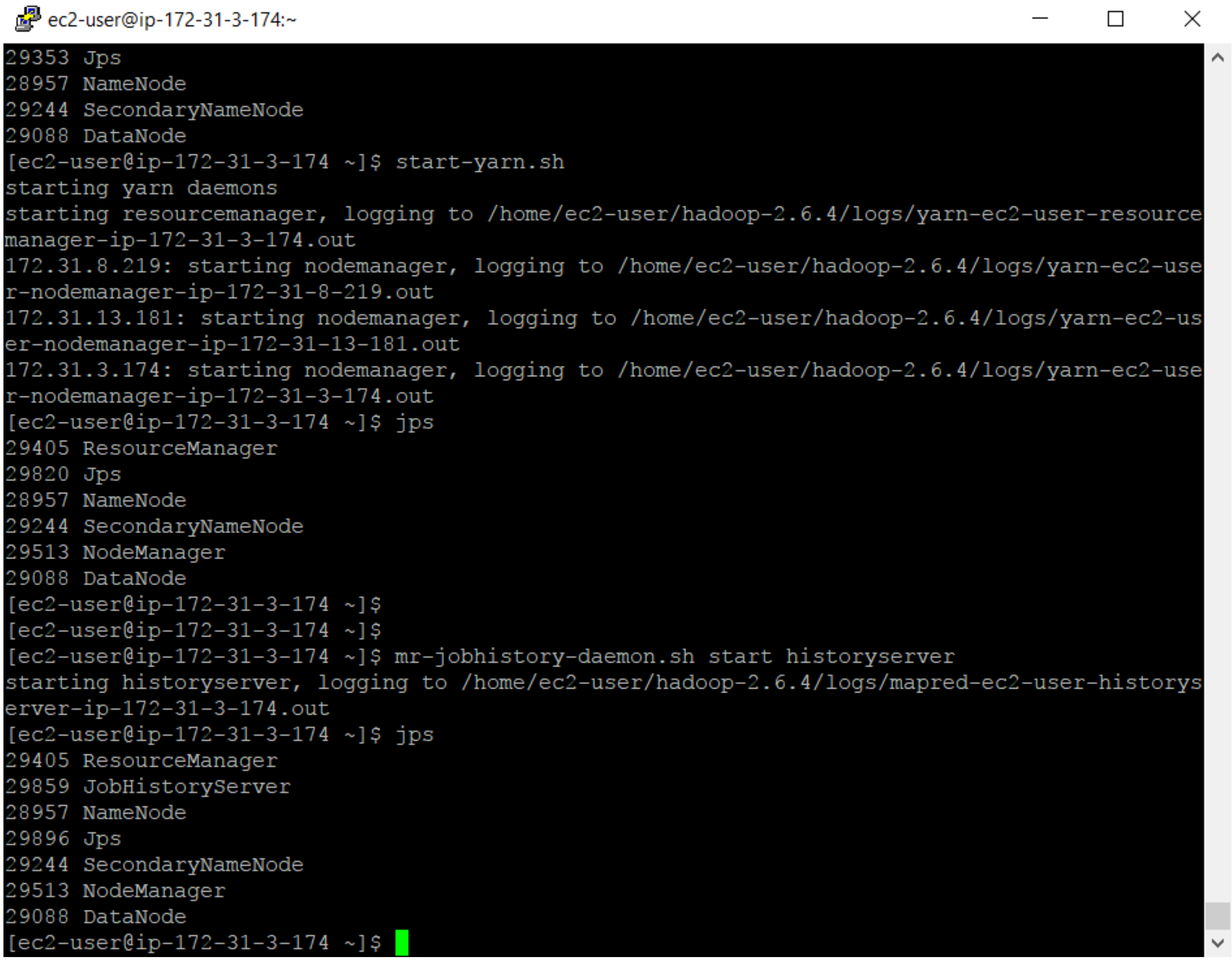
Kari Palmier
CSC 555 Winter 2018
Project Phase 1

Part 1 – Multi-Node Cluster

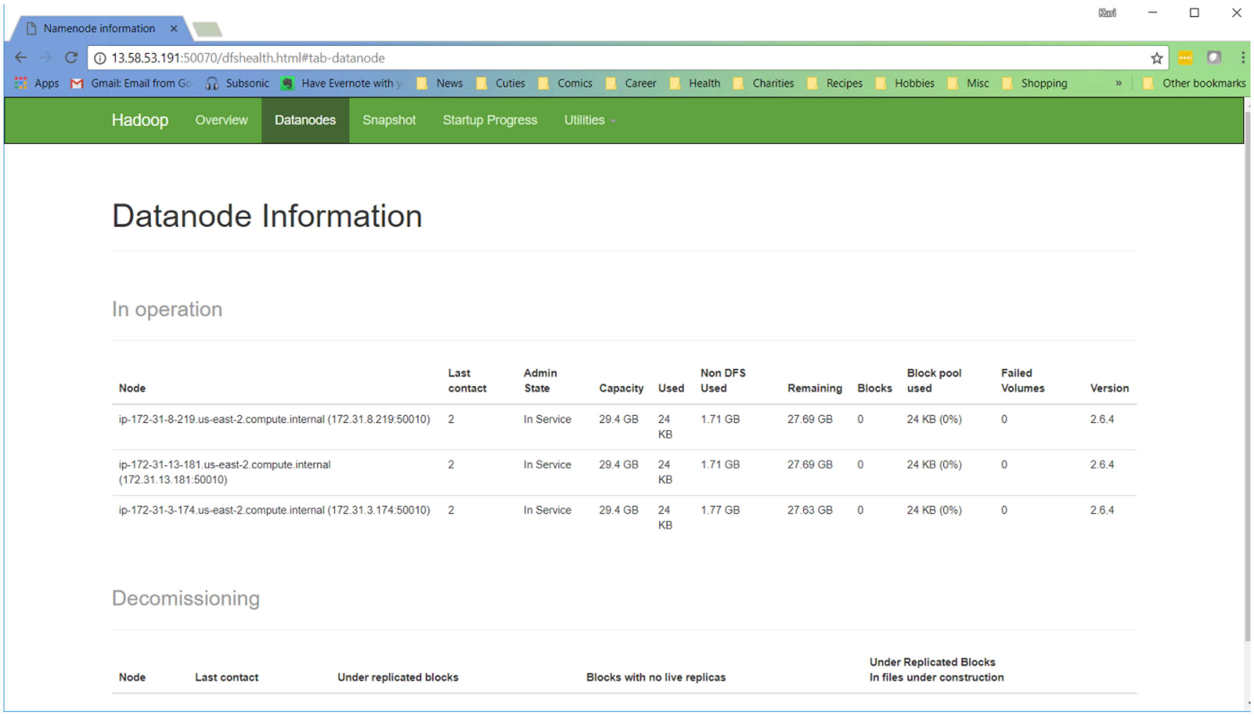
Cluster JPS process status after Hadoop installation on all 3 nodes (set up on master, then copied to nodes):



```
29353 Jps
28957 NameNode
29244 SecondaryNameNode
29088 DataNode
[ec2-user@ip-172-31-3-174 ~]$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /home/ec2-user/hadoop-2.6.4/logs/yarn-ec2-user-resource
manager-ip-172-31-3-174.out
172.31.8.219: starting nodemanager, logging to /home/ec2-user/hadoop-2.6.4/logs/yarn-ec2-use
r-nodemanager-ip-172-31-8-219.out
172.31.13.181: starting nodemanager, logging to /home/ec2-user/hadoop-2.6.4/logs/yarn-ec2-us
er-nodemanager-ip-172-31-13-181.out
172.31.3.174: starting nodemanager, logging to /home/ec2-user/hadoop-2.6.4/logs/yarn-ec2-use
r-nodemanager-ip-172-31-3-174.out
[ec2-user@ip-172-31-3-174 ~]$ jps
29405 ResourceManager
29820 Jps
28957 NameNode
29244 SecondaryNameNode
29513 NodeManager
29088 DataNode
[ec2-user@ip-172-31-3-174 ~]$
[ec2-user@ip-172-31-3-174 ~]$
[ec2-user@ip-172-31-3-174 ~]$ mr-jobhistory-daemon.sh start historyserver
starting historyserver, logging to /home/ec2-user/hadoop-2.6.4/logs/mapred-ec2-user-historys
erver-ip-172-31-3-174.out
[ec2-user@ip-172-31-3-174 ~]$ jps
29405 ResourceManager
29859 JobHistoryServer
28957 NameNode
29896 Jps
29244 SecondaryNameNode
29513 NodeManager
29088 DataNode
[ec2-user@ip-172-31-3-174 ~]$
```

Browser cluster verification (shows all 3 nodes up and working):

Browser Datanode Information:



Datanode Information

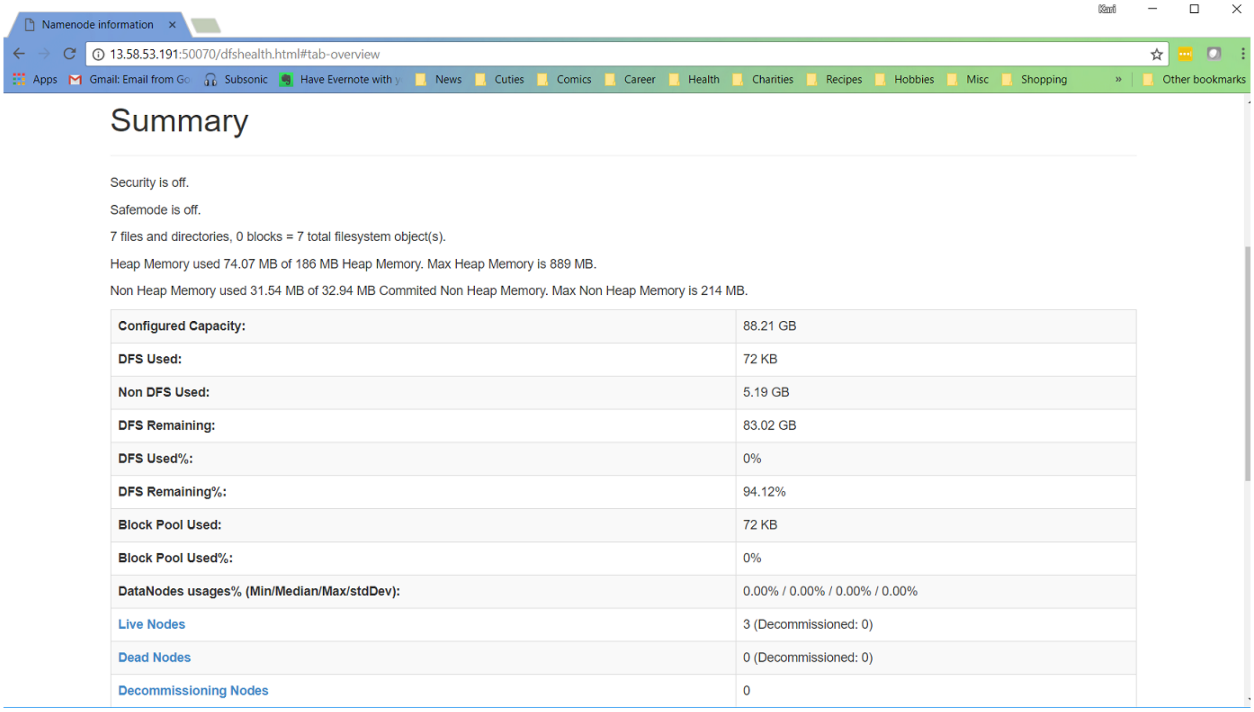
In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
ip-172-31-8-219.us-east-2.compute.internal (172.31.8.219:50010)	2	In Service	29.4 GB	24 KB	1.71 GB	27.69 GB	0	24 KB (0%)	0	2.6.4
ip-172-31-13-181.us-east-2.compute.internal (172.31.13.181:50010)	2	In Service	29.4 GB	24 KB	1.71 GB	27.69 GB	0	24 KB (0%)	0	2.6.4
ip-172-31-3-174.us-east-2.compute.internal (172.31.3.174:50010)	2	In Service	29.4 GB	24 KB	1.77 GB	27.63 GB	0	24 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

Browser Summary Information:



Summary

Security is off.

Safemode is off.

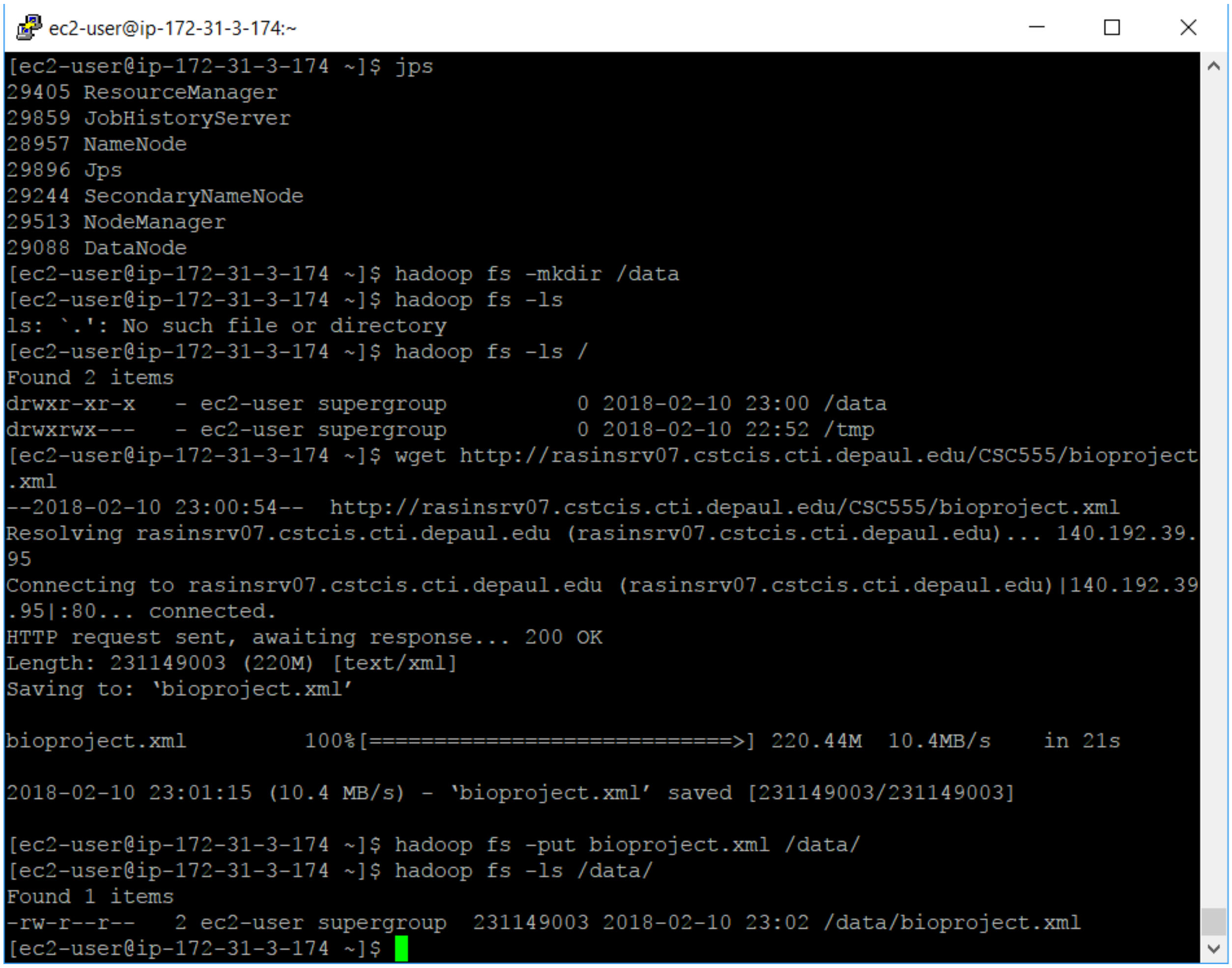
7 files and directories, 0 blocks = 7 total filesystem object(s).

Heap Memory used 74.07 MB of 186 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 31.54 MB of 32.94 MB Commited Non Heap Memory. Max Non Heap Memory is 214 MB.

Configured Capacity:	88.21 GB
DFS Used:	72 KB
Non DFS Used:	5.19 GB
DFS Remaining:	83.02 GB
DFS Used%:	0%
DFS Remaining%:	94.12%
Block Pool Used:	72 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

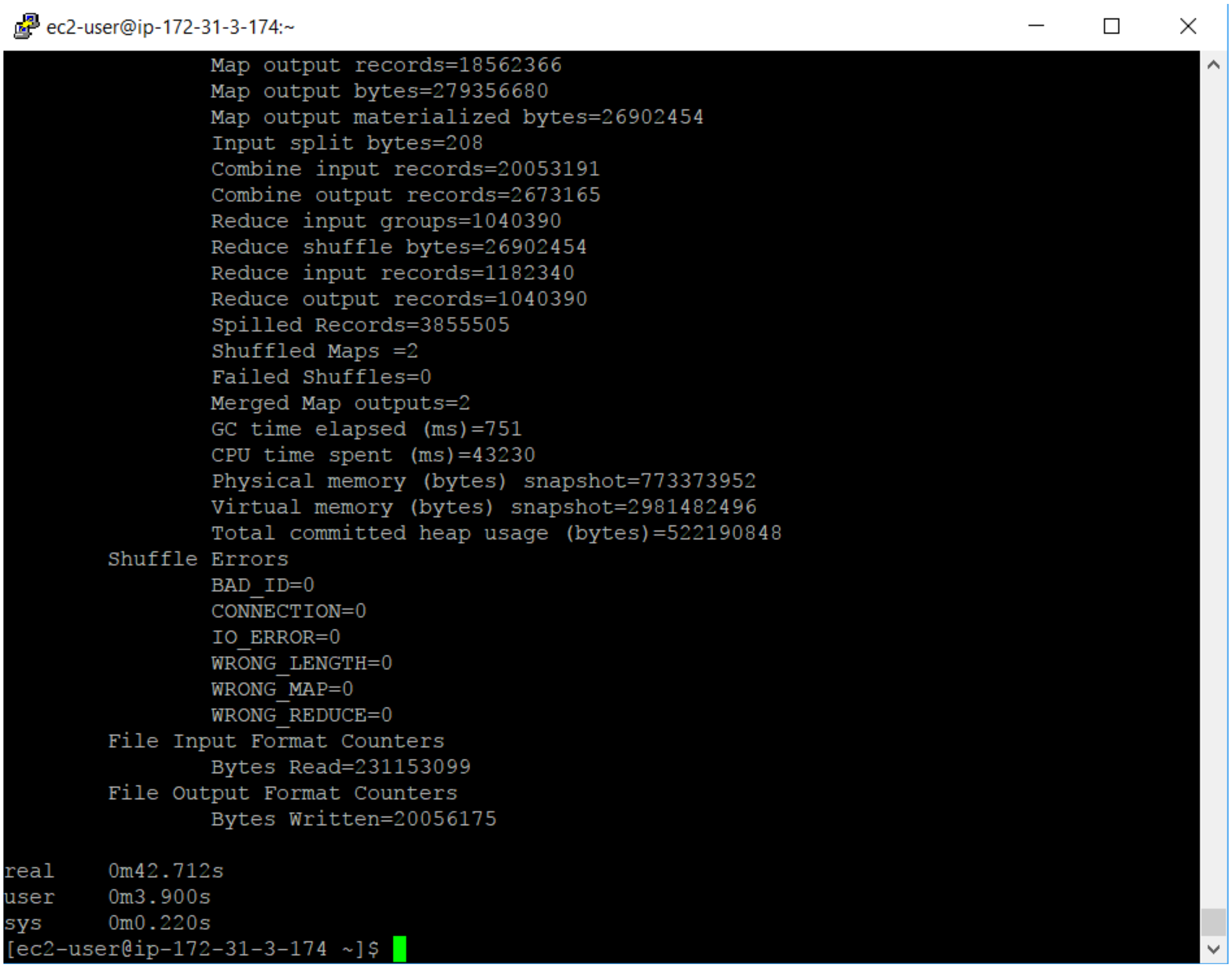
Bioproject.xml file download and placement into HDFS data directory:



```
[ec2-user@ip-172-31-3-174 ~]$ jps
29405 ResourceManager
29859 JobHistoryServer
28957 NameNode
29896 Jps
29244 SecondaryNameNode
29513 NodeManager
29088 DataNode
[ec2-user@ip-172-31-3-174 ~]$ hadoop fs -mkdir /data
[ec2-user@ip-172-31-3-174 ~]$ hadoop fs -ls
ls: `.': No such file or directory
[ec2-user@ip-172-31-3-174 ~]$ hadoop fs -ls /
Found 2 items
drwxr-xr-x   - ec2-user supergroup          0 2018-02-10 23:00 /data
drwxrwx---   - ec2-user supergroup          0 2018-02-10 22:52 /tmp
[ec2-user@ip-172-31-3-174 ~]$ wget http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/bioproject
.xml
--2018-02-10 23:00:54--  http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/bioproject.xml
Resolving rasinsrv07.cstcis.cti.depaul.edu (rasinsrv07.cstcis.cti.depaul.edu)... 140.192.39.
95
Connecting to rasinsrv07.cstcis.cti.depaul.edu (rasinsrv07.cstcis.cti.depaul.edu)|140.192.39
.95|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 231149003 (220M) [text/xml]
Saving to: ‘bioproject.xml’

bioproject.xml         100%[============================>] 220.44M  10.4MB/s    in 21s

2018-02-10 23:01:15 (10.4 MB/s) - ‘bioproject.xml’ saved [231149003/231149003]

[ec2-user@ip-172-31-3-174 ~]$ hadoop fs -put bioproject.xml /data/
[ec2-user@ip-172-31-3-174 ~]$ hadoop fs -ls /data/
Found 1 items
-rw-r--r--   2 ec2-user supergroup  231149003 2018-02-10 23:02 /data/bioproject.xml
[ec2-user@ip-172-31-3-174 ~]$
```

Output of Wordcount command:

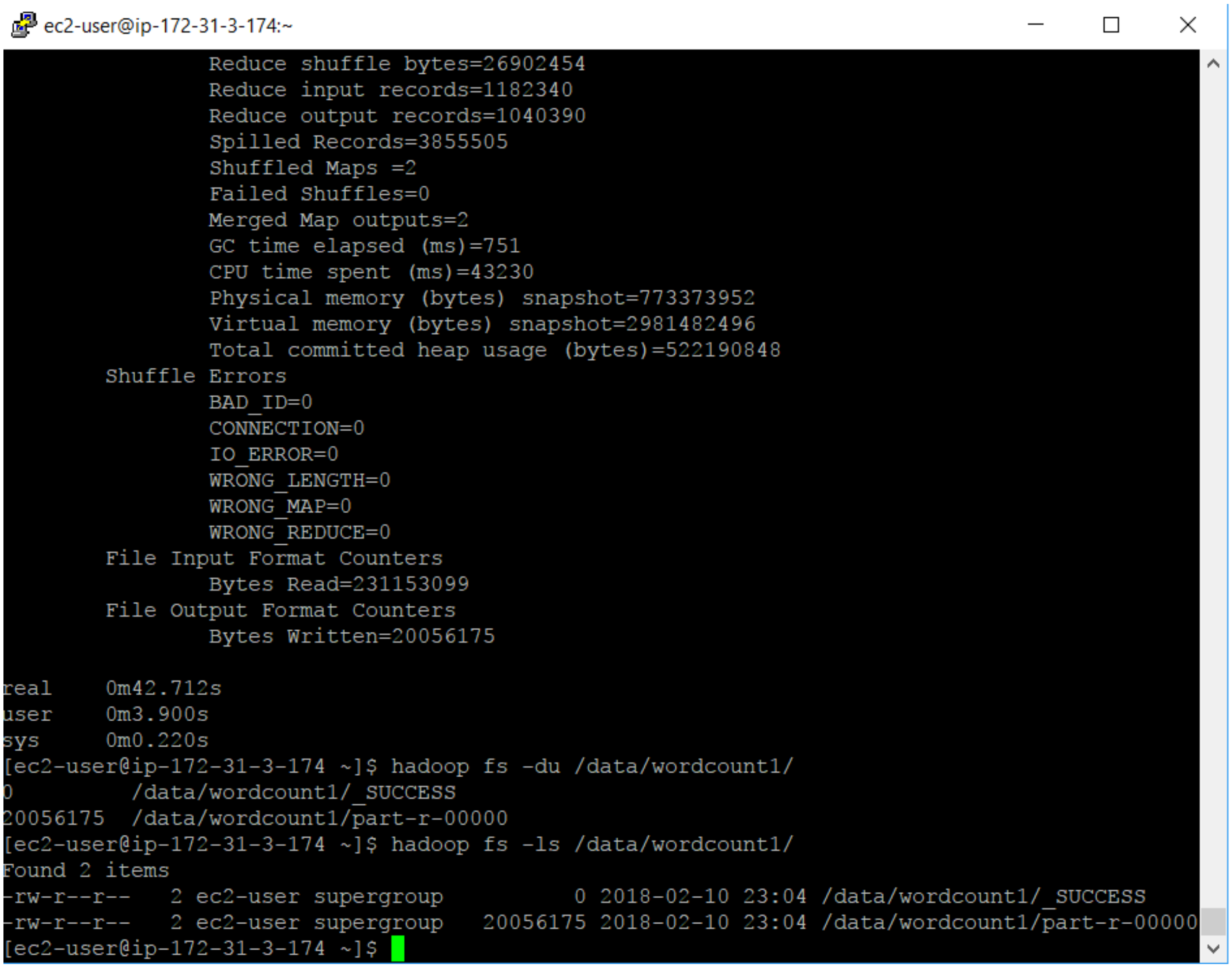
Command run: time Hadoop jar Hadoop-2.6.4/share/Hadoop/mapreduce/Hadoop-mapreduce-examples-2.6.4.jar wordcount /data/bioproject.xml /data/wordcount1



```
                Map output records=18562366
                Map output bytes=279356680
                Map output materialized bytes=26902454
                Input split bytes=208
                Combine input records=20053191
                Combine output records=2673165
                Reduce input groups=1040390
                Reduce shuffle bytes=26902454
                Reduce input records=1182340
                Reduce output records=1040390
                Spilled Records=3855505
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=751
                CPU time spent (ms)=43230
                Physical memory (bytes) snapshot=773373952
                Virtual memory (bytes) snapshot=2981482496
                Total committed heap usage (bytes)=522190848
        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    0m42.712s
user    0m3.900s
sys     0m0.220s
[ec2-user@ip-172-31-3-174 ~]$
```

It took 42.712 seconds to run wordcount.

