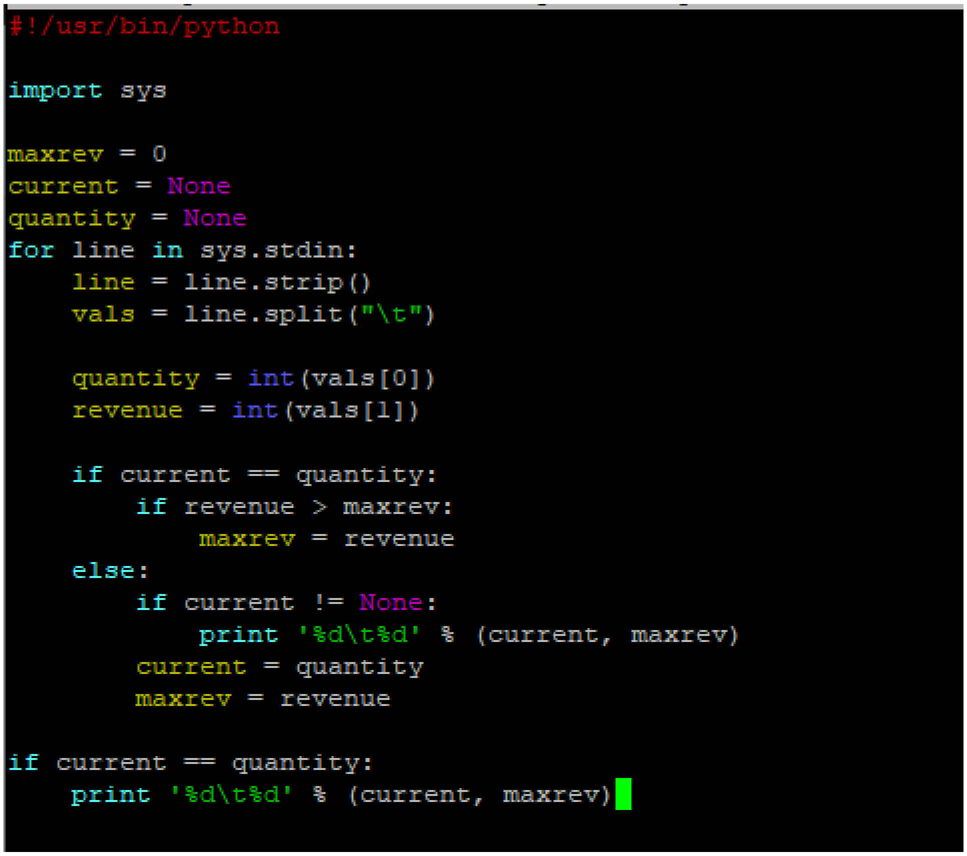
Sample Output of MyMapper2 (cat lineorder.tbl | python MyMapper1.py | sort -n | python MyReducer1.py | python MyMapper2.py) :



```
9       999631
7       999633
6       999654
7       999656
8       999666
7       999669
10      999678
6       999683
8       999686
8       999694
8       999701
9       999751
6       999785
7       999791
6       999796
1       99980
6       999809
8       999811
7       999829
6       999831
7       999840
1       99984
7       999848
6       999853
6       999863
6       999893
6       999898
[ec2-user@ip-172-31-26-188 ~]$
```

Key is now Quantity, Value is Revenue

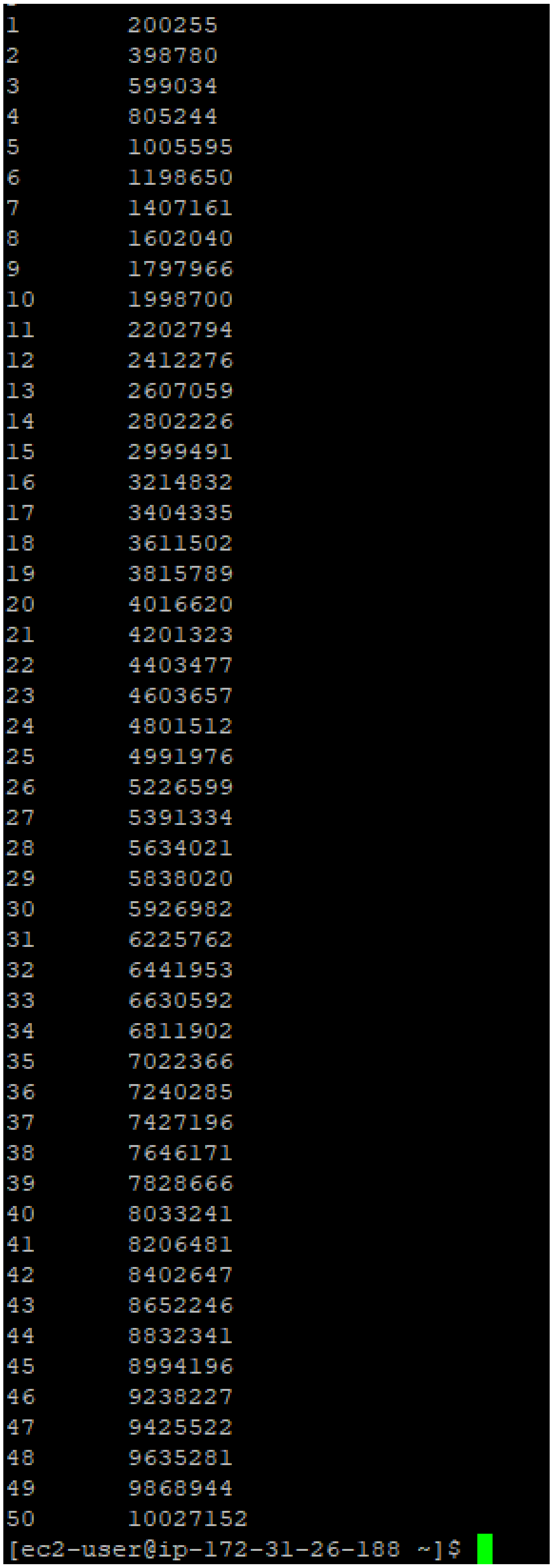
MyReducer2.py:



```
#!/usr/bin/python

import sys

maxrev = 0
current = None
quantity = None
for line in sys.stdin:
    line = line.strip()
    vals = line.split("\t")

    quantity = int(vals[0])
    revenue = int(vals[1])

    if current == quantity:
        if revenue > maxrev:
            maxrev = revenue
    else:
        if current != None:
            print '%d\t%d' % (current, maxrev)
        current = quantity
        maxrev = revenue

if current == quantity:
    print '%d\t%d' % (current, maxrev)
```

Output: cat lineorder.tbl | python MyMapper1.py | sort -n | python MyReducer1.py | python MyMapper2.py | sort -n | python MyReducer2.py

FINAL OUTPUT:



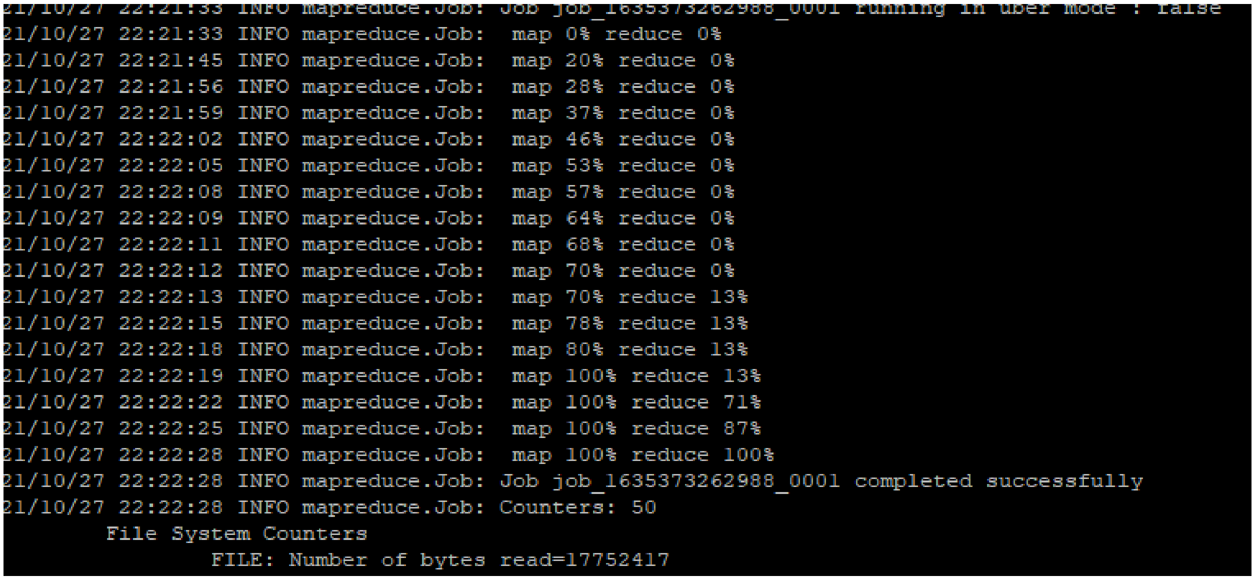
```
1       200255
2       398780
3       599034
4       805244
5       1005595
6       1198650
7       1407161
8       1602040
9       1797966
10      1998700
11      2202794
12      2412276
13      2607059
14      2802226
15      2999491
16      3214832
17      3404335
18      3611502
19      3815789
20      4016620
21      4201323
22      4403477
23      4603657
24      4801512
25      4991976
26      5226599
27      5391334
28      5634021
29      5838020
30      5926982
31      6225762
32      6441953
33      6630592
34      6811902
35      7022366
36      7240285
37      7427196
38      7646171
39      7828666
40      8033241
41      8206481
42      8402647
43      8652246
44      8832341
45      8994196
46      9238227
47      9425522
48      9635281
49      9868944
50      10027152
[ec2-user@ip-172-31-26-188 ~]$
```