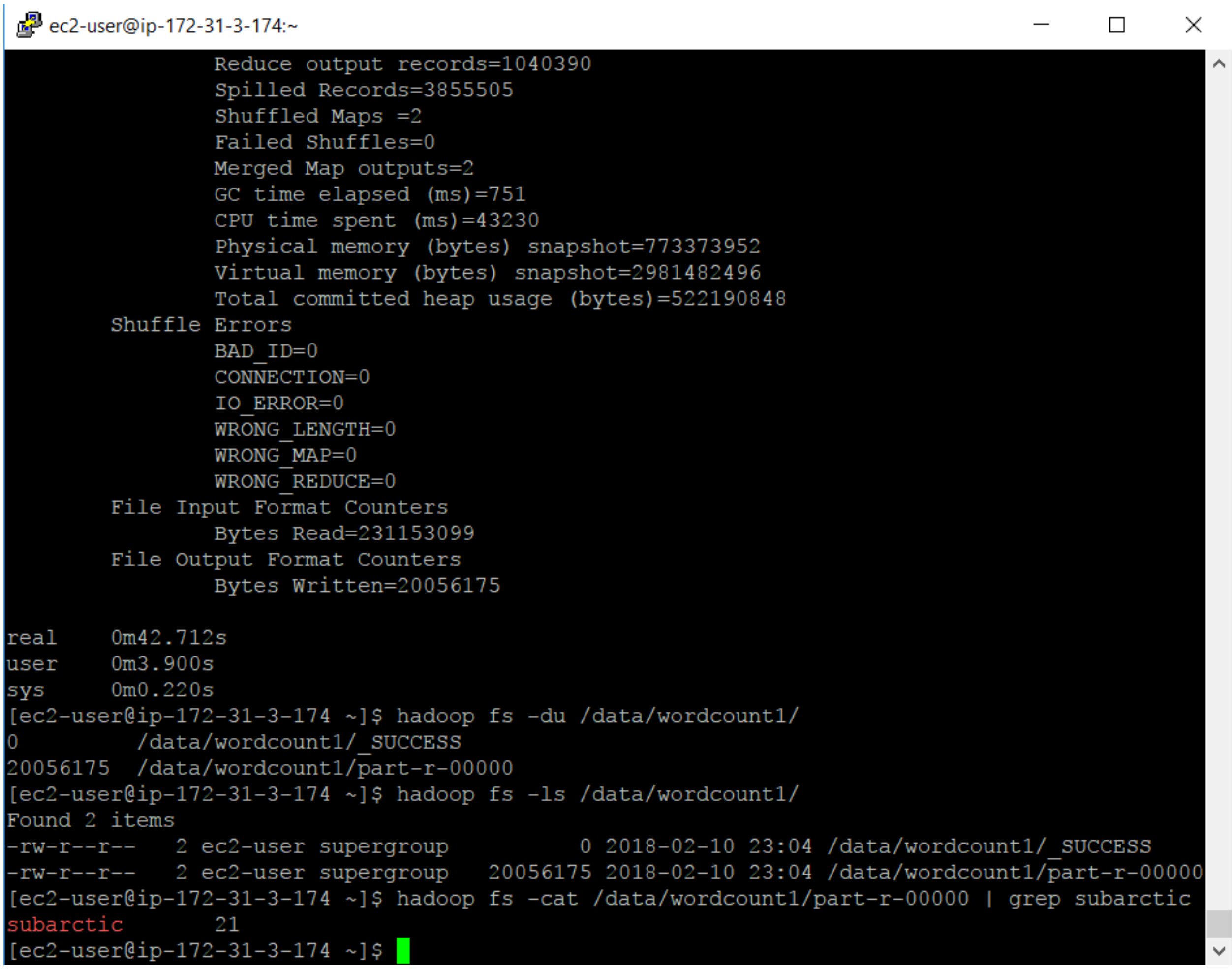
Size of wordcount output file generated:



```
                Reduce shuffle bytes=26902454
                Reduce input records=1182340
                Reduce output records=1040390
                Spilled Records=3855505
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=751
                CPU time spent (ms)=43230
                Physical memory (bytes) snapshot=773373952
                Virtual memory (bytes) snapshot=2981482496
                Total committed heap usage (bytes)=522190848
        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    0m42.712s
user    0m3.900s
sys     0m0.220s
[ec2-user@ip-172-31-3-174 ~]$ hadoop fs -du /data/wordcount1/
0         /data/wordcount1/_SUCCESS
20056175  /data/wordcount1/part-r-00000
[ec2-user@ip-172-31-3-174 ~]$ hadoop fs -ls /data/wordcount1/
Found 2 items
-rw-r--r--   2 ec2-user supergroup          0 2018-02-10 23:04 /data/wordcount1/_SUCCESS
-rw-r--r--   2 ec2-user supergroup   20056175 2018-02-10 23:04 /data/wordcount1/part-r-00000
[ec2-user@ip-172-31-3-174 ~]$
```

Size of the part-r-00000 file created is 20,056,175 bytes.

Number of occurrences of the word “subarctic” found by wordcount:



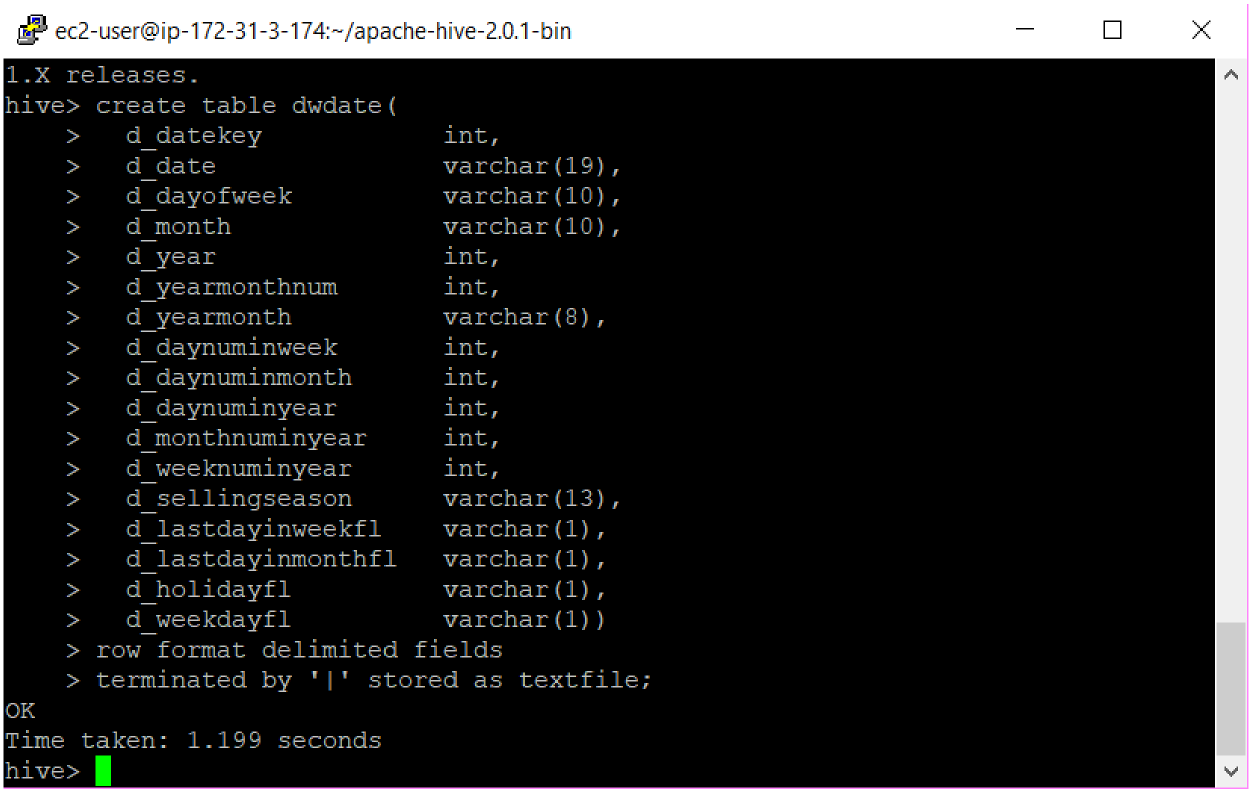
```
                Reduce output records=1040390
                Spilled Records=3855505
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=751
                CPU time spent (ms)=43230
                Physical memory (bytes) snapshot=773373952
                Virtual memory (bytes) snapshot=2981482496
                Total committed heap usage (bytes)=522190848
        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    0m42.712s
user    0m3.900s
sys     0m0.220s
[ec2-user@ip-172-31-3-174 ~]$ hadoop fs -du /data/wordcount1/
0         /data/wordcount1/_SUCCESS
20056175  /data/wordcount1/part-r-00000
[ec2-user@ip-172-31-3-174 ~]$ hadoop fs -ls /data/wordcount1/
Found 2 items
-rw-r--r--   2 ec2-user supergroup          0 2018-02-10 23:04 /data/wordcount1/_SUCCESS
-rw-r--r--   2 ec2-user supergroup   20056175 2018-02-10 23:04 /data/wordcount1/part-r-00000
[ec2-user@ip-172-31-3-174 ~]$ hadoop fs -cat /data/wordcount1/part-r-00000 | grep subarctic
subarctic       21
[ec2-user@ip-172-31-3-174 ~]$
```

Number of occurrences of subarctic is 21.

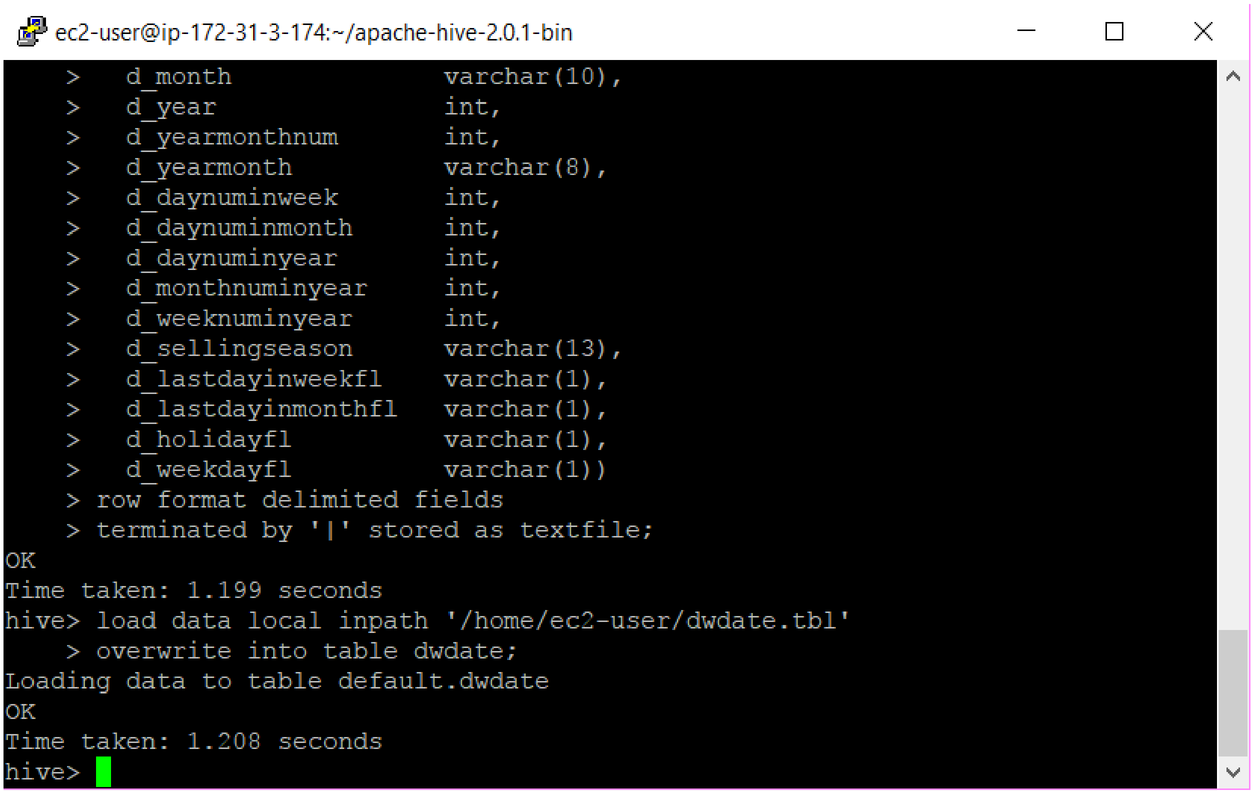
Part 2 – Hive

Hive Table Creation Code:

```
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;

load data local inpath '/home/ec2-user/dwdate.tbl'
overwrite into table dwdate;
```

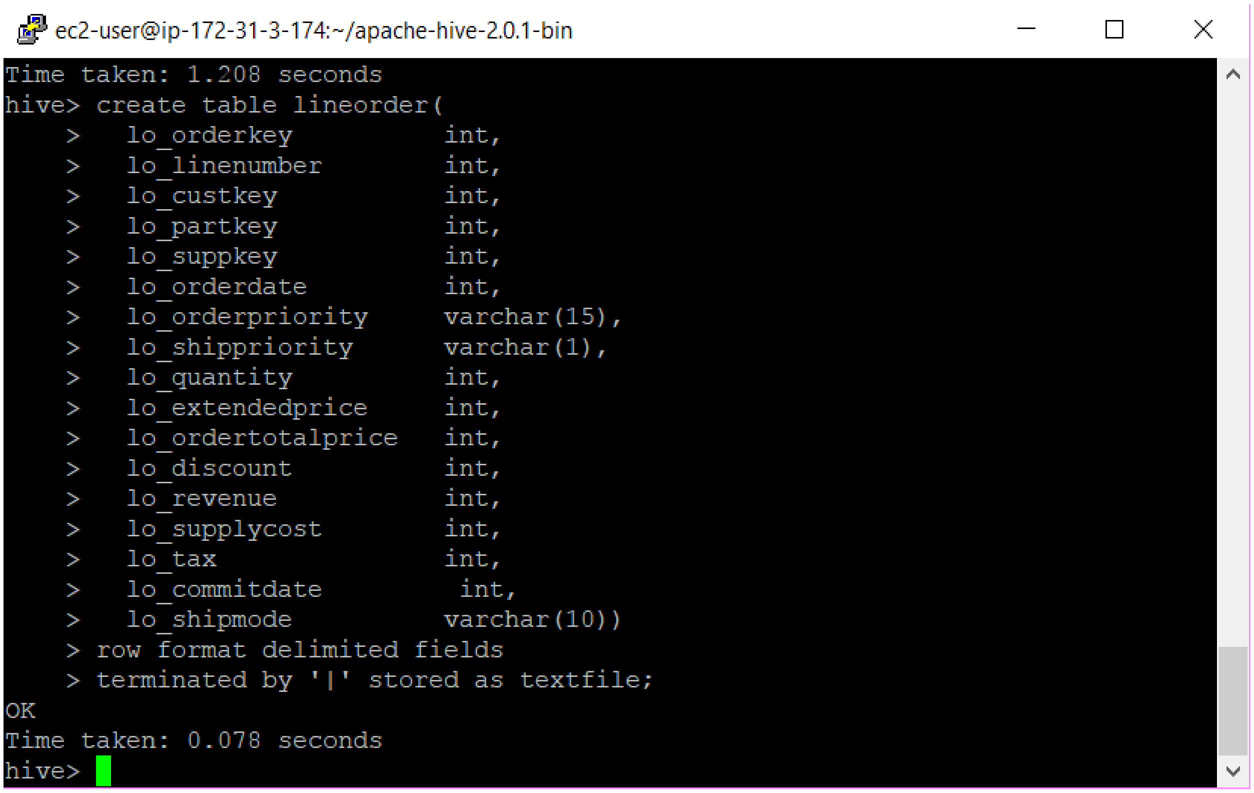

ec2-user@ip-172-31-3-174:~/apache-hive-2.0.1-bin

```
1.X releases.
hive> 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;
OK
Time taken: 1.199 seconds
hive>
```


ec2-user@ip-172-31-3-174:~/apache-hive-2.0.1-bin

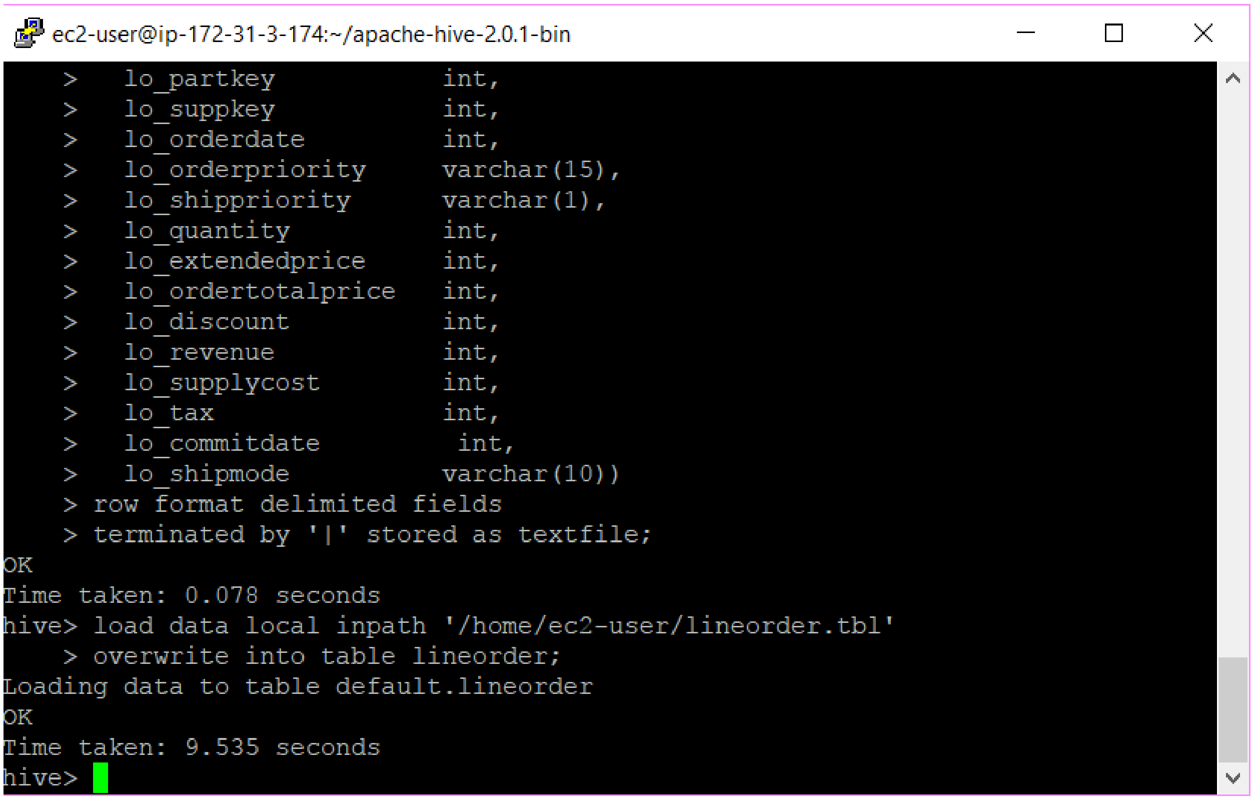
```
    >   d_month                varchar(10),
    >   d_year                 int,
    >   d_yearmonthnum         int,
    >   d_yearmonth            varchar(8),
    >   d_daynuminweek         int,
    >   d_daynuminmonth        int,
    >   d_daynuminyear         int,
    >   d_monthnuminyear       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;
OK
Time taken: 1.199 seconds
hive> load data local inpath '/home/ec2-user/dwdate.tbl'
    > overwrite into table dwdate;
Loading data to table default.dwdate
OK
Time taken: 1.208 seconds
hive>
```

```
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;

load data local inpath '/home/ec2-user/lineorder.tbl'
overwrite into table lineorder;
```



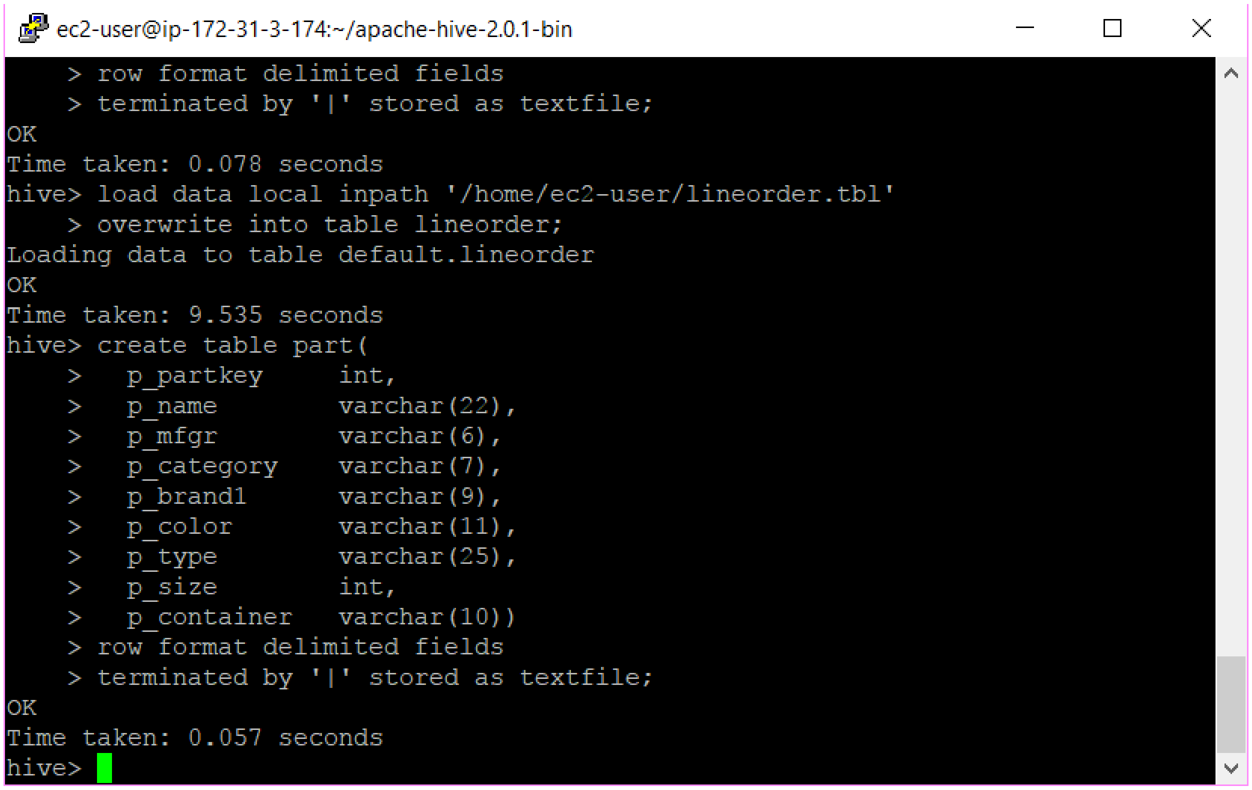
ec2-user@ip-172-31-3-174:~/apache-hive-2.0.1-bin

```
Time taken: 1.208 seconds
hive> 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;
OK
Time taken: 0.078 seconds
hive> 
```

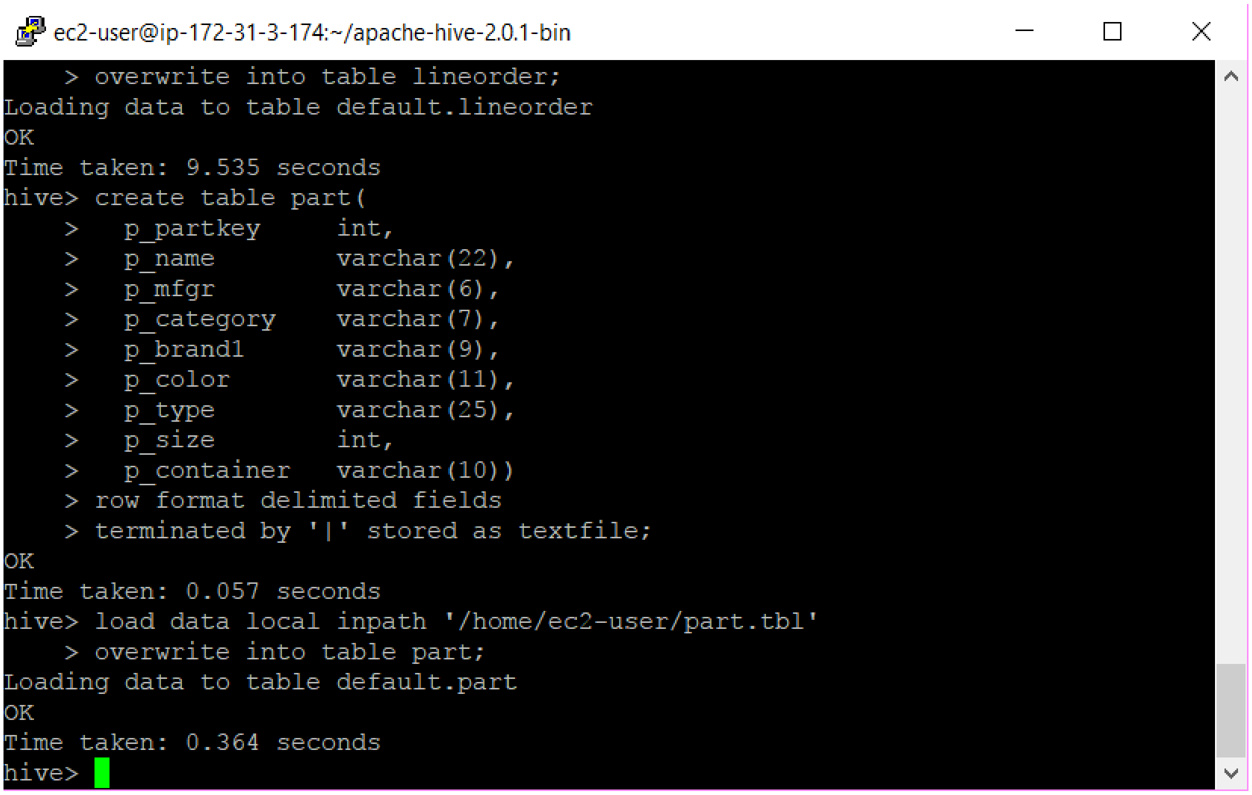


```
    >   lo_partkey           int,
    >   lo_suppkey           int,
    >   lo_orderdate         int,
    >   lo_orderpriority     varchar(15),
    >   lo_shippriority      varchar(1),
    >   lo_quantity          int,
    >   lo_extendedprice     int,
    >   lo_ordertotalprice   int,
    >   lo_discount          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;
OK
Time taken: 0.078 seconds
hive> load data local inpath '/home/ec2-user/lineorder.tbl'
    > overwrite into table lineorder;
Loading data to table default.lineorder
OK
Time taken: 9.535 seconds
hive> 
```

```
create table part(
  p_partkey       int,
  p_name          varchar(22),
  p_mfgr          varchar(6),
  p_category      varchar(7),
  p_brand1        varchar(9),
  p_color         varchar(11),
  p_type          varchar(25),
  p_size          int,
  p_container     varchar(10))
row format delimited fields
terminated by '|' stored as textfile;

load data local inpath '/home/ec2-user/part.tbl'
overwrite into table part;
```

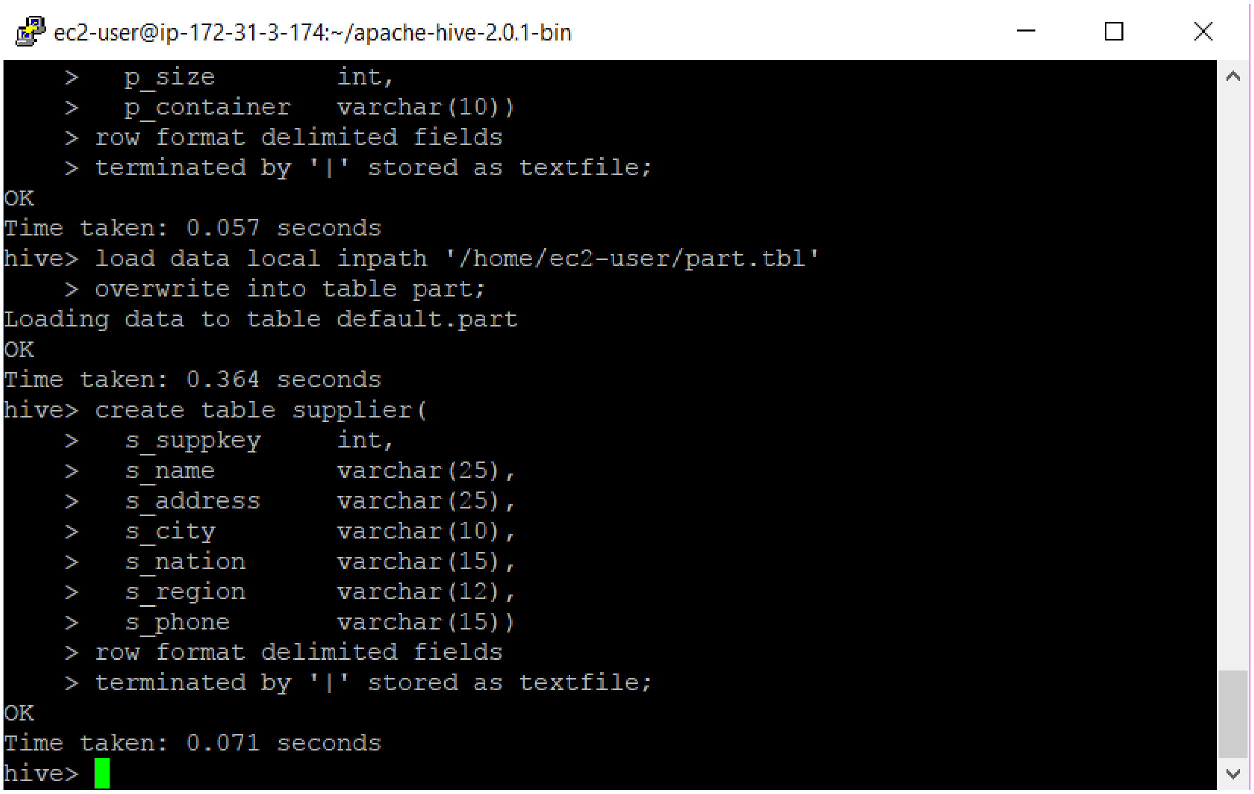


```
    > row format delimited fields
    > terminated by '|' stored as textfile;
OK
Time taken: 0.078 seconds
hive> load data local inpath '/home/ec2-user/lineorder.tbl'
    > overwrite into table lineorder;
Loading data to table default.lineorder
OK
Time taken: 9.535 seconds
hive> create table part(
    >   p_partkey     int,
    >   p_name        varchar(22),
    >   p_mfgr        varchar(6),
    >   p_category    varchar(7),
    >   p_brand1      varchar(9),
    >   p_color       varchar(11),
    >   p_type        varchar(25),
    >   p_size        int,
    >   p_container   varchar(10))
    > row format delimited fields
    > terminated by '|' stored as textfile;
OK
Time taken: 0.057 seconds
hive>
```

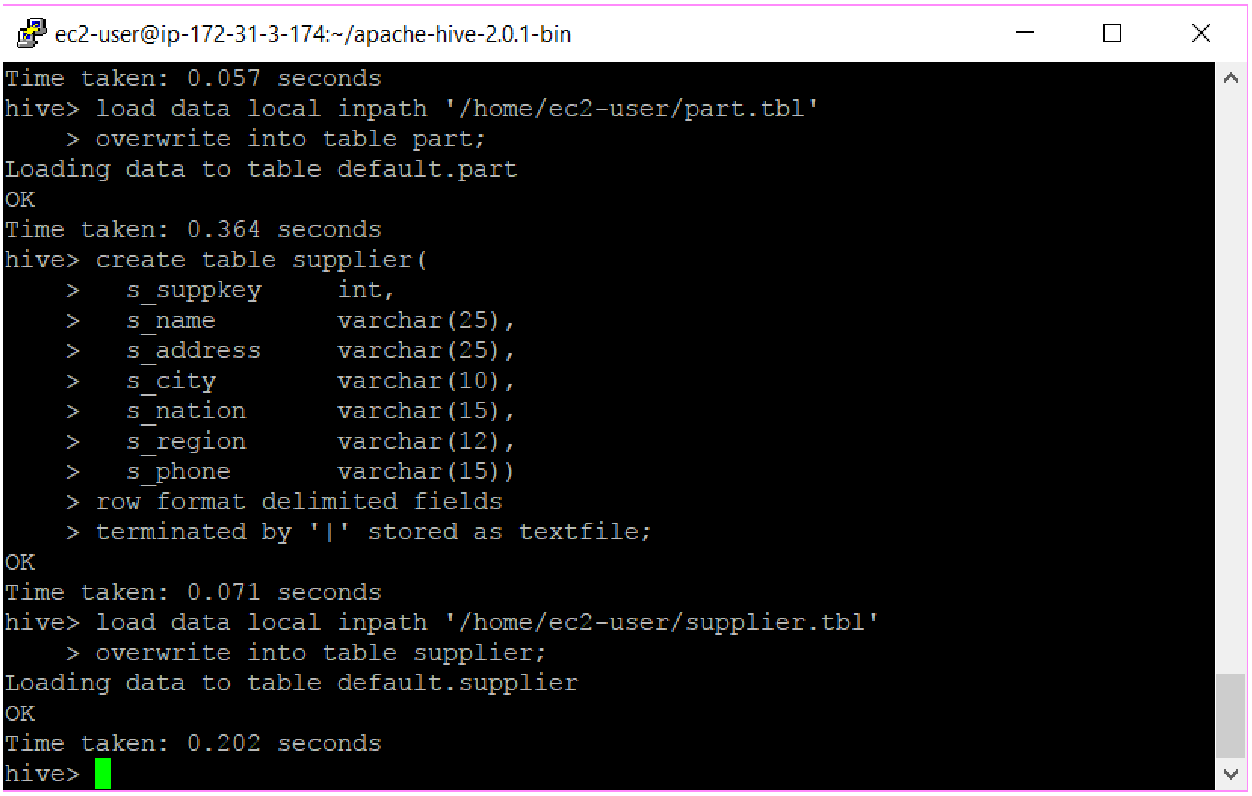


```
    > overwrite into table lineorder;
Loading data to table default.lineorder
OK
Time taken: 9.535 seconds
hive> create table part(
    >   p_partkey     int,
    >   p_name        varchar(22),
    >   p_mfgr        varchar(6),
    >   p_category    varchar(7),
    >   p_brand1      varchar(9),
    >   p_color       varchar(11),
    >   p_type        varchar(25),
    >   p_size        int,
    >   p_container   varchar(10))
    > row format delimited fields
    > terminated by '|' stored as textfile;
OK
Time taken: 0.057 seconds
hive> load data local inpath '/home/ec2-user/part.tbl'
    > overwrite into table part;
Loading data to table default.part
OK
Time taken: 0.364 seconds
hive> 
```

```
create table supplier(
  s_suppkey       int,
  s_name          varchar(25),
  s_address       varchar(25),
  s_city          varchar(10),
  s_nation        varchar(15),
  s_region        varchar(12),
  s_phone         varchar(15))
row format delimited fields
terminated by '|' stored as textfile;

load data local inpath '/home/ec2-user/supplier.tbl'
overwrite into table supplier;
```

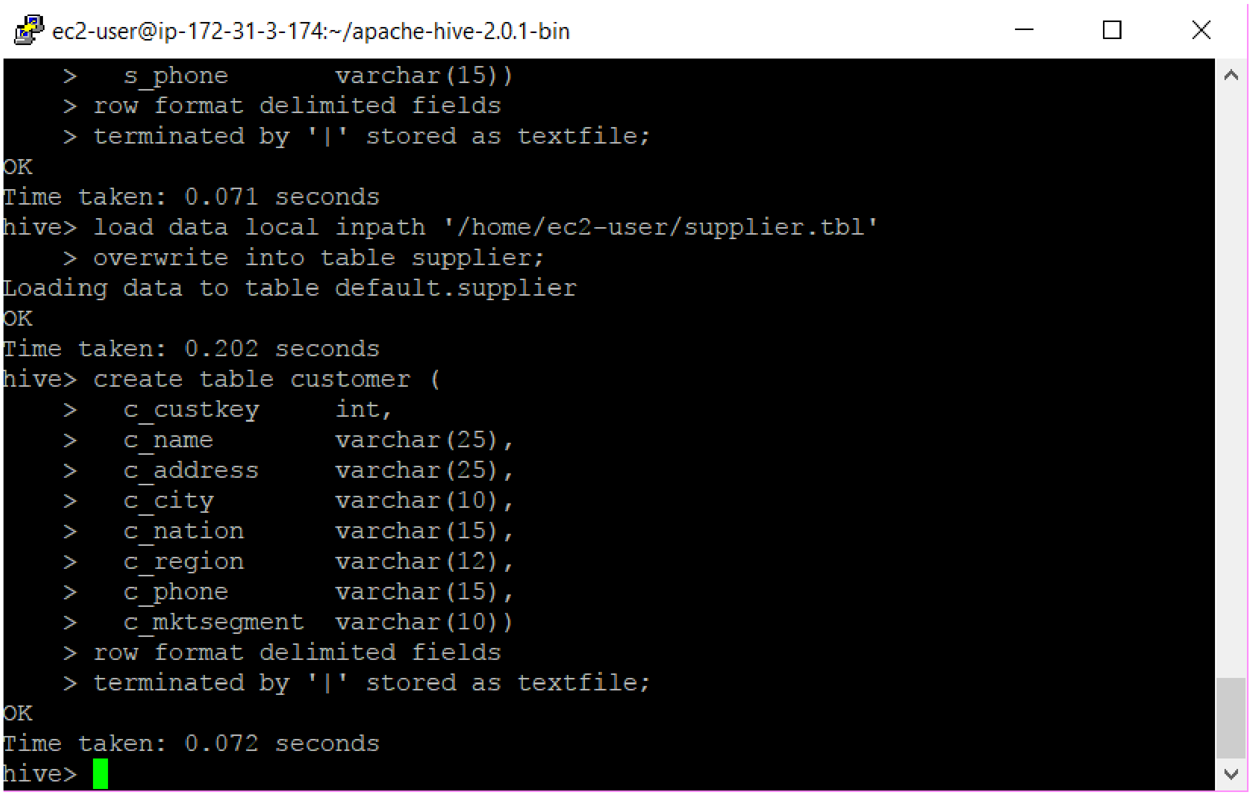


```
    >   p_size         int,
    >   p_container    varchar(10))
    > row format delimited fields
    > terminated by '|' stored as textfile;
OK
Time taken: 0.057 seconds
hive> load data local inpath '/home/ec2-user/part.tbl'
    > overwrite into table part;
Loading data to table default.part
OK
Time taken: 0.364 seconds
hive> create table supplier(
    >   s_suppkey      int,
    >   s_name         varchar(25),
    >   s_address      varchar(25),
    >   s_city         varchar(10),
    >   s_nation       varchar(15),
    >   s_region       varchar(12),
    >   s_phone        varchar(15))
    > row format delimited fields
    > terminated by '|' stored as textfile;
OK
Time taken: 0.071 seconds
hive>
```

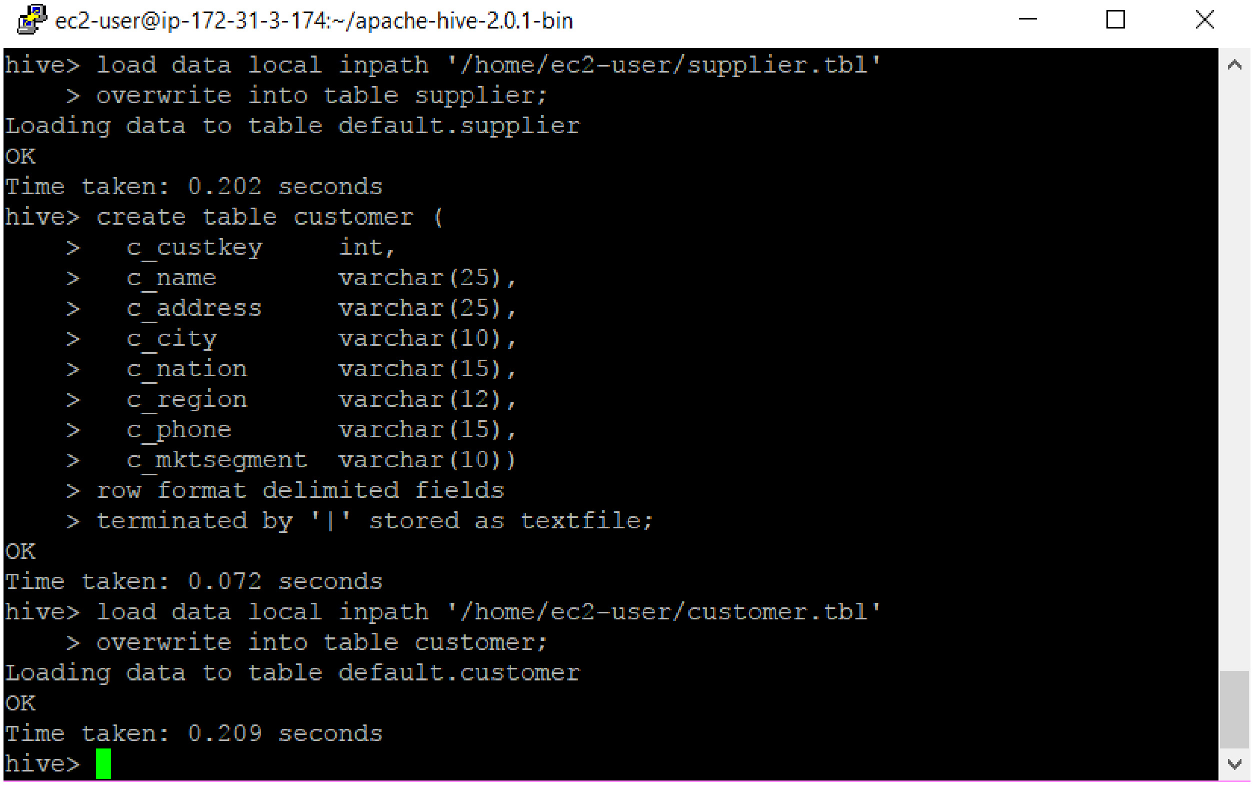


```
Time taken: 0.057 seconds
hive> load data local inpath '/home/ec2-user/part.tbl'
    > overwrite into table part;
Loading data to table default.part
OK
Time taken: 0.364 seconds
hive> create table supplier(
    >   s_suppkey     int,
    >   s_name        varchar(25),
    >   s_address     varchar(25),
    >   s_city        varchar(10),
    >   s_nation      varchar(15),
    >   s_region      varchar(12),
    >   s_phone       varchar(15))
    > row format delimited fields
    > terminated by '|' stored as textfile;
OK
Time taken: 0.071 seconds
hive> load data local inpath '/home/ec2-user/supplier.tbl'
    > overwrite into table supplier;
Loading data to table default.supplier
OK
Time taken: 0.202 seconds
hive>
```

```
create table customer (
  c_custkey              int,
  c_name                 varchar(25),
  c_address              varchar(25),
  c_city                 varchar(10),
  c_nation               varchar(15),
  c_region               varchar(12),
  c_phone                varchar(15),
  c_mktsegment           varchar(10))
row format delimited fields
terminated by '|' stored as textfile;

load data local inpath '/home/ec2-user/customer.tbl'
overwrite into table customer;
```



```
    >   s_phone        varchar(15))
    > row format delimited fields
    > terminated by '|' stored as textfile;
OK
Time taken: 0.071 seconds
hive> load data local inpath '/home/ec2-user/supplier.tbl'
    > overwrite into table supplier;
Loading data to table default.supplier
OK
Time taken: 0.202 seconds
hive> create table customer (
    >   c_custkey      int,
    >   c_name         varchar(25),
    >   c_address      varchar(25),
    >   c_city         varchar(10),
    >   c_nation       varchar(15),
    >   c_region       varchar(12),
    >   c_phone        varchar(15),
    >   c_mktsegment   varchar(10))
    > row format delimited fields
    > terminated by '|' stored as textfile;
OK
Time taken: 0.072 seconds
hive> 
```