Key is Quantity, Value is revenue

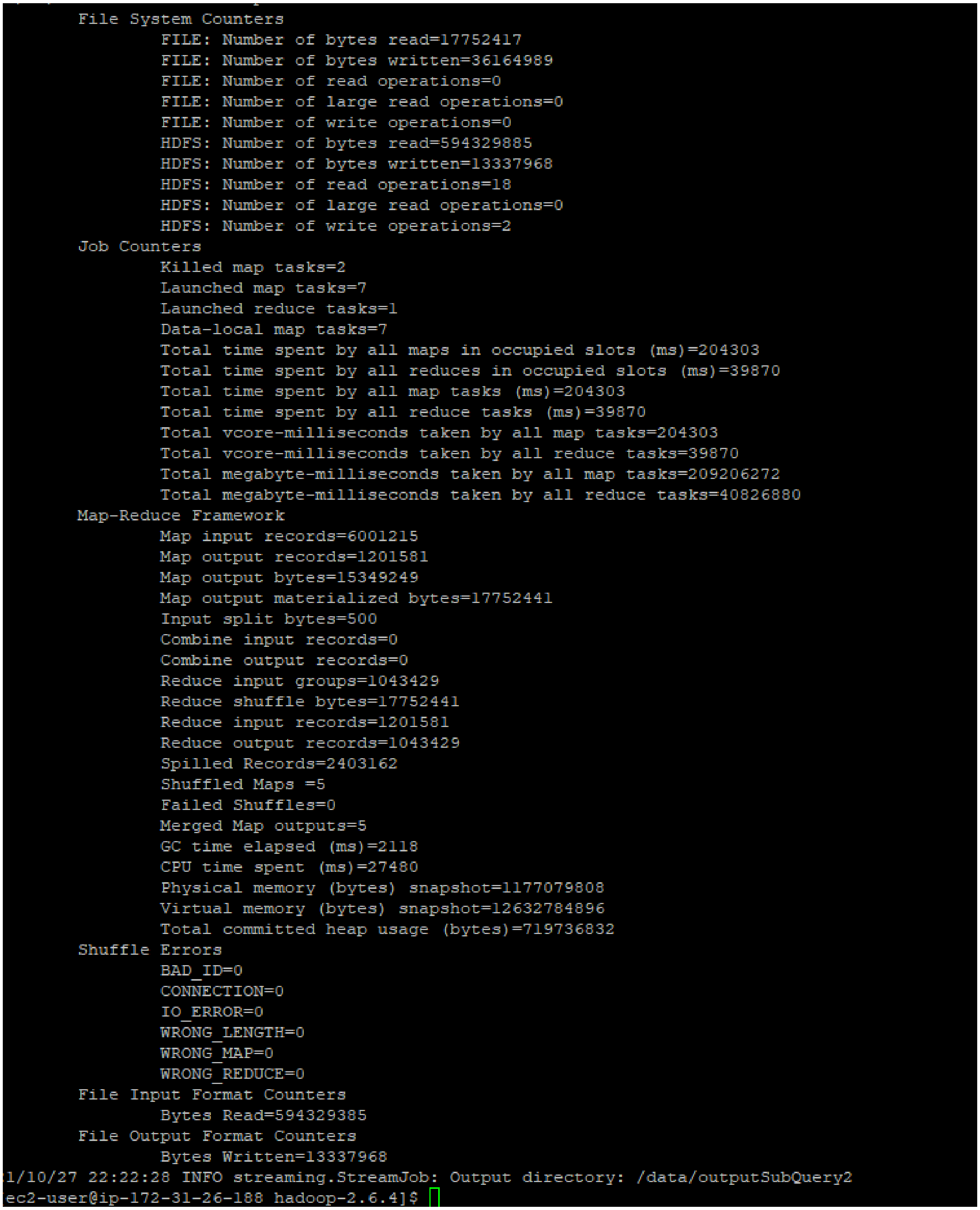
Using Hadoop Streaming:

FIRST MAP REDUCE JOB (ran in cd $HADOOP_HOME)

hadoop jar hadoop-streaming-2.6.4.jar -input /user/ec2-user/lineorder.tbl -output /data/outputSubQuery2 -mapper MyMapper1.py -reducer MyReducer1.py -file ../MyReducer1.py -file ../MyMapper1.py



```
21/10/27 22:21:33 INFO mapreduce.Job: Job job_1635373262988_0001 running in uber mode : false
21/10/27 22:21:33 INFO mapreduce.Job:  map 0% reduce 0%
21/10/27 22:21:45 INFO mapreduce.Job:  map 20% reduce 0%
21/10/27 22:21:56 INFO mapreduce.Job:  map 28% reduce 0%
21/10/27 22:21:59 INFO mapreduce.Job:  map 37% reduce 0%
21/10/27 22:22:02 INFO mapreduce.Job:  map 46% reduce 0%
21/10/27 22:22:05 INFO mapreduce.Job:  map 53% reduce 0%
21/10/27 22:22:08 INFO mapreduce.Job:  map 57% reduce 0%
21/10/27 22:22:09 INFO mapreduce.Job:  map 64% reduce 0%
21/10/27 22:22:11 INFO mapreduce.Job:  map 68% reduce 0%
21/10/27 22:22:12 INFO mapreduce.Job:  map 70% reduce 0%
21/10/27 22:22:13 INFO mapreduce.Job:  map 70% reduce 13%
21/10/27 22:22:15 INFO mapreduce.Job:  map 78% reduce 13%
21/10/27 22:22:18 INFO mapreduce.Job:  map 80% reduce 13%
21/10/27 22:22:19 INFO mapreduce.Job:  map 100% reduce 13%
21/10/27 22:22:22 INFO mapreduce.Job:  map 100% reduce 71%
21/10/27 22:22:25 INFO mapreduce.Job:  map 100% reduce 87%
21/10/27 22:22:28 INFO mapreduce.Job:  map 100% reduce 100%
21/10/27 22:22:28 INFO mapreduce.Job: Job job_1635373262988_0001 completed successfully
21/10/27 22:22:28 INFO mapreduce.Job: Counters: 50
        File System Counters
                FILE: Number of bytes read=17752417
```