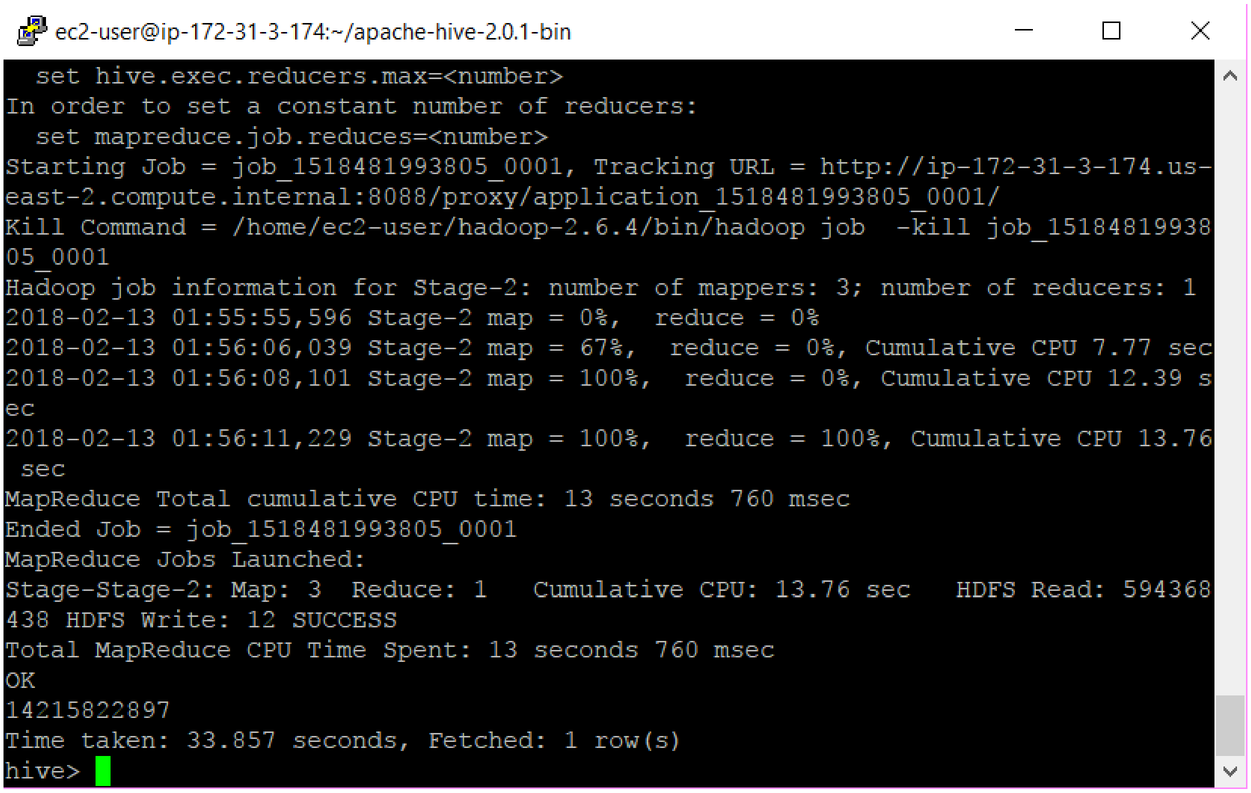
```
hive> load data local inpath '/home/ec2-user/supplier.tbl'
    > overwrite into table supplier;
Loading data to table default.supplier
OK
Time taken: 0.202 seconds
hive> create table customer (
    >   c_custkey     int,
    >   c_name        varchar(25),
    >   c_address     varchar(25),
    >   c_city        varchar(10),
    >   c_nation      varchar(15),
    >   c_region      varchar(12),
    >   c_phone       varchar(15),
    >   c_mktsegment  varchar(10))
    > row format delimited fields
    > terminated by '|' stored as textfile;
OK
Time taken: 0.072 seconds
hive> load data local inpath '/home/ec2-user/customer.tbl'
    > overwrite into table customer;
Loading data to table default.customer
OK
Time taken: 0.209 seconds
hive>
```

Query 1.2 Hive Code and Execution

Hive Query Code:

```
select sum(lo_extendedprice) as revenue
from lineorder, dwdate
where lo_orderdate = d_datekey
  and d_yearmonth = 'Jan1993'
  and lo_discount between 5 and 6
  and lo_quantity between 25 and 35;
```

Hive Query Execution:



```
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1518481993805_0001, Tracking URL = http://ip-172-31-3-174.us-
east-2.compute.internal:8088/proxy/application_1518481993805_0001/
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job  -kill job_15184819938
05_0001
Hadoop job information for Stage-2: number of mappers: 3; number of reducers: 1
2018-02-13 01:55:55,596 Stage-2 map = 0%,  reduce = 0%
2018-02-13 01:56:06,039 Stage-2 map = 67%,  reduce = 0%, Cumulative CPU 7.77 sec
2018-02-13 01:56:08,101 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 12.39 s
ec
2018-02-13 01:56:11,229 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 13.76
 sec
MapReduce Total cumulative CPU time: 13 seconds 760 msec
Ended Job = job_1518481993805_0001
MapReduce Jobs Launched:
Stage-Stage-2: Map: 3  Reduce: 1   Cumulative CPU: 13.76 sec   HDFS Read: 594368
438 HDFS Write: 12 SUCCESS
Total MapReduce CPU Time Spent: 13 seconds 760 msec
OK
14215822897
Time taken: 33.857 seconds, Fetched: 1 row(s)
hive> 
```

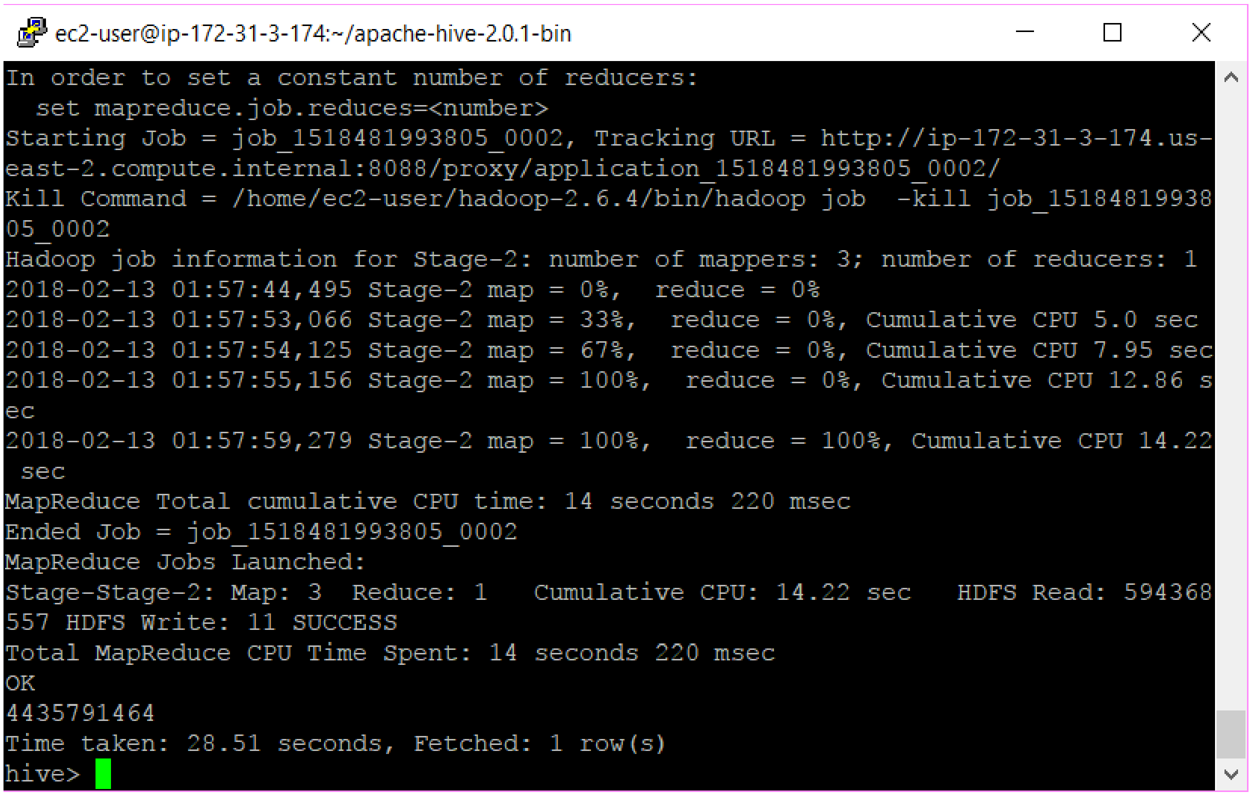
The sum of lo_extendedprice reported is 1,421,582,287. The query took 13.76 secs to run.

Query 1.3 Hive Code and Execution

Hive Query Code:

```
select sum(lo_extendedprice) as revenue
from lineorder, dwdate
where lo_orderdate = d_datekey
  and d_weeknuminyear = 6 and d_year = 1994
  and lo_discount between 5 and 8
  and lo_quantity between 36 and 41;
```

Hive Query Execution:



```
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1518481993805_0002, Tracking URL = http://ip-172-31-3-174.us-
east-2.compute.internal:8088/proxy/application_1518481993805_0002/
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job  -kill job_15184819938
05_0002
Hadoop job information for Stage-2: number of mappers: 3; number of reducers: 1
2018-02-13 01:57:44,495 Stage-2 map = 0%,  reduce = 0%
2018-02-13 01:57:53,066 Stage-2 map = 33%,  reduce = 0%, Cumulative CPU 5.0 sec
2018-02-13 01:57:54,125 Stage-2 map = 67%,  reduce = 0%, Cumulative CPU 7.95 sec
2018-02-13 01:57:55,156 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 12.86 s
ec
2018-02-13 01:57:59,279 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 14.22
 sec
MapReduce Total cumulative CPU time: 14 seconds 220 msec
Ended Job = job_1518481993805_0002
MapReduce Jobs Launched:
Stage-Stage-2: Map: 3  Reduce: 1   Cumulative CPU: 14.22 sec   HDFS Read: 594368
557 HDFS Write: 11 SUCCESS
Total MapReduce CPU Time Spent: 14 seconds 220 msec
OK
4435791464
Time taken: 28.51 seconds, Fetched: 1 row(s)
hive>
```

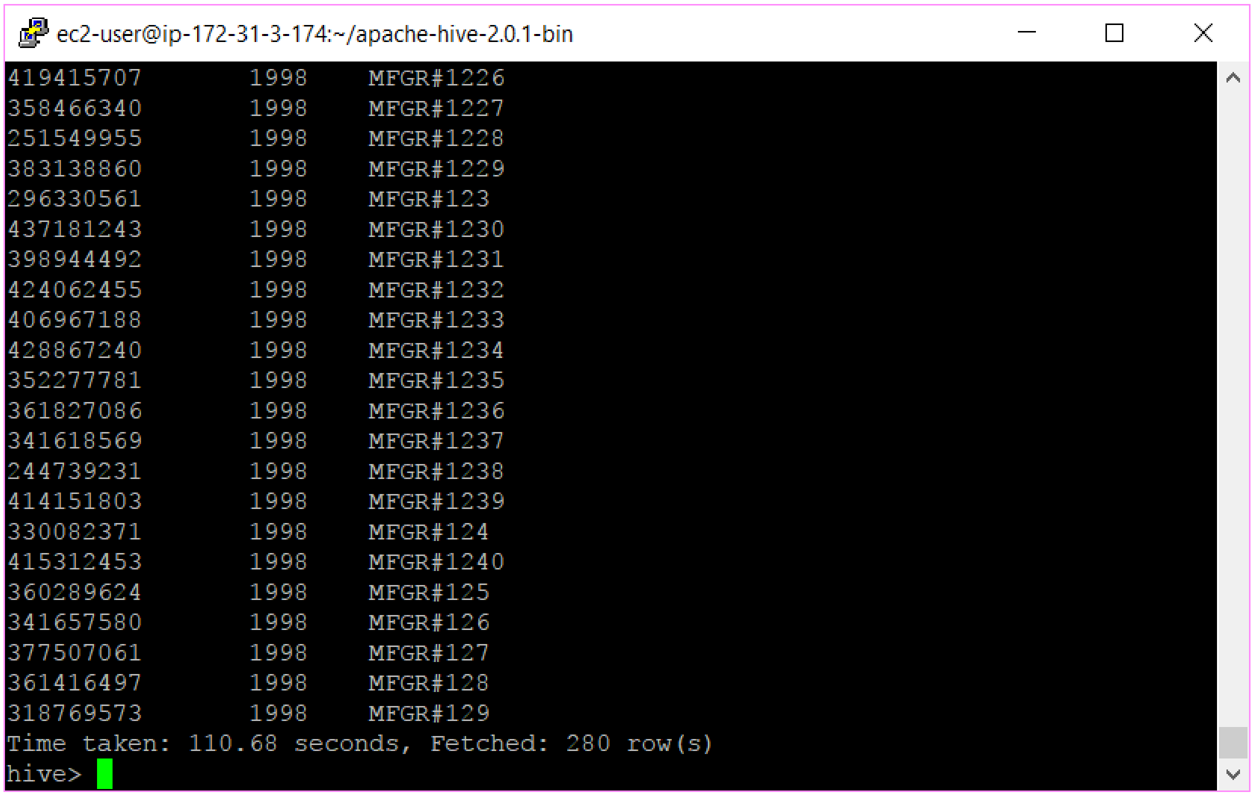
The sum of lo_extendedprice reported is4,435,791,464. The query took 14.22 secs to run.

Query 2.1 Hive Code and Execution

Hive Query Code:

```
select sum(lo_revenue), d_year, p_brand1
from lineorder, dwdate, part, supplier
where lo_orderdate = d_datekey
  and lo_partkey = p_partkey
  and lo_suppkey = s_suppkey
  and p_category = 'MFGR#12'
  and s_region = 'AMERICA'
group by d_year, p_brand1
order by d_year, p_brand1;
```

Hive Query Execution:



```
419415707       1998    MFGR#1226
358466340       1998    MFGR#1227
251549955       1998    MFGR#1228
383138860       1998    MFGR#1229
296330561       1998    MFGR#123
437181243       1998    MFGR#1230
398944492       1998    MFGR#1231
424062455       1998    MFGR#1232
406967188       1998    MFGR#1233
428867240       1998    MFGR#1234
352277781       1998    MFGR#1235
361827086       1998    MFGR#1236
341618569       1998    MFGR#1237
244739231       1998    MFGR#1238
414151803       1998    MFGR#1239
330082371       1998    MFGR#124
415312453       1998    MFGR#1240
360289624       1998    MFGR#125
341657580       1998    MFGR#126
377507061       1998    MFGR#127
361416497       1998    MFGR#128
318769573       1998    MFGR#129
Time taken: 110.68 seconds, Fetched: 280 row(s)
hive>
```

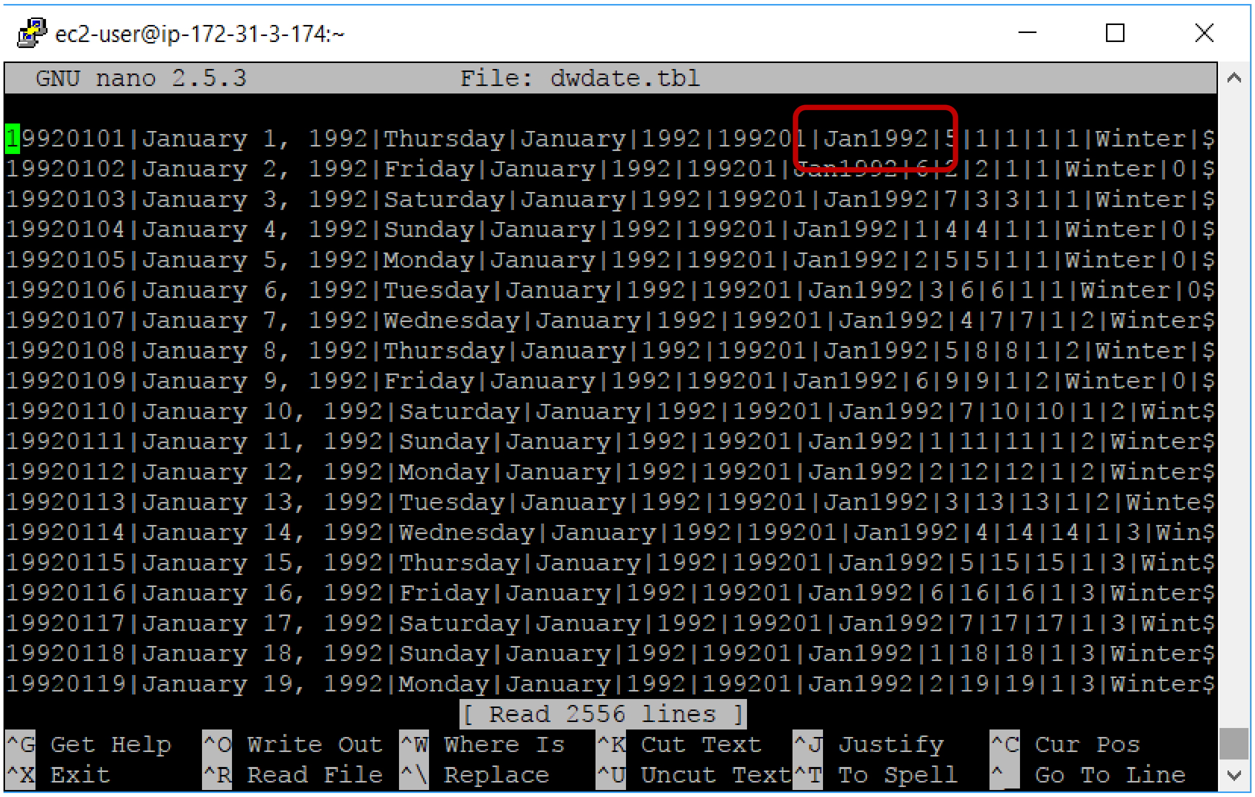
The query took 110.68 secs to run.

dwdate Table Transformation

Original dwdate table stored in Hive:

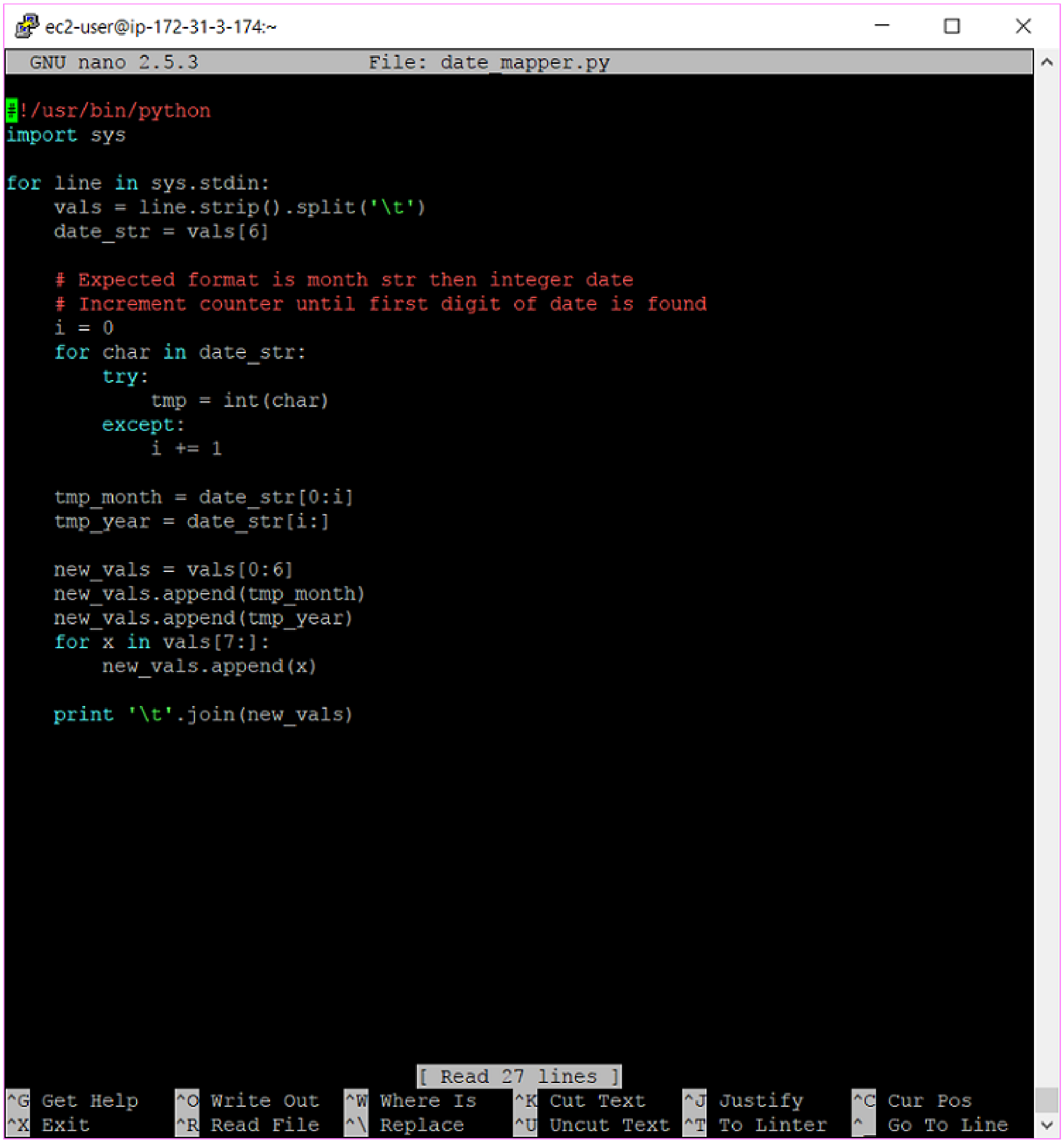
Copied the Hive dwdate table from HDFS to Linux home directory with the following command run from the Linux /home/ec2-user/ directory:
hadoop fs -get /user/hive/warehouse/dwdate/dwdate.tbl

Viewed the copied dwdate.tbl in Nano with nano dwdate.tbl.



```
 GNU nano 2.5.3              File: dwdate.tbl

19920101|January 1, 1992|Thursday|January|1992|199201|Jan1992|5|1|1|1|1|Winter|$
19920102|January 2, 1992|Friday|January|1992|199201|Jan1992|6|2|2|1|1|Winter|0|$
19920103|January 3, 1992|Saturday|January|1992|199201|Jan1992|7|3|3|1|1|Winter|$
19920104|January 4, 1992|Sunday|January|1992|199201|Jan1992|1|4|4|1|1|Winter|0|$
19920105|January 5, 1992|Monday|January|1992|199201|Jan1992|2|5|5|1|1|Winter|0|$
19920106|January 6, 1992|Tuesday|January|1992|199201|Jan1992|3|6|6|1|1|Winter|0$
19920107|January 7, 1992|Wednesday|January|1992|199201|Jan1992|4|7|7|1|2|Winter$
19920108|January 8, 1992|Thursday|January|1992|199201|Jan1992|5|8|8|1|2|Winter|$
19920109|January 9, 1992|Friday|January|1992|199201|Jan1992|6|9|9|1|2|Winter|0|$
19920110|January 10, 1992|Saturday|January|1992|199201|Jan1992|7|10|10|1|2|Wint$
19920111|January 11, 1992|Sunday|January|1992|199201|Jan1992|1|11|11|1|2|Winter$
19920112|January 12, 1992|Monday|January|1992|199201|Jan1992|2|12|12|1|2|Winter$
19920113|January 13, 1992|Tuesday|January|1992|199201|Jan1992|3|13|13|1|2|Winte$
19920114|January 14, 1992|Wednesday|January|1992|199201|Jan1992|4|14|14|1|3|Win$
19920115|January 15, 1992|Thursday|January|1992|199201|Jan1992|5|15|15|1|3|Wint$
19920116|January 16, 1992|Friday|January|1992|199201|Jan1992|6|16|16|1|3|Winter$
19920117|January 17, 1992|Saturday|January|1992|199201|Jan1992|7|17|17|1|3|Wint$
19920118|January 18, 1992|Sunday|January|1992|199201|Jan1992|1|18|18|1|3|Winter$
19920119|January 19, 1992|Monday|January|1992|199201|Jan1992|2|19|19|1|3|Winter$
                              [ Read 2556 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

Python transformation code that splits the contents of the original column 17 (Jan1992) into 2 columns (Jan 1992):



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

for line in sys.stdin:
    vals = line.strip().split('\t')
    date_str = vals[6]

    # Expected format is month str then integer date
    # Increment counter until first digit of date is found
    i = 0
    for char in date_str:
        try:
            tmp = int(char)
        except:
            i += 1

    tmp_month = date_str[0:i]
    tmp_year = date_str[i:]

    new_vals = vals[0:6]
    new_vals.append(tmp_month)
    new_vals.append(tmp_year)
    for x in vals[7:]:
        new_vals.append(x)

    print '\t'.join(new_vals)
```

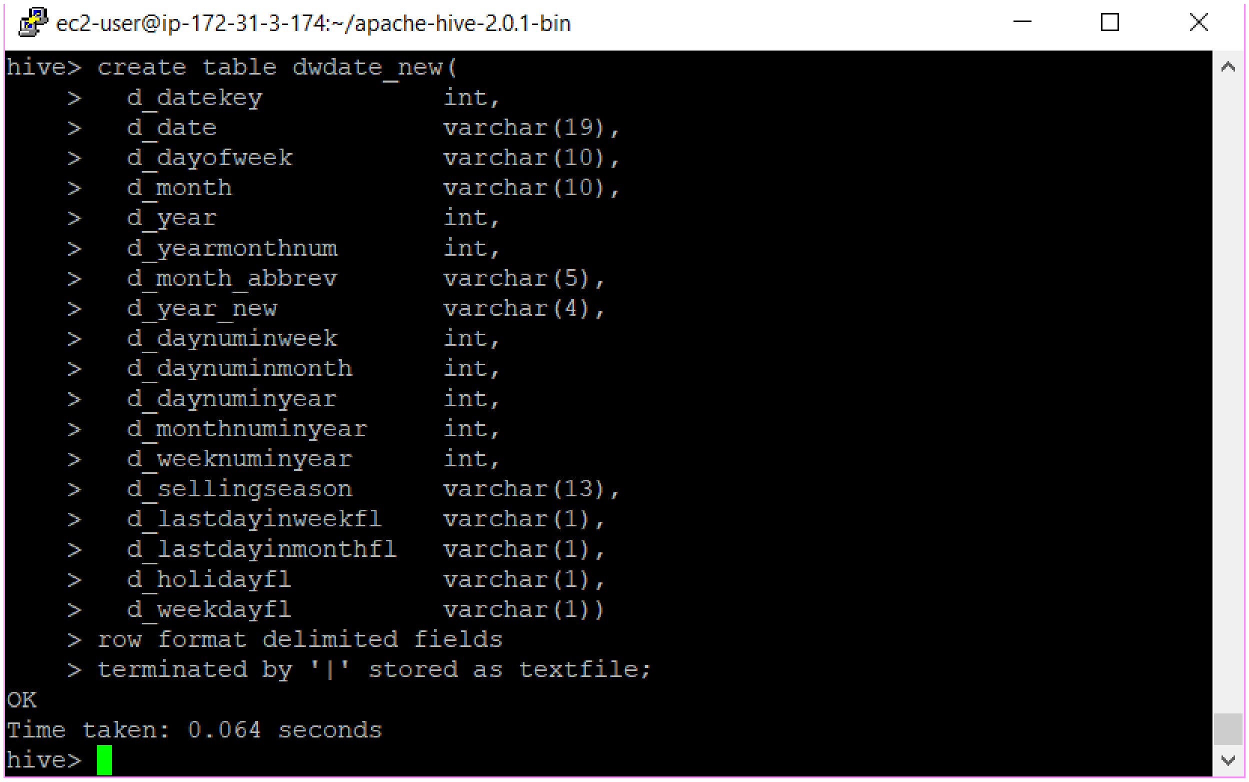
[Read 27 lines]

Create new Hive dwdate_new table that has 18 columns (one for the month abbreviation of original column 17 and one for the year of original column 17):

Hive dwdate_new Table Creation Code:

```
create table dwdate_new(
  d_datekey           int,
  d_date              varchar(19),
  d_dayofweek         varchar(10),
  d_month             varchar(10),
  d_year              int,
  d_yearmonthnum      int,
  d_month_abbrev      varchar(5),
  d_year_new          varchar(4),
  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;
```

Hive dwdate_new Table Creation Execution:



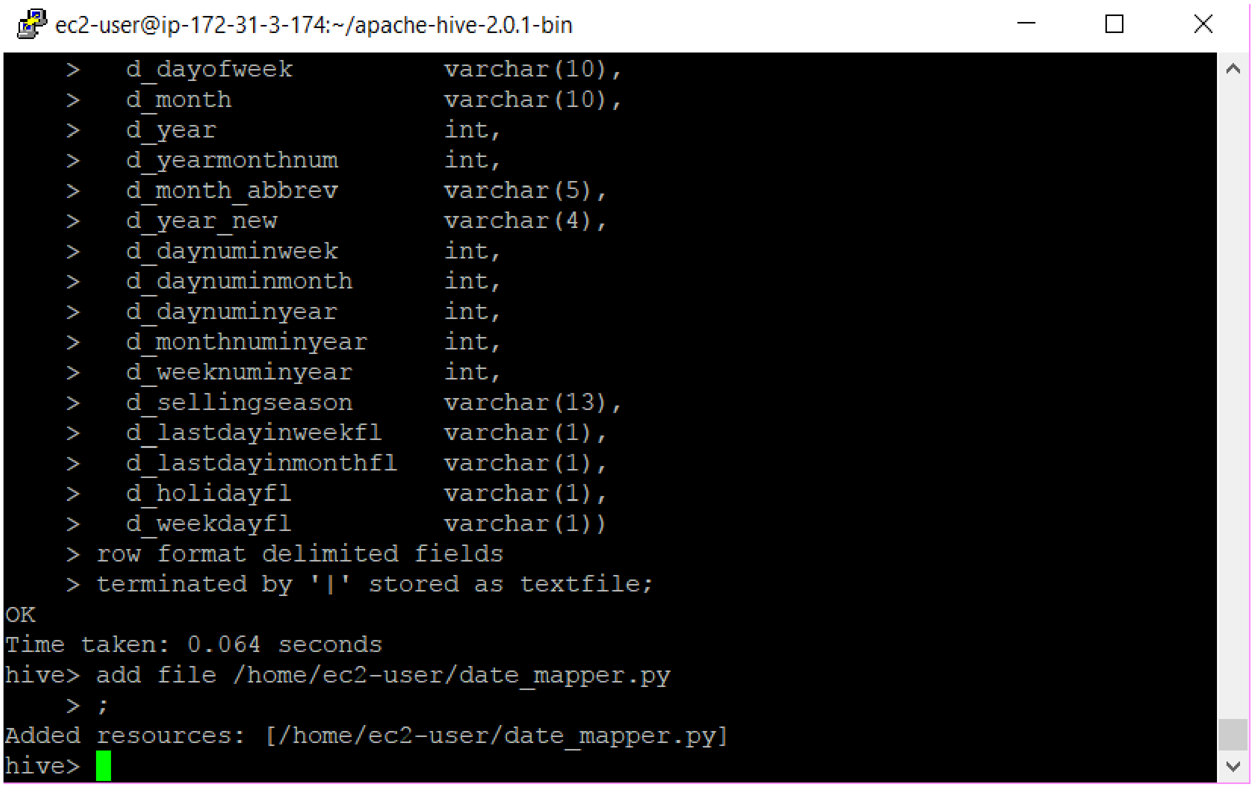
```
hive> create table dwdate_new(
    >   d_datekey                 int,
    >   d_date                    varchar(19),
    >   d_dayofweek               varchar(10),
    >   d_month                   varchar(10),
    >   d_year                    int,
    >   d_yearmonthnum            int,
    >   d_month_abbrev            varchar(5),
    >   d_year_new                varchar(4),
    >   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;
OK
Time taken: 0.064 seconds
hive>
```

Hive add file command to add python transformation file:

Hive Transformation Add Code:

add file /home/ec2-user/date_mapper.py;

Hive Transformation Add Execution:



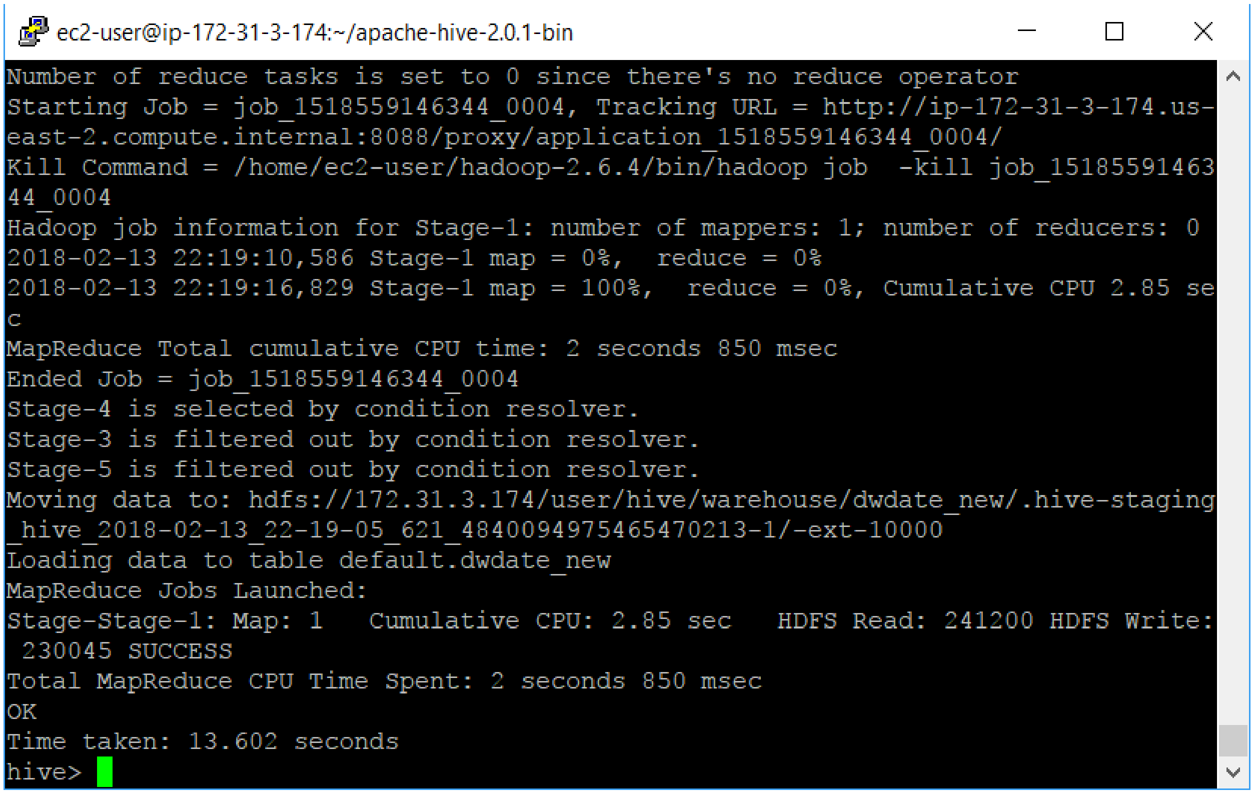
```
    >   d_dayofweek          varchar(10),
    >   d_month              varchar(10),
    >   d_year               int,
    >   d_yearmonthnum       int,
    >   d_month_abbrev       varchar(5),
    >   d_year_new           varchar(4),
    >   d_daynuminweek       int,
    >   d_daynuminmonth      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;
OK
Time taken: 0.064 seconds
hive> add file /home/ec2-user/date_mapper.py
    > ;
Added resources: [/home/ec2-user/date_mapper.py]
hive>
```

Hive transformation population of dwdate_new table:

Hive Transformation Code:

insert overwrite table dwdate_new 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 date_mapper.py'
as (d_datekey, d_date, d_dayofweek, d_month, d_year, d_yearmonthnum, d_month_abbrev, d_year_new, d_daynuminweek, d_daynuminmonth, d_daynuminyear, d_monthnuminyear, d_weeknuminyear, d_sellingseason, d_lastdayinweekfl, d_lastdayinmonthfl, d_holidayfl, d_weekdayfl) from dwdate;

Hive Transformation Execution:

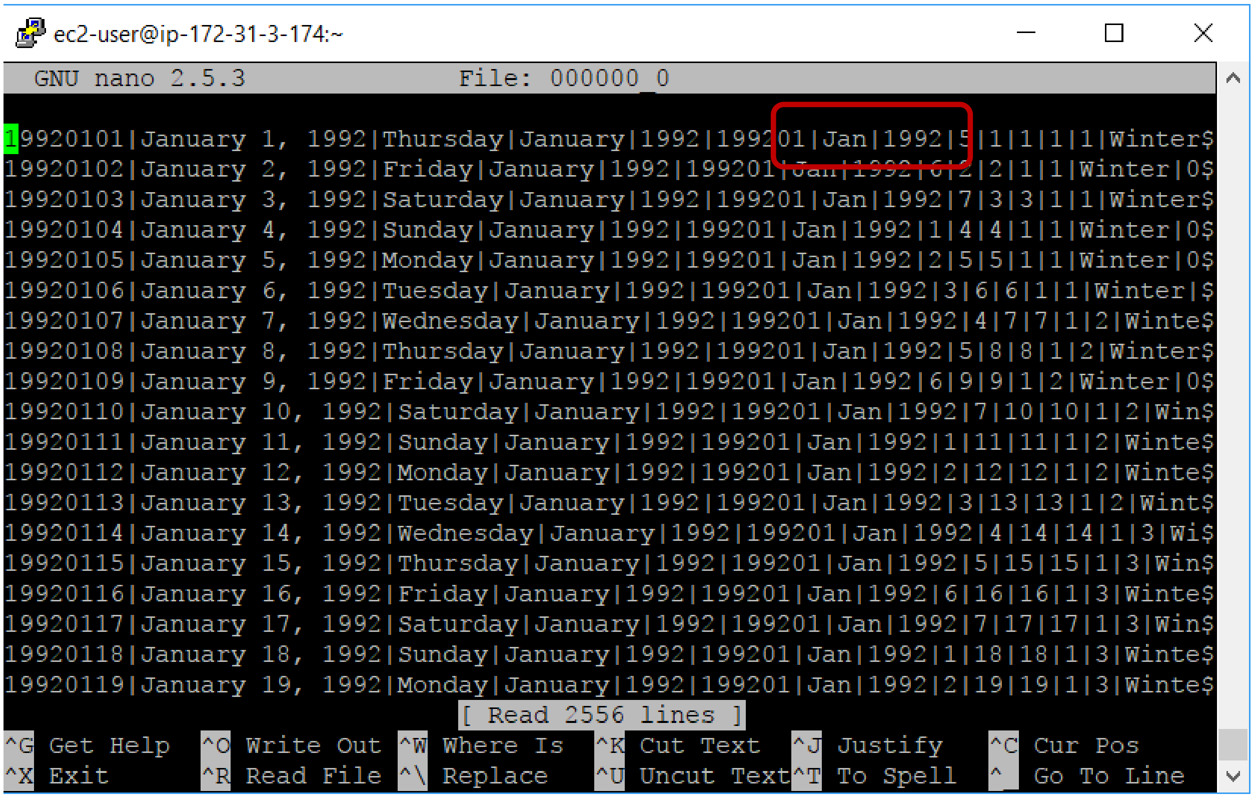


```
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1518559146344_0004, Tracking URL = http://ip-172-31-3-174.us-
east-2.compute.internal:8088/proxy/application_1518559146344_0004/
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job  -kill job_15185591463
44_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2018-02-13 22:19:10,586 Stage-1 map = 0%,  reduce = 0%
2018-02-13 22:19:16,829 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.85 se
c
MapReduce Total cumulative CPU time: 2 seconds 850 msec
Ended Job = job_1518559146344_0004
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://172.31.3.174/user/hive/warehouse/dwdate_new/.hive-staging
_hive_2018-02-13_22-19-05_621_4840094975465470213-1/-ext-10000
Loading data to table default.dwdate_new
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 2.85 sec   HDFS Read: 241200 HDFS Write:
 230045 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 850 msec
OK
Time taken: 13.602 seconds
hive>
```

New dwdate_new table stored in Hive:

Copied the Hive dwdate_new table from HDFS to Linux home directory with the following command run from the Linux /home/ec2-user/apache-hive-2.0.1-bin/ directory:
hadoop fs -get /user/hive/warehouse/dwdate_new/000000_0 /home/ec2-user/

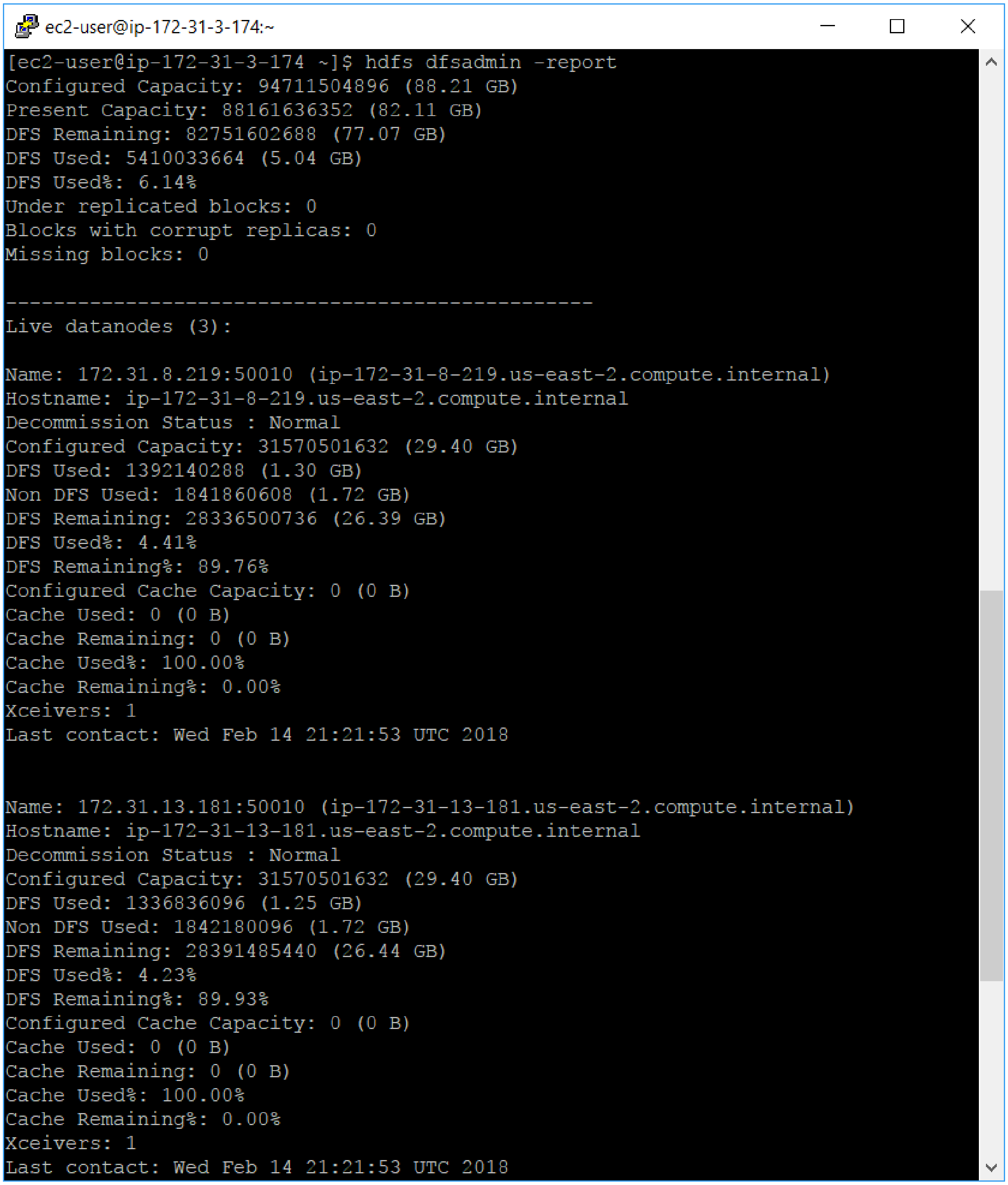
Viewed the copied 000000_0 in Nano with 000000_0. Note that the original 17th column has been separated into 2 different columns.