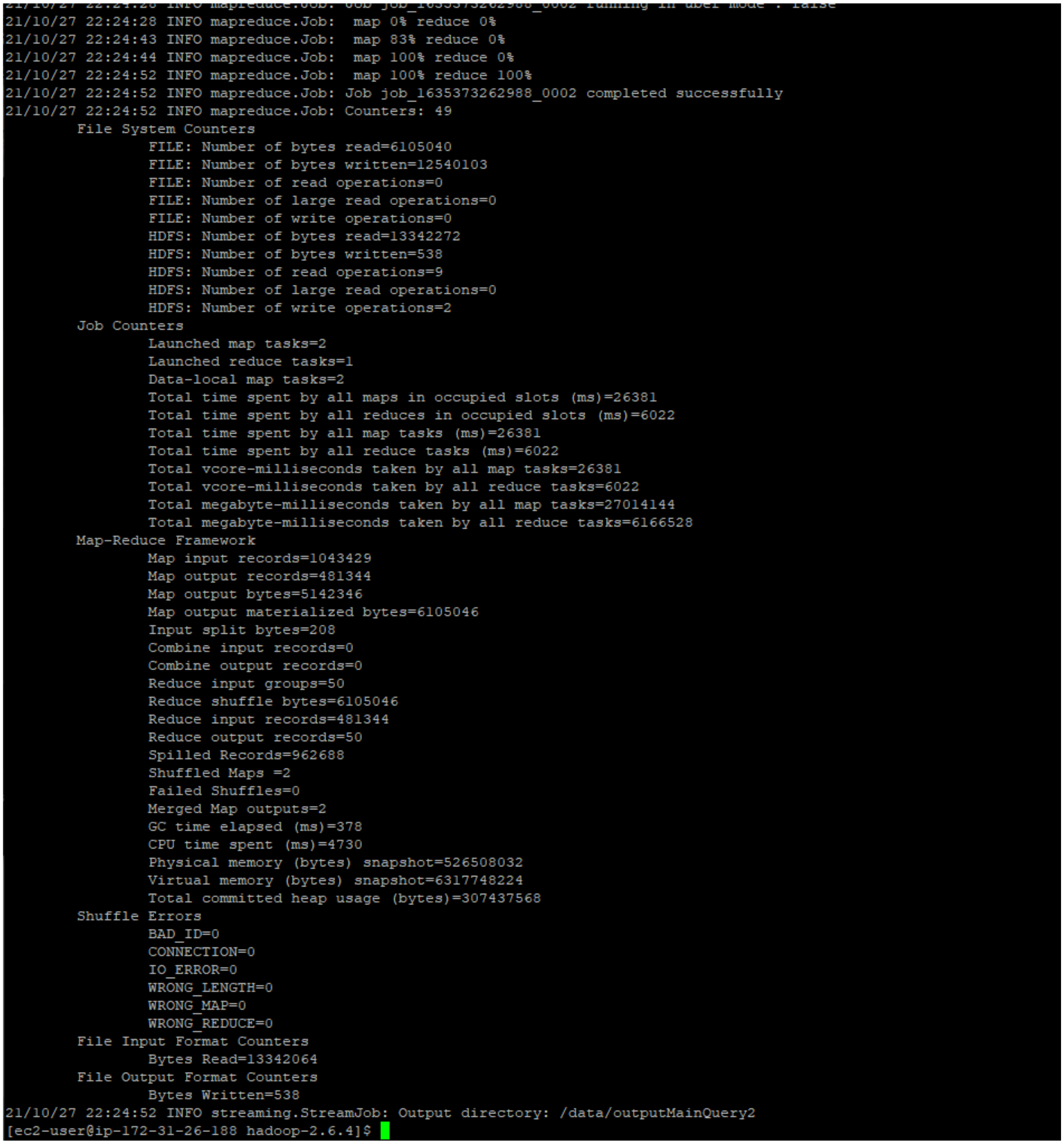
```
        File System Counters
                FILE: Number of bytes read=17752417
                FILE: Number of bytes written=36164989
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=594329885
                HDFS: Number of bytes written=13337968
                HDFS: Number of read operations=18
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Killed map tasks=2
                Launched map tasks=7
                Launched reduce tasks=1
                Data-local map tasks=7
                Total time spent by all maps in occupied slots (ms)=204303
                Total time spent by all reduces in occupied slots (ms)=39870
                Total time spent by all map tasks (ms)=204303
                Total time spent by all reduce tasks (ms)=39870
                Total vcore-milliseconds taken by all map tasks=204303
                Total vcore-milliseconds taken by all reduce tasks=39870
                Total megabyte-milliseconds taken by all map tasks=209206272
                Total megabyte-milliseconds taken by all reduce tasks=40826880
        Map-Reduce Framework
                Map input records=6001215
                Map output records=1201581
                Map output bytes=15349249
                Map output materialized bytes=17752441
                Input split bytes=500
                Combine input records=0
                Combine output records=0
                Reduce input groups=1043429
                Reduce shuffle bytes=17752441
                Reduce input records=1201581
                Reduce output records=1043429
                Spilled Records=2403162
                Shuffled Maps =5
                Failed Shuffles=0
                Merged Map outputs=5
                GC time elapsed (ms)=2118
                CPU time spent (ms)=27480
                Physical memory (bytes) snapshot=1177079808
                Virtual memory (bytes) snapshot=12632784896
                Total committed heap usage (bytes)=719736832
        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=594329385
        File Output Format Counters
                Bytes Written=13337968
1/10/27 22:22:28 INFO streaming.StreamJob: Output directory: /data/outputSubQuery2
ec2-user@ip-172-31-26-188 hadoop-2.6.4]$
```

First map reduce ran successfully and wrote output to /data/outputSubQuery2

Now using that output to run the second map reduce job

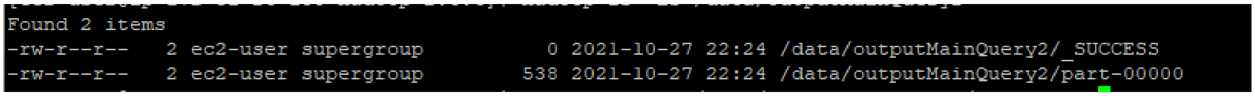
SECOND MAP REDUCE:

hadoop jar hadoop-streaming-2.6.4.jar -input /data/outputSubQuery2 -output /data/outputMainQuery2 -mapper MyMapper2.py -reducer MyReducer2.py -file ../MyReducer2.py -file ../MyMapper2.py



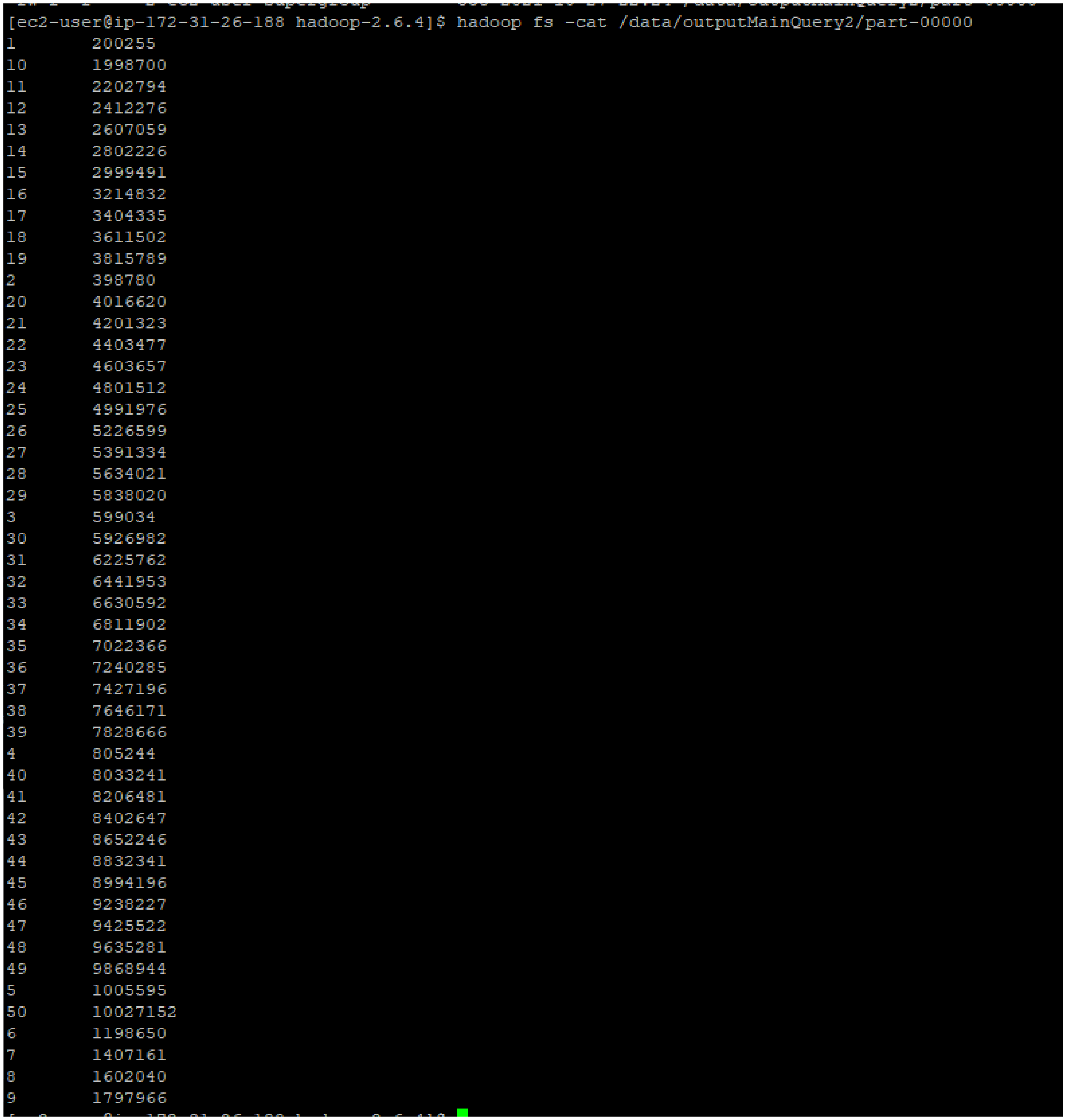
```
21/10/27 22:24:28 INFO mapreduce.Job: Job job_1635373262988_0002 running in uber mode : false
21/10/27 22:24:28 INFO mapreduce.Job:  map 0% reduce 0%
21/10/27 22:24:43 INFO mapreduce.Job:  map 83% reduce 0%
21/10/27 22:24:44 INFO mapreduce.Job:  map 100% reduce 0%
21/10/27 22:24:52 INFO mapreduce.Job:  map 100% reduce 100%
21/10/27 22:24:52 INFO mapreduce.Job: Job job_1635373262988_0002 completed successfully
21/10/27 22:24:52 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=6105040
                FILE: Number of bytes written=12540103
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=13342272
                HDFS: Number of bytes written=538
                HDFS: Number of read operations=9
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=26381
                Total time spent by all reduces in occupied slots (ms)=6022
                Total time spent by all map tasks (ms)=26381
                Total time spent by all reduce tasks (ms)=6022
                Total vcore-milliseconds taken by all map tasks=26381
                Total vcore-milliseconds taken by all reduce tasks=6022
                Total megabyte-milliseconds taken by all map tasks=27014144
                Total megabyte-milliseconds taken by all reduce tasks=6166528
        Map-Reduce Framework
                Map input records=1043429
                Map output records=481344
                Map output bytes=5142346
                Map output materialized bytes=6105046
                Input split bytes=208
                Combine input records=0
                Combine output records=0
                Reduce input groups=50
                Reduce shuffle bytes=6105046
                Reduce input records=481344
                Reduce output records=50
                Spilled Records=962688
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=378
                CPU time spent (ms)=4730
                Physical memory (bytes) snapshot=526508032
                Virtual memory (bytes) snapshot=6317748224
                Total committed heap usage (bytes)=307437568
        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=13342064
        File Output Format Counters
                Bytes Written=538
21/10/27 22:24:52 INFO streaming.StreamJob: Output directory: /data/outputMainQuery2
[ec2-user@ip-172-31-26-188 hadoop-2.6.4]$
```