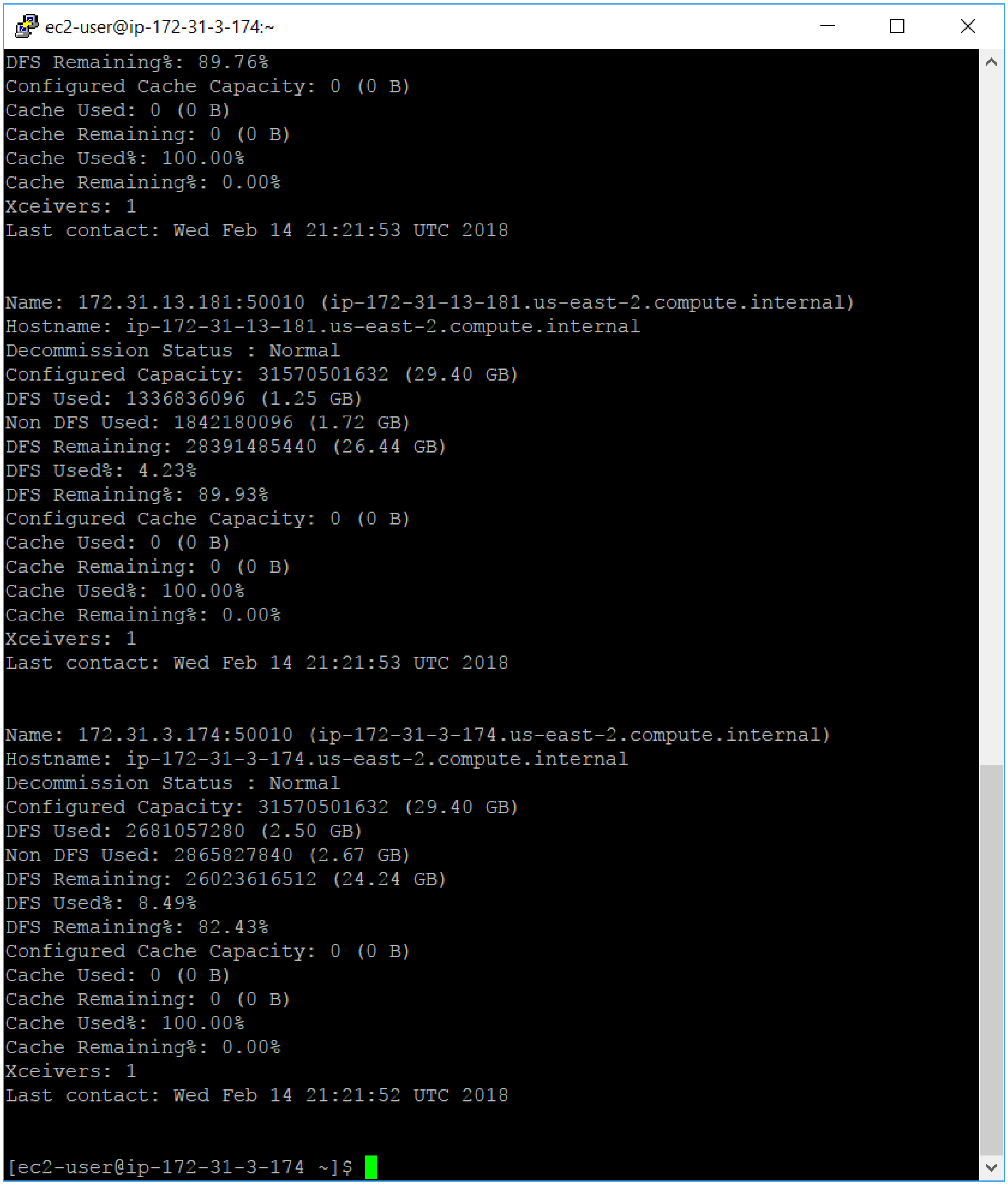
```
  GNU nano 2.5.3              File: 000000_0

19920101|January 1, 1992|Thursday|January|1992|199201|Jan|1992|5|1|1|1|1|Winter$
19920102|January 2, 1992|Friday|January|1992|199201|Jan|1992|6|2|2|1|1|Winter|0$
19920103|January 3, 1992|Saturday|January|1992|199201|Jan|1992|7|3|3|1|1|Winter$
19920104|January 4, 1992|Sunday|January|1992|199201|Jan|1992|1|4|4|1|1|Winter|0$
19920105|January 5, 1992|Monday|January|1992|199201|Jan|1992|2|5|5|1|1|Winter|0$
19920106|January 6, 1992|Tuesday|January|1992|199201|Jan|1992|3|6|6|1|1|Winter|$
19920107|January 7, 1992|Wednesday|January|1992|199201|Jan|1992|4|7|7|1|2|Winte$
19920108|January 8, 1992|Thursday|January|1992|199201|Jan|1992|5|8|8|1|2|Winter$
19920109|January 9, 1992|Friday|January|1992|199201|Jan|1992|6|9|9|1|2|Winter|0$
19920110|January 10, 1992|Saturday|January|1992|199201|Jan|1992|7|10|10|1|2|Win$
19920111|January 11, 1992|Sunday|January|1992|199201|Jan|1992|1|11|11|1|2|Winte$
19920112|January 12, 1992|Monday|January|1992|199201|Jan|1992|2|12|12|1|2|Winte$
19920113|January 13, 1992|Tuesday|January|1992|199201|Jan|1992|3|13|13|1|2|Wint$
19920114|January 14, 1992|Wednesday|January|1992|199201|Jan|1992|4|14|14|1|3|Wi$
19920115|January 15, 1992|Thursday|January|1992|199201|Jan|1992|5|15|15|1|3|Win$
19920116|January 16, 1992|Friday|January|1992|199201|Jan|1992|6|16|16|1|3|Winte$
19920117|January 17, 1992|Saturday|January|1992|199201|Jan|1992|7|17|17|1|3|Win$
19920118|January 18, 1992|Sunday|January|1992|199201|Jan|1992|1|18|18|1|3|Winte$
19920119|January 19, 1992|Monday|January|1992|199201|Jan|1992|2|19|19|1|3|Winte$
                                   [ Read 2556 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

Part 3 – Pig

HDFS Storage Remaining



```
[ec2-user@ip-172-31-3-174 ~]$ hdfs dfsadmin -report
Configured Capacity: 94711504896 (88.21 GB)
Present Capacity: 88161636352 (82.11 GB)
DFS Remaining: 82751602688 (77.07 GB)
DFS Used: 5410033664 (5.04 GB)
DFS Used%: 6.14%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0

-------------------------------------------------
Live datanodes (3):

Name: 172.31.8.219:50010 (ip-172-31-8-219.us-east-2.compute.internal)
Hostname: ip-172-31-8-219.us-east-2.compute.internal
Decommission Status : Normal
Configured Capacity: 31570501632 (29.40 GB)
DFS Used: 1392140288 (1.30 GB)
Non DFS Used: 1841860608 (1.72 GB)
DFS Remaining: 28336500736 (26.39 GB)
DFS Used%: 4.41%
DFS Remaining%: 89.76%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Wed Feb 14 21:21:53 UTC 2018


Name: 172.31.13.181:50010 (ip-172-31-13-181.us-east-2.compute.internal)
Hostname: ip-172-31-13-181.us-east-2.compute.internal
Decommission Status : Normal
Configured Capacity: 31570501632 (29.40 GB)
DFS Used: 1336836096 (1.25 GB)
Non DFS Used: 1842180096 (1.72 GB)
DFS Remaining: 28391485440 (26.44 GB)
DFS Used%: 4.23%
DFS Remaining%: 89.93%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Wed Feb 14 21:21:53 UTC 2018
```



```
DFS Remaining%: 89.76%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Wed Feb 14 21:21:53 UTC 2018


Name: 172.31.13.181:50010 (ip-172-31-13-181.us-east-2.compute.internal)
Hostname: ip-172-31-13-181.us-east-2.compute.internal
Decommission Status : Normal
Configured Capacity: 31570501632 (29.40 GB)
DFS Used: 1336836096 (1.25 GB)
Non DFS Used: 1842180096 (1.72 GB)
DFS Remaining: 28391485440 (26.44 GB)
DFS Used%: 4.23%
DFS Remaining%: 89.93%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Wed Feb 14 21:21:53 UTC 2018


Name: 172.31.3.174:50010 (ip-172-31-3-174.us-east-2.compute.internal)
Hostname: ip-172-31-3-174.us-east-2.compute.internal
Decommission Status : Normal
Configured Capacity: 31570501632 (29.40 GB)
DFS Used: 2681057280 (2.50 GB)
Non DFS Used: 2865827840 (2.67 GB)
DFS Remaining: 26023616512 (24.24 GB)
DFS Used%: 8.49%
DFS Remaining%: 82.43%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Wed Feb 14 21:21:52 UTC 2018


[ec2-user@ip-172-31-3-174 ~]$
```

There is 89.93% of DFS storage free (only 4.23% is used).

Query 0.1 Code and Execution

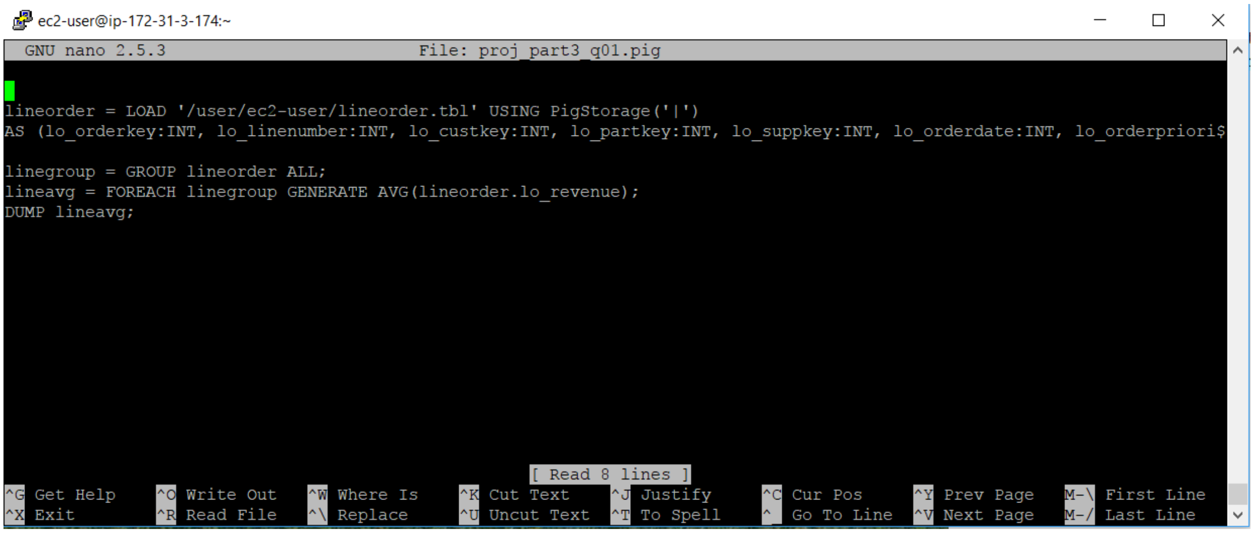
Pig Query Code:

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);

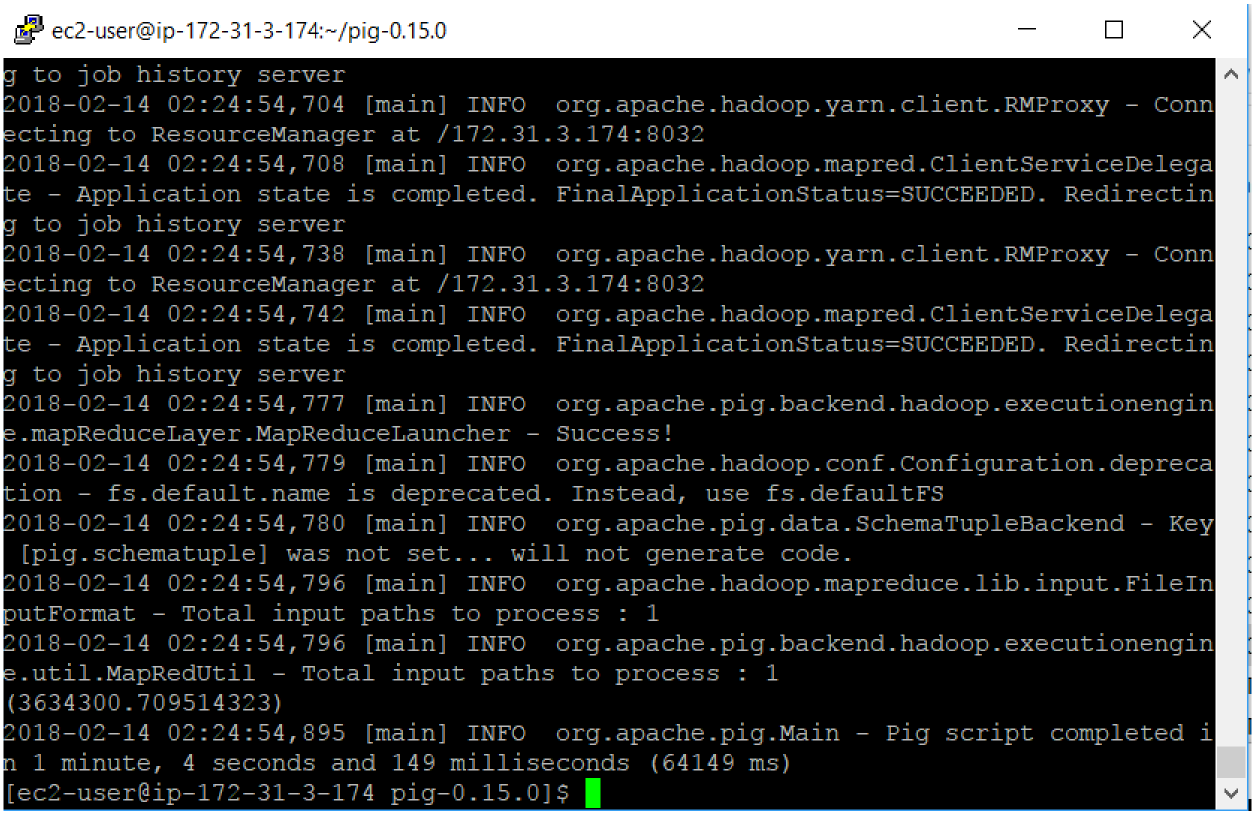
linegroup = GROUP lineorder ALL;
lineavg = FOREACH linegroup GENERATE AVG(lineorder.lo_revenue);
DUMP lineavg;

Pig Query Execution:

I created a script file named proj_part3_q01.pig containing the pig query code. I then ran this script using this command to get the time the pig command took to run (run from the /home/user-ec2-user/pig-0.15.0/ directory): bin/pit -f proj_part3_q01.pig



```
  GNU nano 2.5.3                      File: proj_part3_q01.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, lo_orderdate:INT, lo_orderpriori$

linegroup = GROUP lineorder ALL;
lineavg = FOREACH linegroup GENERATE AVG(lineorder.lo_revenue);
DUMP lineavg;









                                          [ Read 8 lines ]
```



```
g to job history server
2018-02-14 02:24:54,704 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Conn
ecting to ResourceManager at /172.31.3.174:8032
2018-02-14 02:24:54,708 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelega
te - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirectin
g to job history server
2018-02-14 02:24:54,738 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Conn
ecting to ResourceManager at /172.31.3.174:8032
2018-02-14 02:24:54,742 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelega
te - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirectin
g to job history server
2018-02-14 02:24:54,777 [main] INFO  org.apache.pig.backend.hadoop.executionengin
e.mapReduceLayer.MapReduceLauncher - Success!
2018-02-14 02:24:54,779 [main] INFO  org.apache.hadoop.conf.Configuration.depreca
tion - fs.default.name is deprecated. Instead, use fs.defaultFS
2018-02-14 02:24:54,780 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key
 [pig.schematuple] was not set... will not generate code.
2018-02-14 02:24:54,796 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileIn
putFormat - Total input paths to process : 1
2018-02-14 02:24:54,796 [main] INFO  org.apache.pig.backend.hadoop.executionengin
e.util.MapRedUtil - Total input paths to process : 1
(3634300.709514323)
2018-02-14 02:24:54,895 [main] INFO  org.apache.pig.Main - Pig script completed i
n 1 minute, 4 seconds and 149 milliseconds (64149 ms)
[ec2-user@ip-172-31-3-174 pig-0.15.0]$
```

The average lo_revenue returned is 3634300.7095. The query took 1 min, 4 sec, and 149 msec to run.

Query 0.2 Code and Execution

Pig Query Code:

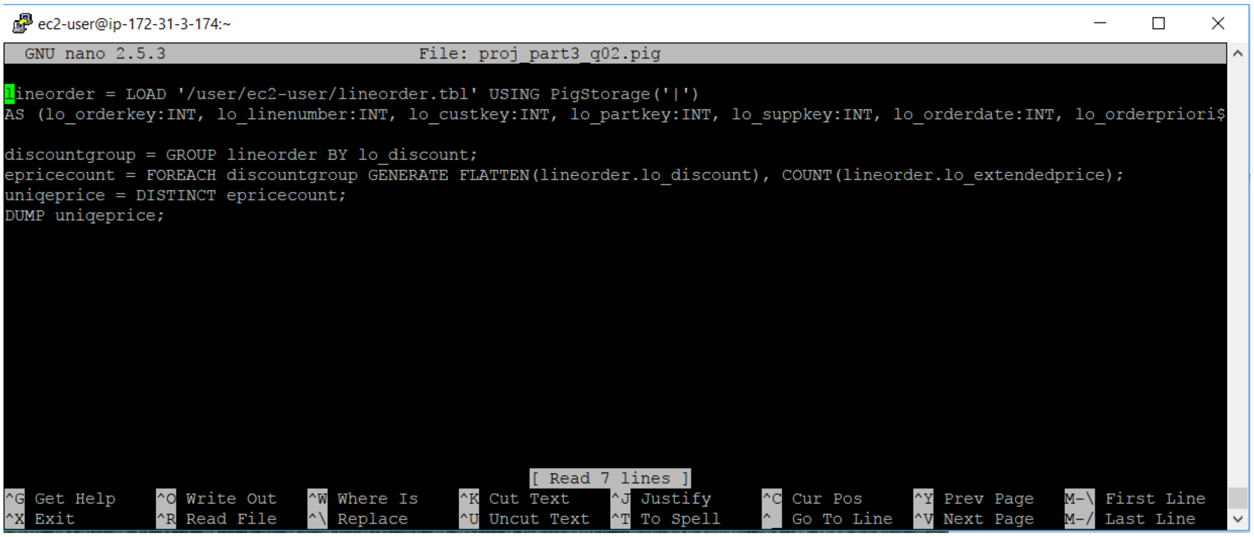
lineorder = LOAD '/user/ec2-user/lineorder.tbl' USING PigStorage('|')
AS (lo_orderkey:INT, lo_linenumber:INT, lo_custkey:INT, lo_partkey:INT, lo_suppkey:INT, lo_orderdate:INT, lo_orderpriority:CHARARRAY, lo_shippriority:CHARARRAY, lo_quantity:INT, lo_extendedprice:INT, lo_ordertotalprice:INT, lo_discount:INT,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

discountgroup = GROUP lineorder BY lo_discount;
epricecount = FOREACH discountgroup GENERATE FLATTEN(lineorder.lo_discount), COUNT(lineorder.lo_extendedprice);
uniqeprice = DISTINCT epricecount;
DUMP uniqeprice;

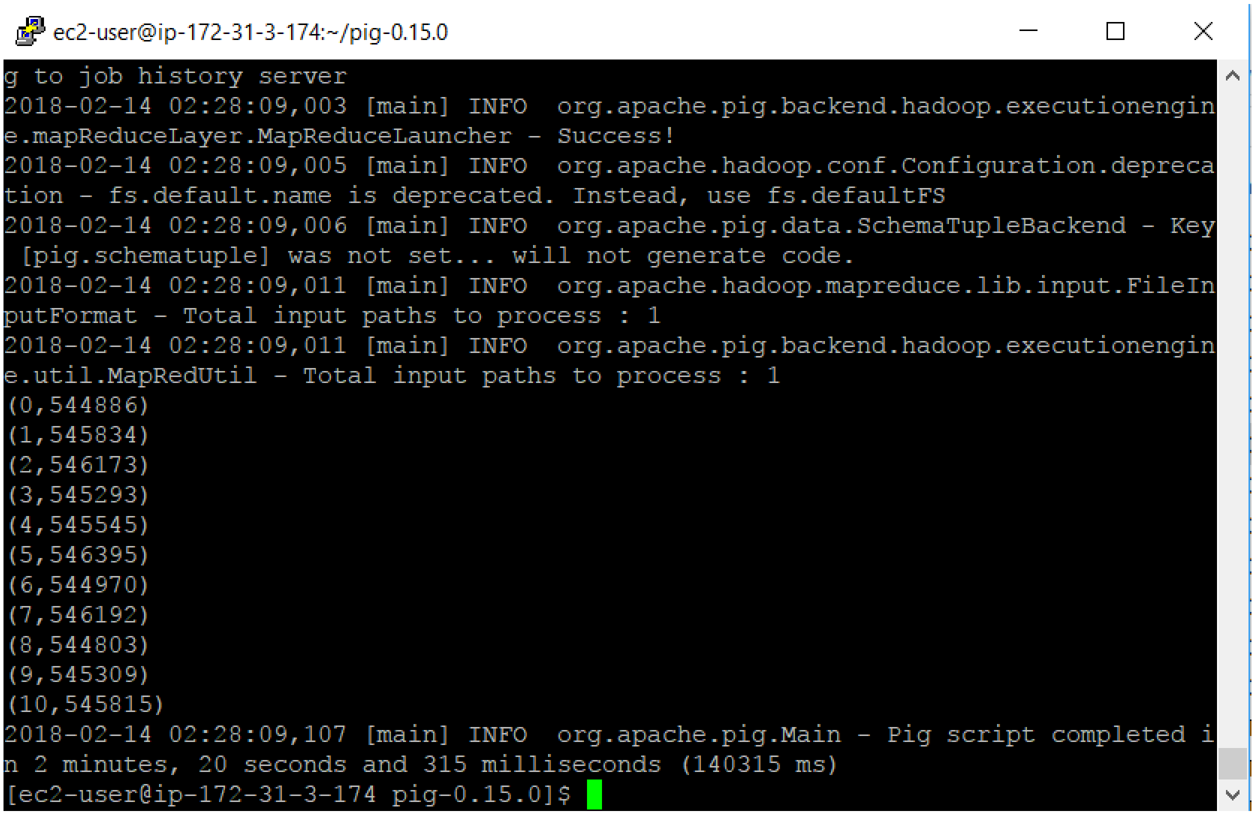
Note that I added the FLATTEN to the FOREACH/GENERATE command because the command originally returned a dictionary type entry per group with the key being a list the number of elements in each group, each element containing the lo_discount value for the group and the value being the lo_extendedprice for the group. The flatten changed this so that each group returned a number of lines equal to the number of elements in the group, each line containing the lo_discount value and the lo_extendedprice of the group. Every line of the group contained the same values. This resulted in several repeated lines per group. To get the output to be just one line per group, I took the DISTINCT of the FOREACH output, which then only returned distinct lines (one for each group).

Pig Query Execution:

I created a script file named proj_part3_q02.pig containing the pig query code. I then ran this script using this command to get the time the pig command took to run (run from the /home/user-ec2-user/pig-0.15.0/ directory): bin/pit -f proj_part3_q02.pig



```
  GNU nano 2.5.3                      File: proj_part3_q02.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, lo_orderdate:INT, lo_orderpriori$

discountgroup = GROUP lineorder BY lo_discount;
epricecount = FOREACH discountgroup GENERATE FLATTEN(lineorder.lo_discount), COUNT(lineorder.lo_extendedprice);
uniqeprice = DISTINCT epricecount;
DUMP uniqeprice;

                                          [ Read 7 lines ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     ^Y Prev Page   M-\ First Line
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^  Go To Line  ^V Next Page   M-/ Last Line
```



```
g to job history server
2018-02-14 02:28:09,003 [main] INFO  org.apache.pig.backend.hadoop.executionengin
e.mapReduceLayer.MapReduceLauncher - Success!
2018-02-14 02:28:09,005 [main] INFO  org.apache.hadoop.conf.Configuration.depreca
tion - fs.default.name is deprecated. Instead, use fs.defaultFS
2018-02-14 02:28:09,006 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key
 [pig.schematuple] was not set... will not generate code.
2018-02-14 02:28:09,011 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileIn
putFormat - Total input paths to process : 1
2018-02-14 02:28:09,011 [main] INFO  org.apache.pig.backend.hadoop.executionengin
e.util.MapRedUtil - Total input paths to process : 1
(0,544886)
(1,545834)
(2,546173)
(3,545293)
(4,545545)
(5,546395)
(6,544970)
(7,546192)
(8,544803)
(9,545309)
(10,545815)
2018-02-14 02:28:09,107 [main] INFO  org.apache.pig.Main - Pig script completed i
n 2 minutes, 20 seconds and 315 milliseconds (140315 ms)
[ec2-user@ip-172-31-3-174 pig-0.15.0]$
```

The query took 2 min, 20 sec, and 315 msec to run.

Query 0.3 Code and Execution

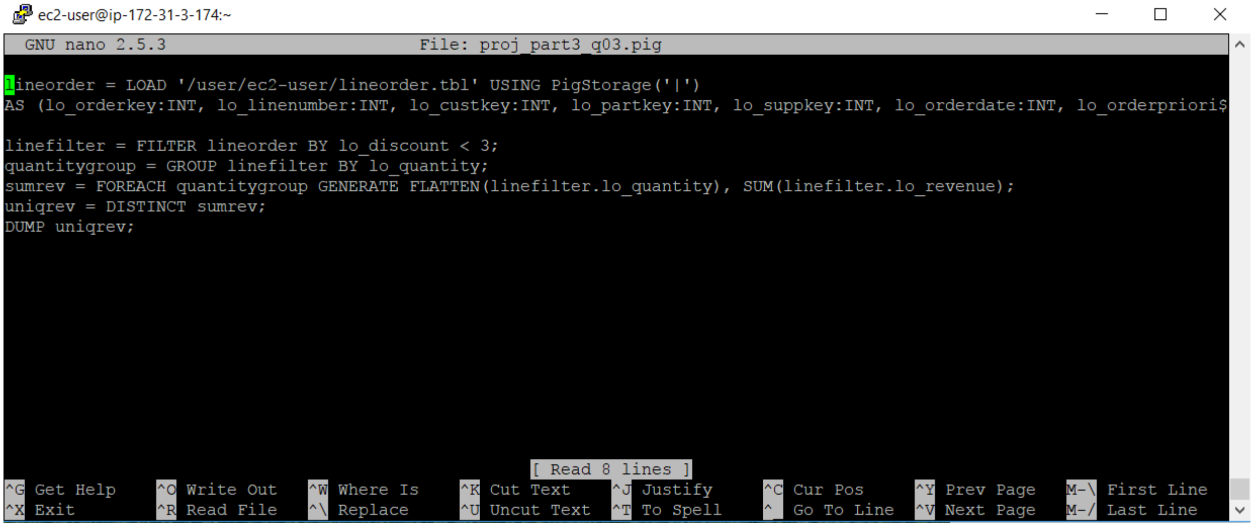
Pig Query Code:

```
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);

linefilter = FILTER lineorder BY lo_discount < 3;
quantitygroup = GROUP linefilter BY lo_quantity;
sumrev = FOREACH quantitygroup GENERATE FLATTEN(linefilter.lo_quantity),
SUM(linefilter.lo_revenue);
uniqrev = DISTINCT sumrev;
DUMP uniqrev;
```

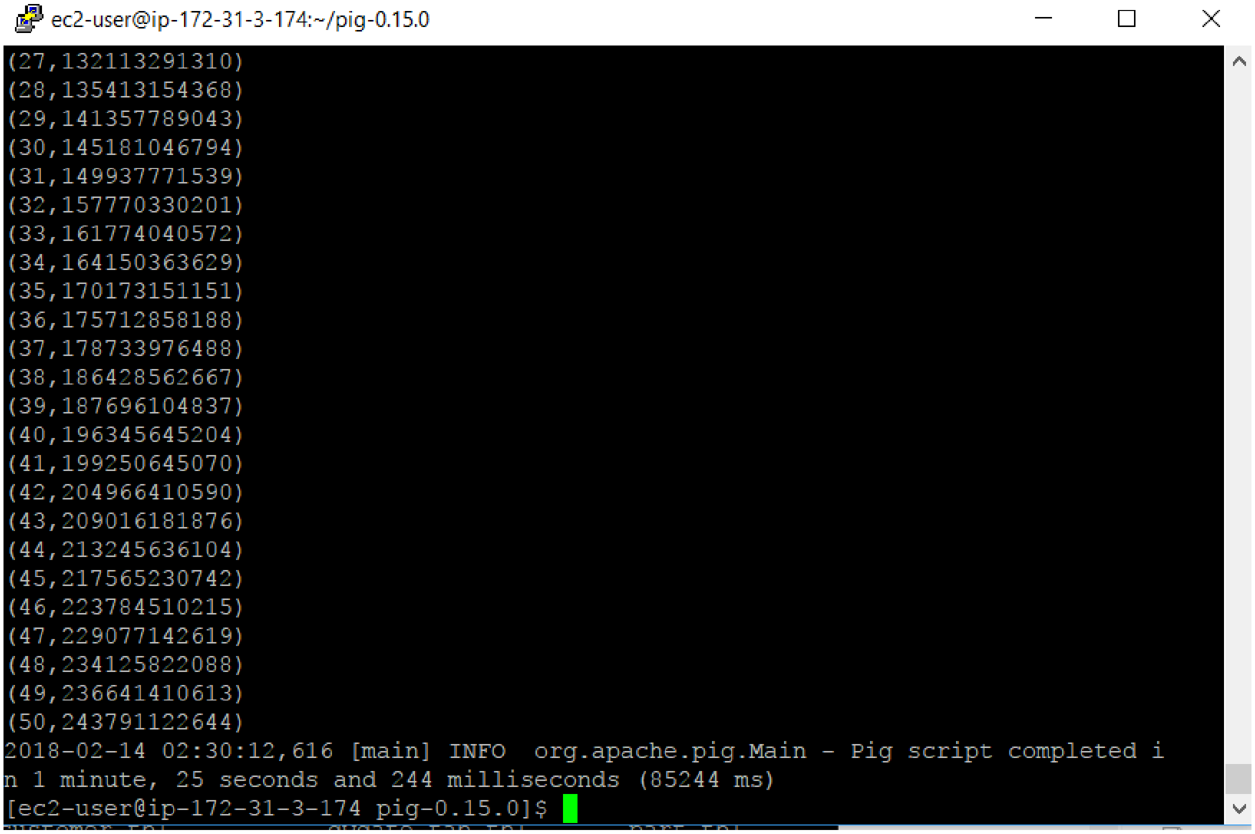
Note that I added the FLATTEN to the FOREACH/GENERATE command because the command originally returned a dictionary type entry per group with the key being a list the number of elements in each group, each element containing the lo_quantity value for the group and the value being the lo_revenue for the group. The flatten changed this so that each group returned a number of lines equal to the number of elements in the group, each line containing the lo_quantity value and the lo_revenue of the group. Every line of the group contained the same values. This resulted in several repeated lines per group. To get the output to be just one line per group, I took the DISTINCT of the FOREACH output, which then only returned distinct lines (one for each group).

Pig Query Execution:

I created a script file named proj_part3_q03.pig containing the pig query code. I then ran this script using this command to get the time the pig command took to run (run from the /home/user-ec2-user/pig-0.15.0/ directory): bin/pit -f proj_part3_q03.pig



```
  GNU nano 2.5.3                         File: proj_part3_q03.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, lo_orderdate:INT, lo_orderpriori$

linefilter = FILTER lineorder BY lo_discount < 3;
quantitygroup = GROUP linefilter BY lo_quantity;
sumrev = FOREACH quantitygroup GENERATE FLATTEN(linefilter.lo_quantity), SUM(linefilter.lo_revenue);
uniqrev = DISTINCT sumrev;
DUMP uniqrev;

                                             [ Read 8 lines ]
^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos      ^Y Prev Page    M-\ First Line
^X Exit         ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell     ^  Go To Line   ^V Next Page    M-/ Last Line
```



```
(27,132113291310)
(28,135413154368)
(29,141357789043)
(30,145181046794)
(31,149937771539)
(32,157770330201)
(33,161774040572)
(34,164150363629)
(35,170173151151)
(36,175712858188)
(37,178733976488)
(38,186428562667)
(39,187696104837)
(40,196345645204)
(41,199250645070)
(42,204966410590)
(43,209016181876)
(44,213245636104)
(45,217565230742)
(46,223784510215)
(47,229077142619)
(48,234125822088)
(49,236641410613)
(50,243791122644)
2018-02-14 02:30:12,616 [main] INFO  org.apache.pig.Main - Pig script completed i
n 1 minute, 25 seconds and 244 milliseconds (85244 ms)
[ec2-user@ip-172-31-3-174 pig-0.15.0]$
```

The query took 1 min, 25 sec, and 244 msec to run.

Part 4 – Hadoop Streaming

Query 0.3 Implemented

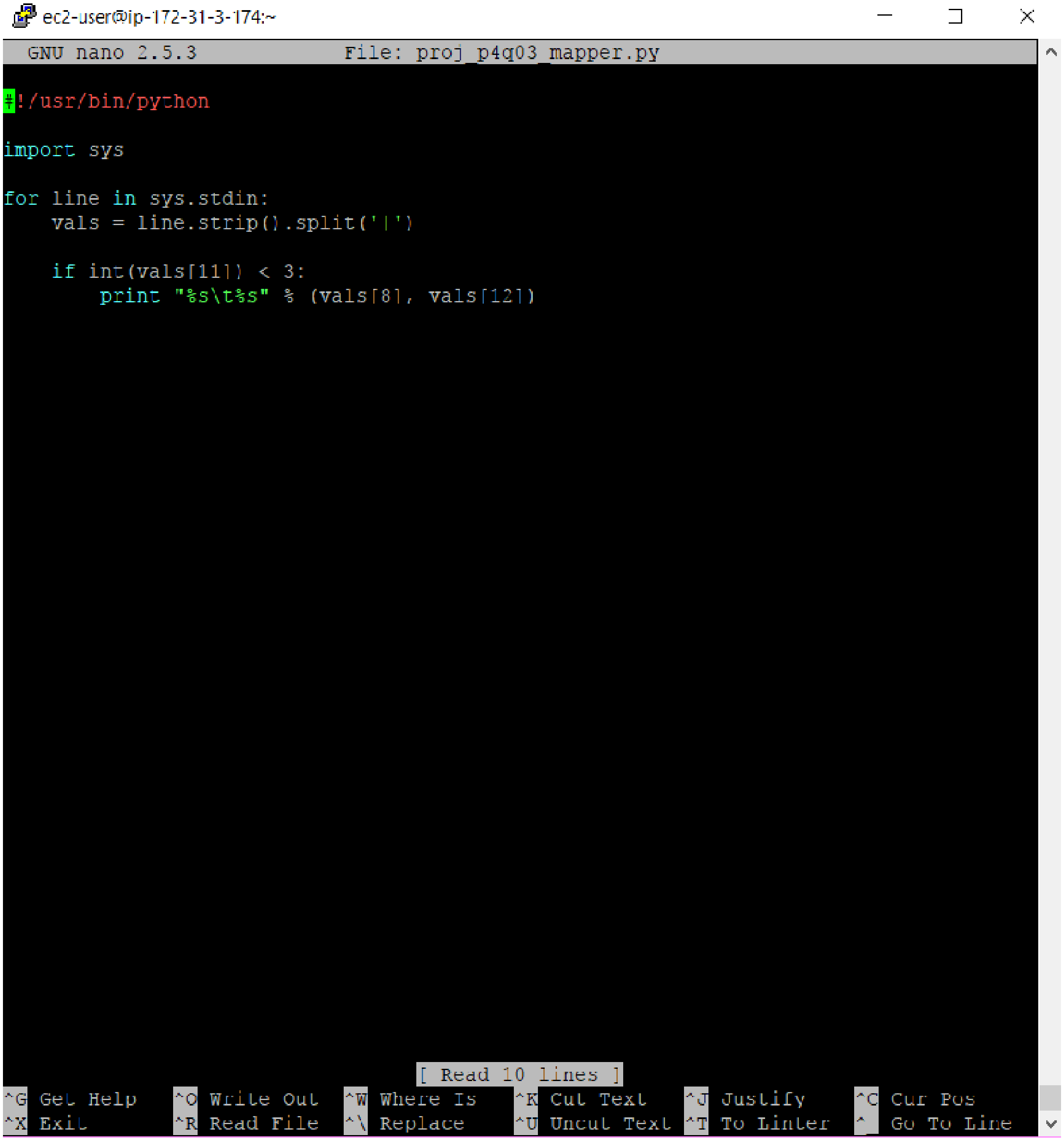
```
SELECT lo_quantity, SUM(lo_revenue)
FROM lineorder
WHERE lo_discount < 3
GROUP BY lo_quantity;
```

Python Mapper

Mapper functionality:

Read in lines of lineorder.tbl that are separated by |. For each line, strip whitespace and split by |. If lo_discount (12th column, python index 11), set key to lo_quantity (9th column, python index 8) and the value to lo_revenue (13th column, python index 12), then print key value pair separated by tab.

Mapper Code:



ec2-user@ip-172-31-3-174:~
GNU nano 2.5.3 File: proj_p4q03_mapper.py

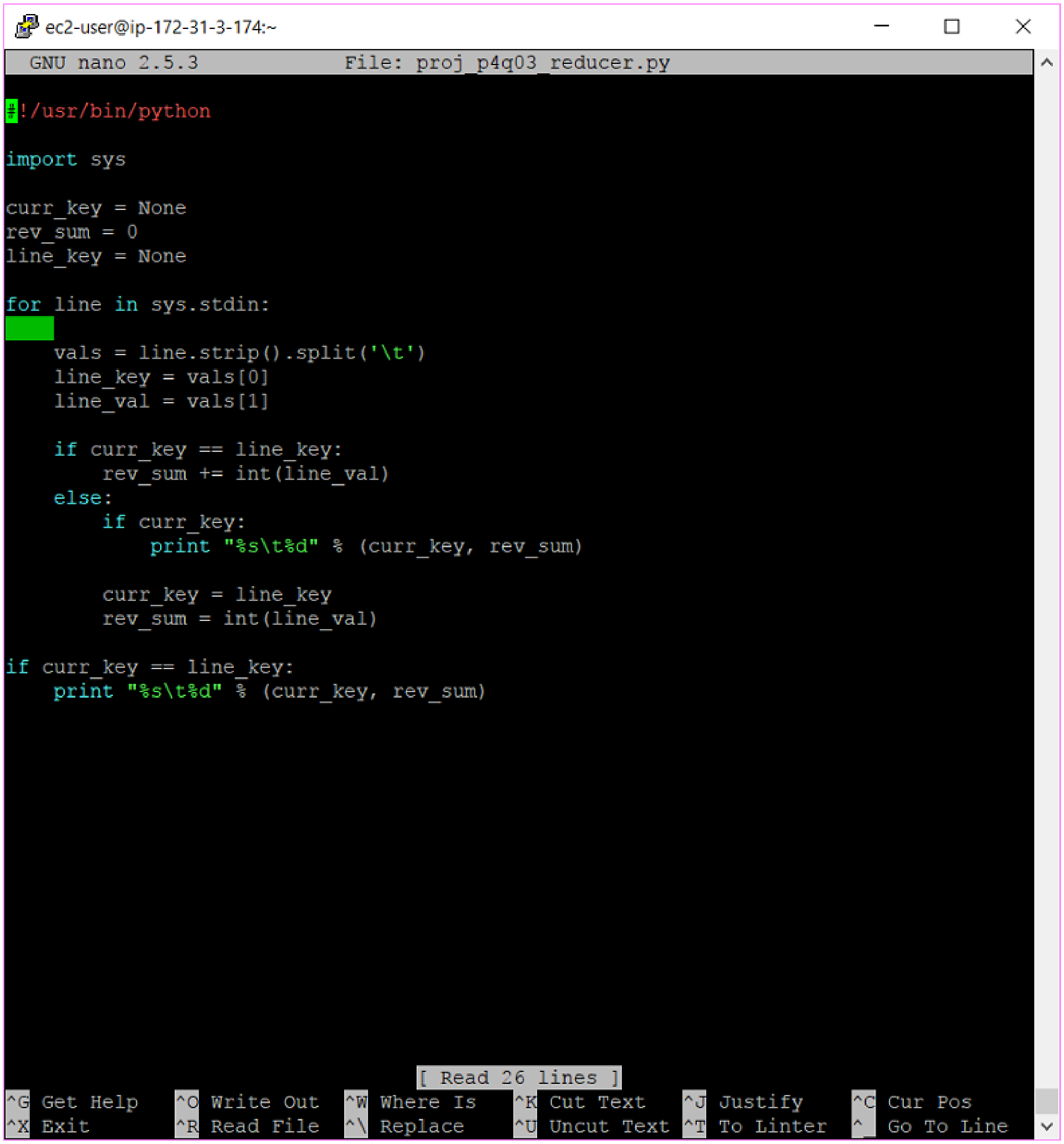
```
#!/usr/bin/python

import sys

for line in sys.stdin:
    vals = line.strip().split('|')

    if int(vals[11]) < 3:
        print "%s\t%s" % (vals[8], vals[12])
```

Python Reducer

Reducer functionality:

Read in lines of Mapper 1 output. Initialize a current key variable to “” before line loop. In line loop, if the current key is different than the key of the current line and if the current key was not “”, print the last key (value of current key) and the value of the accumulator (the sum of lo_revenue). Next, for any cause of current key value, set the current key to the key of the current line and set the accumulator to the current value of the line (the current lo_revenue). If the current key is the same (the else condition), add the line value to the accumulator. After the loop has finished, print out the last current key and accumulator values separated by tab.

Reducer Code:



```
  GNU nano 2.5.3                 File: proj_p4q03_reducer.py

#!/usr/bin/python

import sys

curr_key = None
rev_sum = 0
line_key = None

for line in sys.stdin:

    vals = line.strip().split('\t')
    line_key = vals[0]
    line_val = vals[1]

    if curr_key == line_key:
        rev_sum += int(line_val)
    else:
        if curr_key:
            print "%s\t%d" % (curr_key, rev_sum)

        curr_key = line_key
        rev_sum = int(line_val)

if curr_key == line_key:
    print "%s\t%d" % (curr_key, rev_sum)

                                  [ Read 26 lines ]
```

Hadoop Streaming Execution

Hadoop Streaming Command Executed:

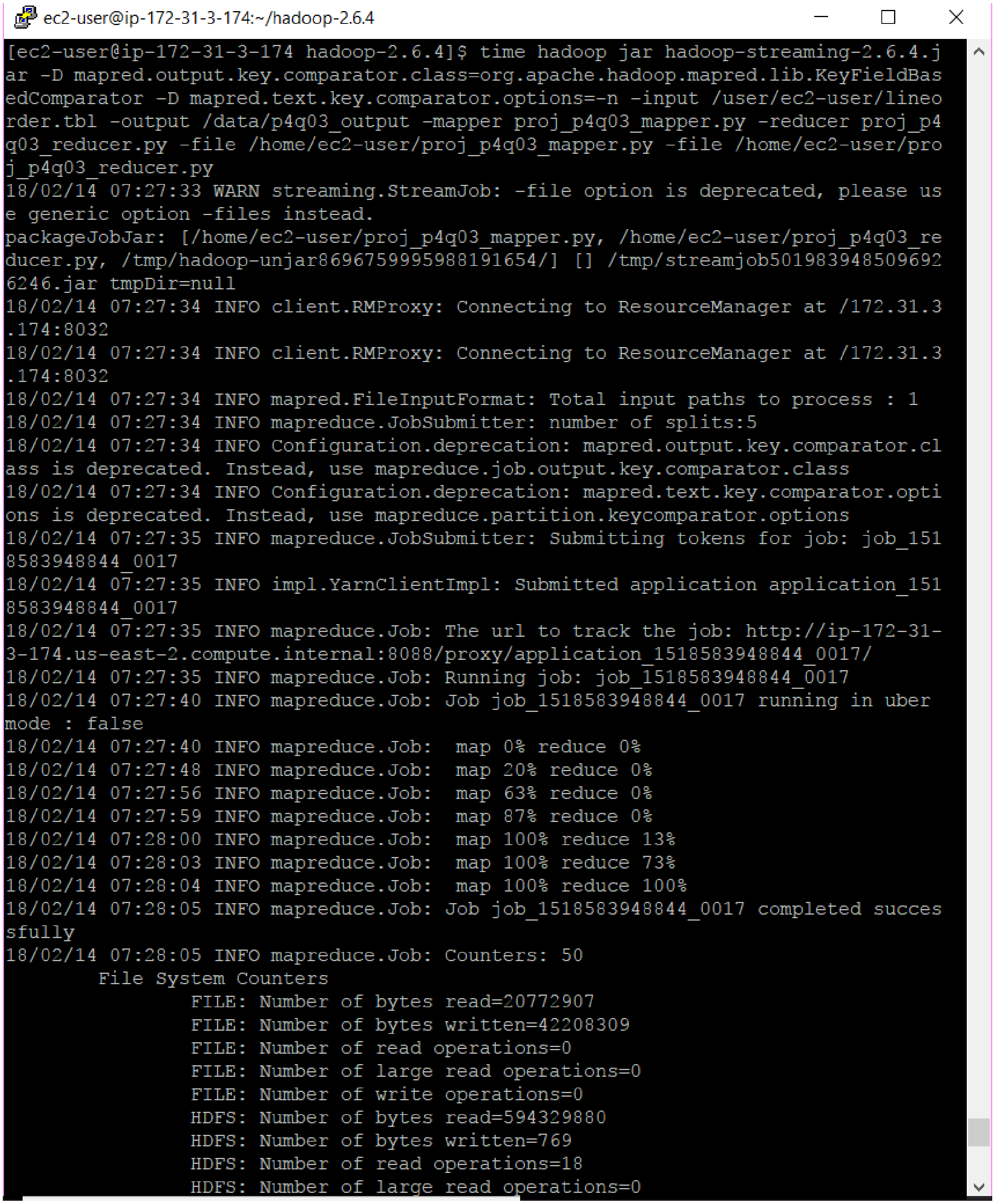
```
time hadoop jar hadoop-streaming-2.6.4.jar -D mapred.output.key.comparator.class=org.apache.hadoop.mapred.lib.KeyFieldBasedComparator -D mapred.text.key.comparator.options=-n -input /user/ec2-user/lineorder.tbl -output /data/p4q03_output -mapper proj_p4q03_mapper.py -reducer proj_p4q03_reducer.py -file /home/ec2-user/proj_p4q03_mapper.py -file /home/ec2-user/proj_p4q03_reducer.py
```

Note: I ran the streaming command with the time command so I would get the amount of time the command took to execute.

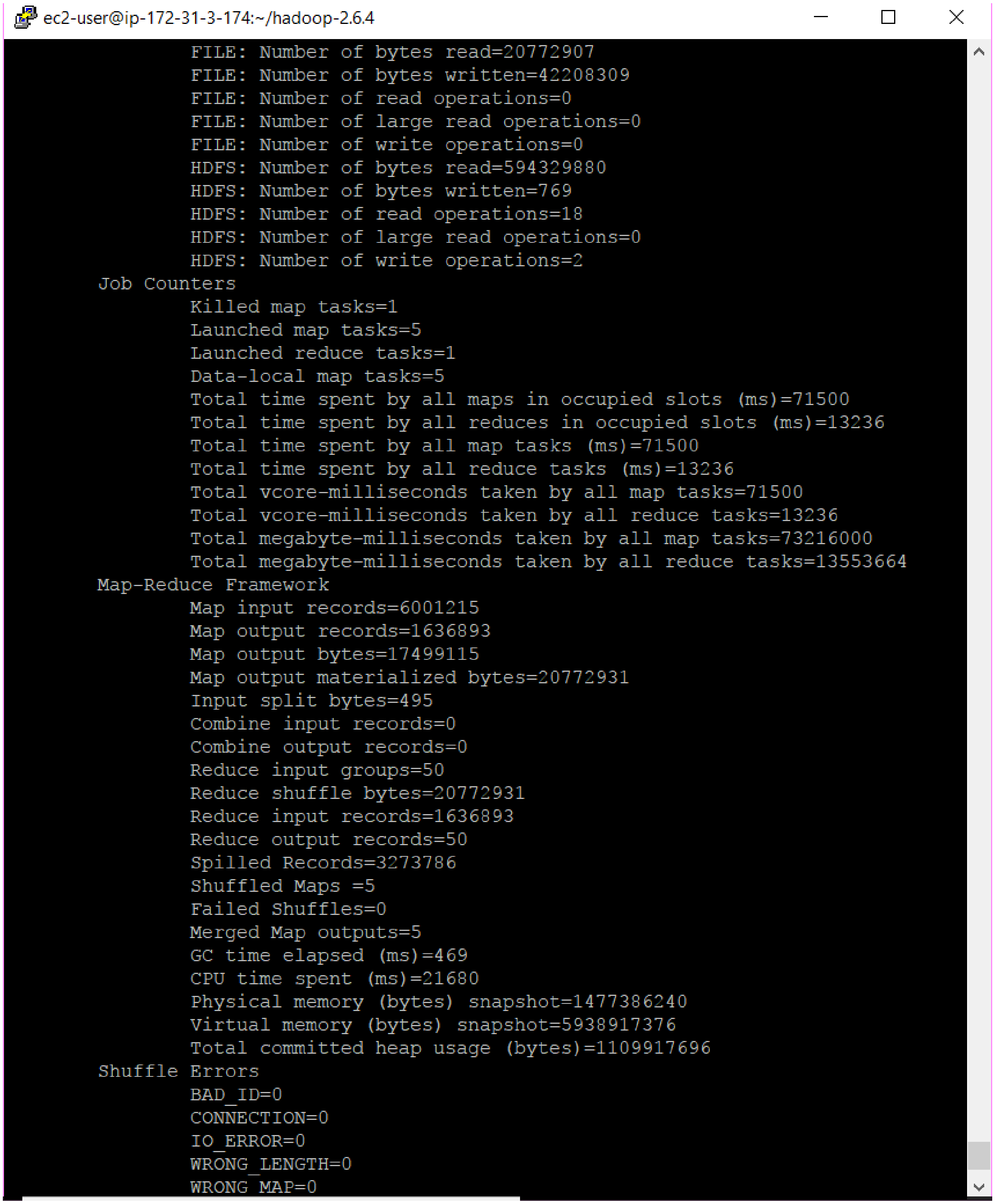
Second note: I included the -D mapred.output.key.comparator.class=org.apache.hadoop.mapred.lib.KeyFieldBasedComparator and -D mapred.text.key.comparator.options=-n options so that the output of the mapper would be interpreted as numeric. This resulted in the reducer output being in numeric order from lowest to highest.

Third note: I had to copy the hadoop-streaming-2.6.4.jar file from the /home/ec2-user/hadoop-2.6.4/share/hadoop/tools/lib/ directory into the /home/ec2-user/hadoop-2.6.4/ directory using the following command (run from the /home/ec2-user/hadoop-2.6.4/ directory):

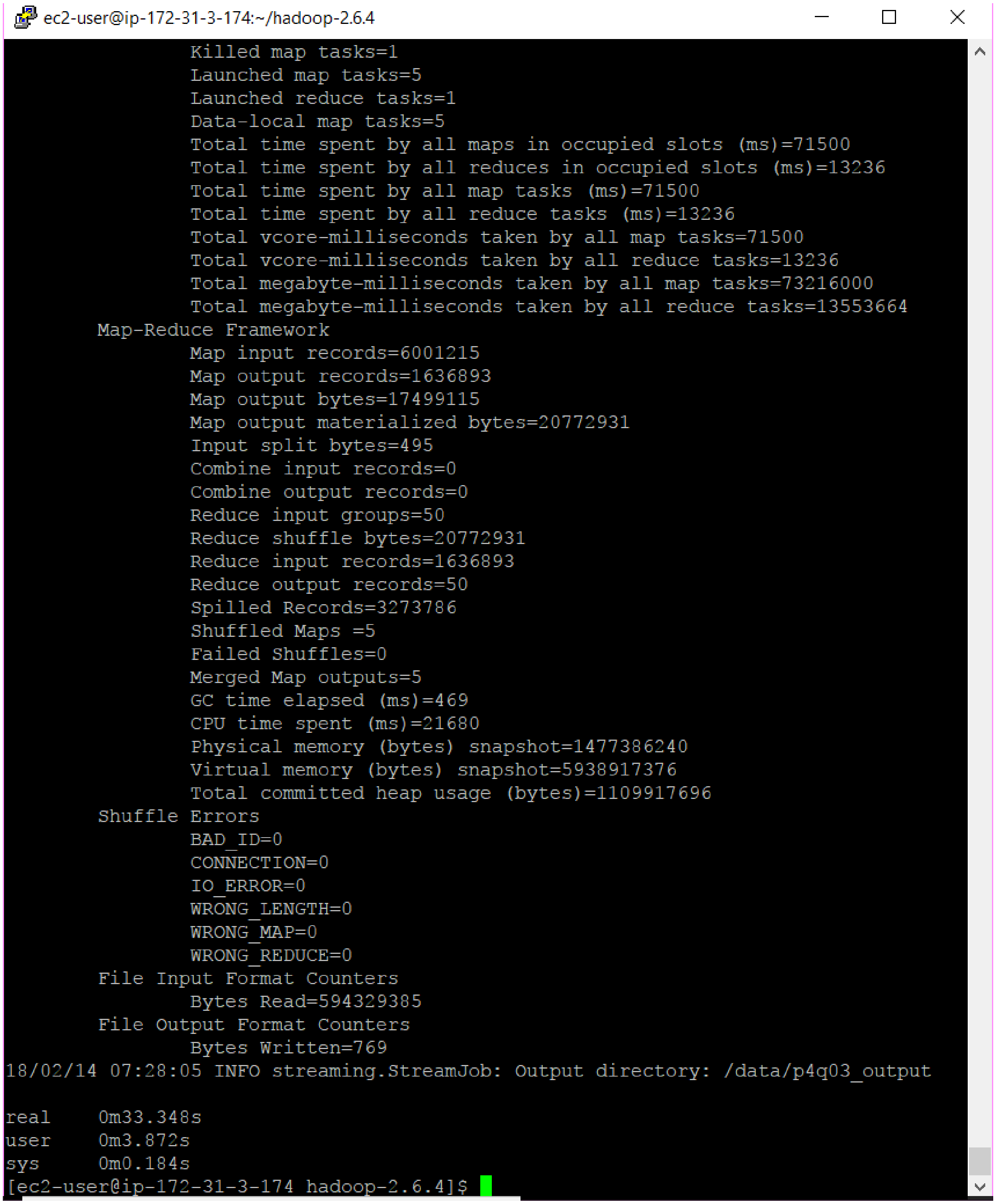
```
cp ./share/hadoop/tools/lib/hadoop-streaming-2.6.4.jar .
```