It ran successfully and stored the final output to /data/outputMainQuery2



```
Found 2 items
-rw-r--r--   2 ec2-user supergroup          0 2021-10-27 22:24 /data/outputMainQuery2/_SUCCESS
-rw-r--r--   2 ec2-user supergroup        538 2021-10-27 22:24 /data/outputMainQuery2/part-00000
```

Output:



```
[ec2-user@ip-172-31-26-188 hadoop-2.6.4]$ hadoop fs -cat /data/outputMainQuery2/part-00000
1       200255
10      1998700
11      2202794
12      2412276
13      2607059
14      2802226
15      2999491
16      3214832
17      3404335
18      3611502
19      3815789
2       398780
20      4016620
21      4201323
22      4403477
23      4603657
24      4801512
25      4991976
26      5226599
27      5391334
28      5634021
29      5838020
3       599034
30      5926982
31      6225762
32      6441953
33      6630592
34      6811902
35      7022366
36      7240285
37      7427196
38      7646171
39      7828666
4       805244
40      8033241
41      8206481
42      8402647
43      8652246
44      8832341
45      8994196
46      9238227
47      9425522
48      9635281
49      9868944
5       1005595
50      10027152
6       1198650
7       1407161
8       1602040
9       1797966
```

I got the same output as when I tested it using pipes, so the two map reduce jobs ran successfully!