```
[ec2-user@ip-172-31-3-174 hadoop-2.6.4]$ time hadoop jar hadoop-streaming-2.6.4.j
ar -D mapred.output.key.comparator.class=org.apache.hadoop.mapred.lib.KeyFieldBas
edComparator -D mapred.text.key.comparator.options=-n -input /user/ec2-user/lineo
rder.tbl -output /data/p4q03_output -mapper proj_p4q03_mapper.py -reducer proj_p4
q03_reducer.py -file /home/ec2-user/proj_p4q03_mapper.py -file /home/ec2-user/pro
j_p4q03_reducer.py
18/02/14 07:27:33 WARN streaming.StreamJob: -file option is deprecated, please us
e generic option -files instead.
packageJobJar: [/home/ec2-user/proj_p4q03_mapper.py, /home/ec2-user/proj_p4q03_re
ducer.py, /tmp/hadoop-unjar8696759995988191654/] [] /tmp/streamjob501983948509692
6246.jar tmpDir=null
18/02/14 07:27:34 INFO client.RMProxy: Connecting to ResourceManager at /172.31.3
.174:8032
18/02/14 07:27:34 INFO client.RMProxy: Connecting to ResourceManager at /172.31.3
.174:8032
18/02/14 07:27:34 INFO mapred.FileInputFormat: Total input paths to process : 1
18/02/14 07:27:34 INFO mapreduce.JobSubmitter: number of splits:5
18/02/14 07:27:34 INFO Configuration.deprecation: mapred.output.key.comparator.cl
ass is deprecated. Instead, use mapreduce.job.output.key.comparator.class
18/02/14 07:27:34 INFO Configuration.deprecation: mapred.text.key.comparator.opti
ons is deprecated. Instead, use mapreduce.partition.keycomparator.options
18/02/14 07:27:35 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_151
8583948844_0017
18/02/14 07:27:35 INFO impl.YarnClientImpl: Submitted application application_151
8583948844_0017
18/02/14 07:27:35 INFO mapreduce.Job: The url to track the job: http://ip-172-31-
3-174.us-east-2.compute.internal:8088/proxy/application_1518583948844_0017/
18/02/14 07:27:35 INFO mapreduce.Job: Running job: job_1518583948844_0017
18/02/14 07:27:40 INFO mapreduce.Job: Job job_1518583948844_0017 running in uber
mode : false
18/02/14 07:27:40 INFO mapreduce.Job:  map 0% reduce 0%
18/02/14 07:27:48 INFO mapreduce.Job:  map 20% reduce 0%
18/02/14 07:27:56 INFO mapreduce.Job:  map 63% reduce 0%
18/02/14 07:27:59 INFO mapreduce.Job:  map 87% reduce 0%
18/02/14 07:28:00 INFO mapreduce.Job:  map 100% reduce 13%
18/02/14 07:28:03 INFO mapreduce.Job:  map 100% reduce 73%
18/02/14 07:28:04 INFO mapreduce.Job:  map 100% reduce 100%
18/02/14 07:28:05 INFO mapreduce.Job: Job job_1518583948844_0017 completed succes
sfully
18/02/14 07:28:05 INFO mapreduce.Job: Counters: 50
        File System Counters
                FILE: Number of bytes read=20772907
                FILE: Number of bytes written=42208309
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=594329880
                HDFS: Number of bytes written=769
                HDFS: Number of read operations=18
                HDFS: Number of large read operations=0
```



```
                FILE: Number of bytes read=20772907
                FILE: Number of bytes written=42208309
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=594329880
                HDFS: Number of bytes written=769
                HDFS: Number of read operations=18
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Killed map tasks=1
                Launched map tasks=5
                Launched reduce tasks=1
                Data-local map tasks=5
                Total time spent by all maps in occupied slots (ms)=71500
                Total time spent by all reduces in occupied slots (ms)=13236
                Total time spent by all map tasks (ms)=71500
                Total time spent by all reduce tasks (ms)=13236
                Total vcore-milliseconds taken by all map tasks=71500
                Total vcore-milliseconds taken by all reduce tasks=13236
                Total megabyte-milliseconds taken by all map tasks=73216000
                Total megabyte-milliseconds taken by all reduce tasks=13553664
        Map-Reduce Framework
                Map input records=6001215
                Map output records=1636893
                Map output bytes=17499115
                Map output materialized bytes=20772931
                Input split bytes=495
                Combine input records=0
                Combine output records=0
                Reduce input groups=50
                Reduce shuffle bytes=20772931
                Reduce input records=1636893
                Reduce output records=50
                Spilled Records=3273786
                Shuffled Maps =5
                Failed Shuffles=0
                Merged Map outputs=5
                GC time elapsed (ms)=469
                CPU time spent (ms)=21680
                Physical memory (bytes) snapshot=1477386240
                Virtual memory (bytes) snapshot=5938917376
                Total committed heap usage (bytes)=1109917696
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
```

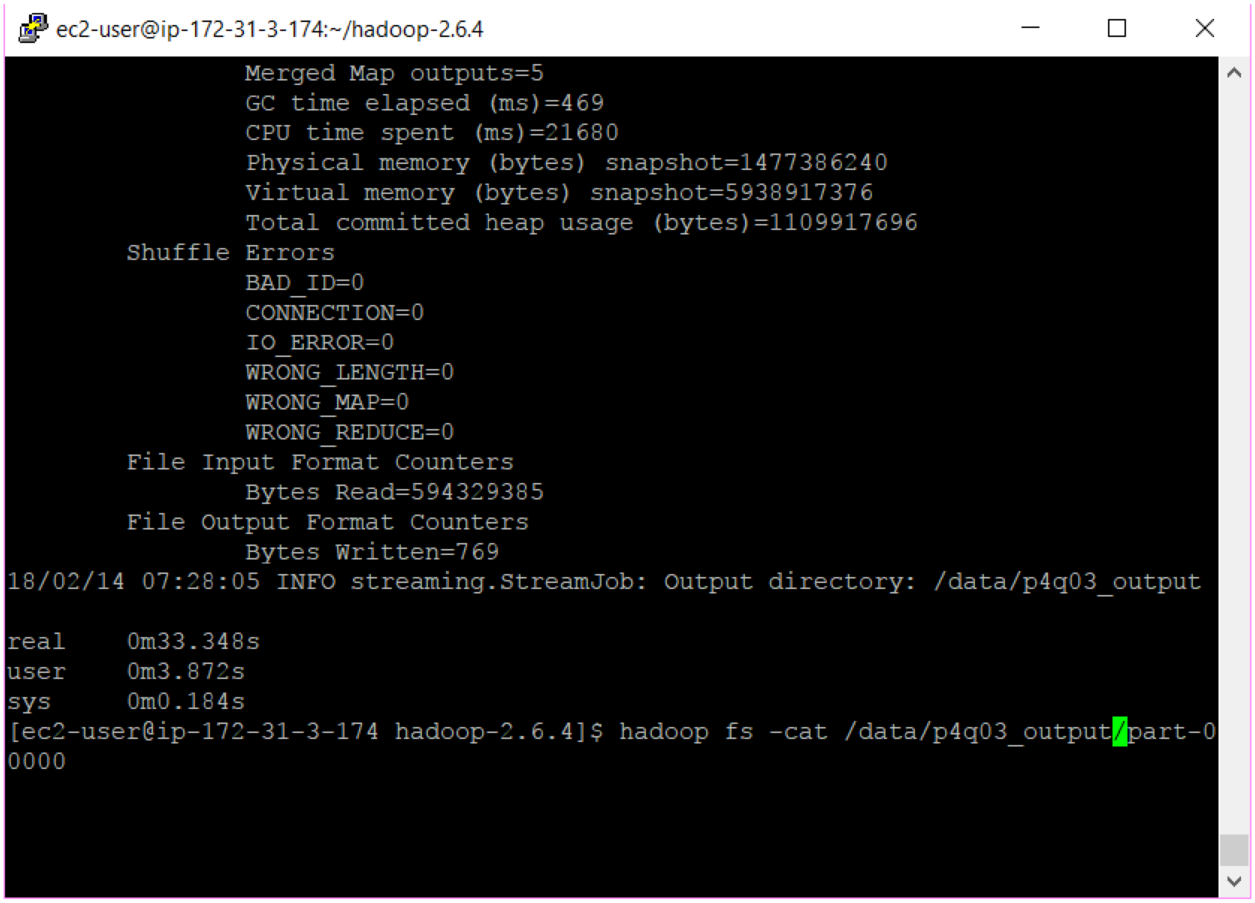


```
                Killed map tasks=1
                Launched map tasks=5
                Launched reduce tasks=1
                Data-local map tasks=5
                Total time spent by all maps in occupied slots (ms)=71500
                Total time spent by all reduces in occupied slots (ms)=13236
                Total time spent by all map tasks (ms)=71500
                Total time spent by all reduce tasks (ms)=13236
                Total vcore-milliseconds taken by all map tasks=71500
                Total vcore-milliseconds taken by all reduce tasks=13236
                Total megabyte-milliseconds taken by all map tasks=73216000
                Total megabyte-milliseconds taken by all reduce tasks=13553664
        Map-Reduce Framework
                Map input records=6001215
                Map output records=1636893
                Map output bytes=17499115
                Map output materialized bytes=20772931
                Input split bytes=495
                Combine input records=0
                Combine output records=0
                Reduce input groups=50
                Reduce shuffle bytes=20772931
                Reduce input records=1636893
                Reduce output records=50
                Spilled Records=3273786
                Shuffled Maps =5
                Failed Shuffles=0
                Merged Map outputs=5
                GC time elapsed (ms)=469
                CPU time spent (ms)=21680
                Physical memory (bytes) snapshot=1477386240
                Virtual memory (bytes) snapshot=5938917376
                Total committed heap usage (bytes)=1109917696
        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=769
18/02/14 07:28:05 INFO streaming.StreamJob: Output directory: /data/p4q03_output

real    0m33.348s
user    0m3.872s
sys     0m0.184s
[ec2-user@ip-172-31-3-174 hadoop-2.6.4]$
```

The streaming command took 33.324 secs to run.

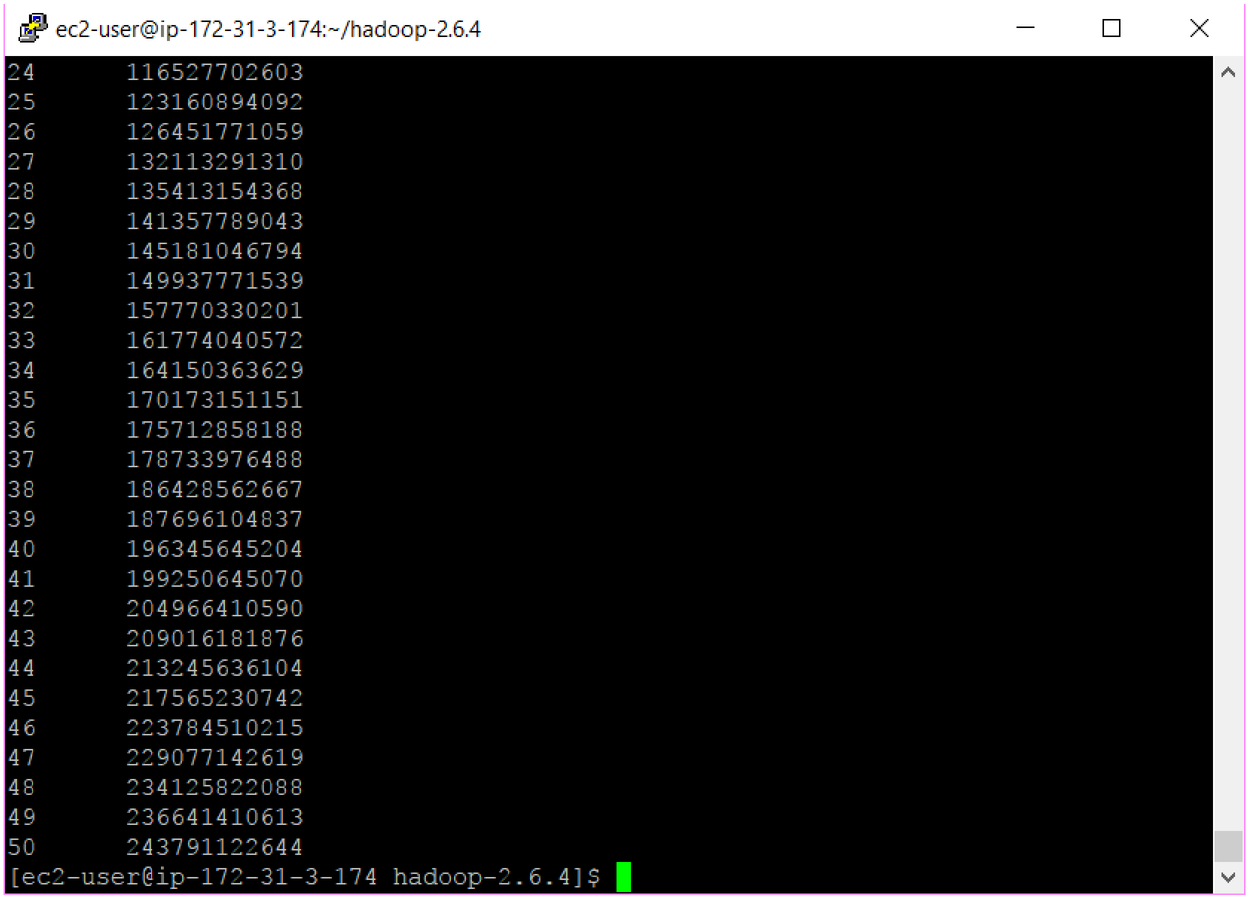
Hadoop Streaming Output

I viewed the result of the Hadoop streaming command with the following command (run from the Linux /home/ec2-user/hadoop-2.6.4/ directory):
hadoop fs -cat /data/p4q03_output/part-00000



```
                Merged Map outputs=5
                GC time elapsed (ms)=469
                CPU time spent (ms)=21680
                Physical memory (bytes) snapshot=1477386240
                Virtual memory (bytes) snapshot=5938917376
                Total committed heap usage (bytes)=1109917696
        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=769
18/02/14 07:28:05 INFO streaming.StreamJob: Output directory: /data/p4q03_output

real    0m33.348s
user    0m3.872s
sys     0m0.184s
[ec2-user@ip-172-31-3-174 hadoop-2.6.4]$ hadoop fs -cat /data/p4q03_output/part-0
0000
```

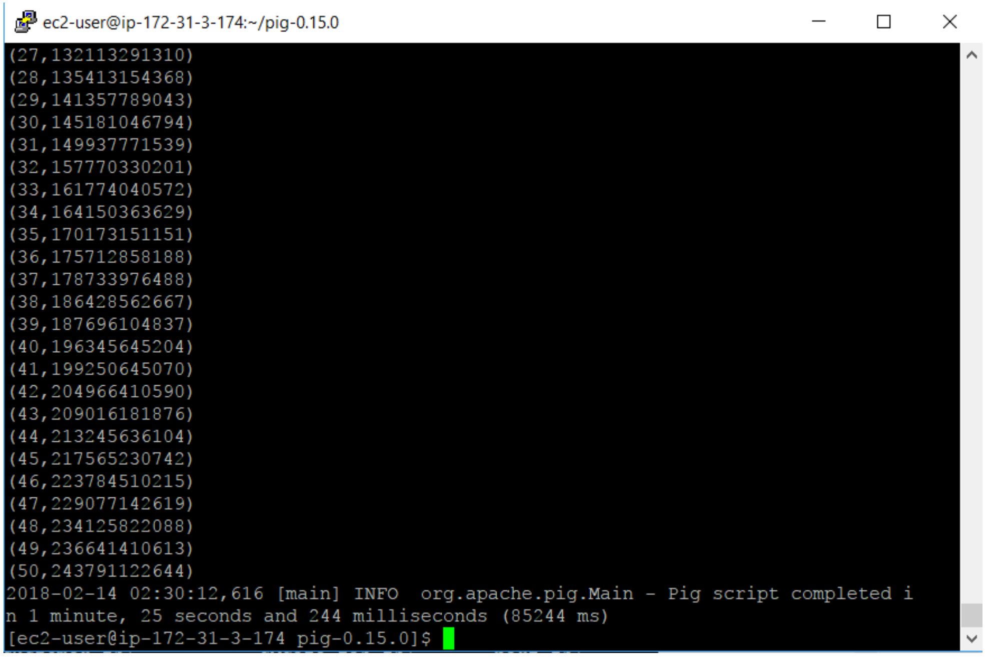
Cat command output:



```
24      116527702603
25      123160894092
26      126451771059
27      132113291310
28      135413154368
29      141357789043
30      145181046794
31      149937771539
32      157770330201
33      161774040572
34      164150363629
35      170173151151
36      175712858188
37      178733976488
38      186428562667
39      187696104837
40      196345645204
41      199250645070
42      204966410590
43      209016181876
44      213245636104
45      217565230742
46      223784510215
47      229077142619
48      234125822088
49      236641410613
50      243791122644
[ec2-user@ip-172-31-3-174 hadoop-2.6.4]$
```

Note that the results match the result of the query run with Pig in part 3. This verifies that the Hadoop streaming command and python mapper and reducer code worked properly.

Pig query 0.3 output from part 3:



```
(27,132113291310)
(28,135413154368)
(29,141357789043)
(30,145181046794)
(31,149937771539)
(32,157770330201)
(33,161774040572)
(34,164150363629)
(35,170173151151)
(36,175712858188)
(37,178733976488)
(38,186428562667)
(39,187696104837)
(40,196345645204)
(41,199250645070)
(42,204966410590)
(43,209016181876)
(44,213245636104)
(45,217565230742)
(46,223784510215)
(47,229077142619)
(48,234125822088)
(49,236641410613)
(50,243791122644)
2018-02-14 02:30:12,616 [main] INFO  org.apache.pig.Main - Pig script completed i
n 1 minute, 25 seconds and 244 milliseconds (85244 ms)
[ec2-user@ip-172-31-3-174 pig-0.15.0]$
```