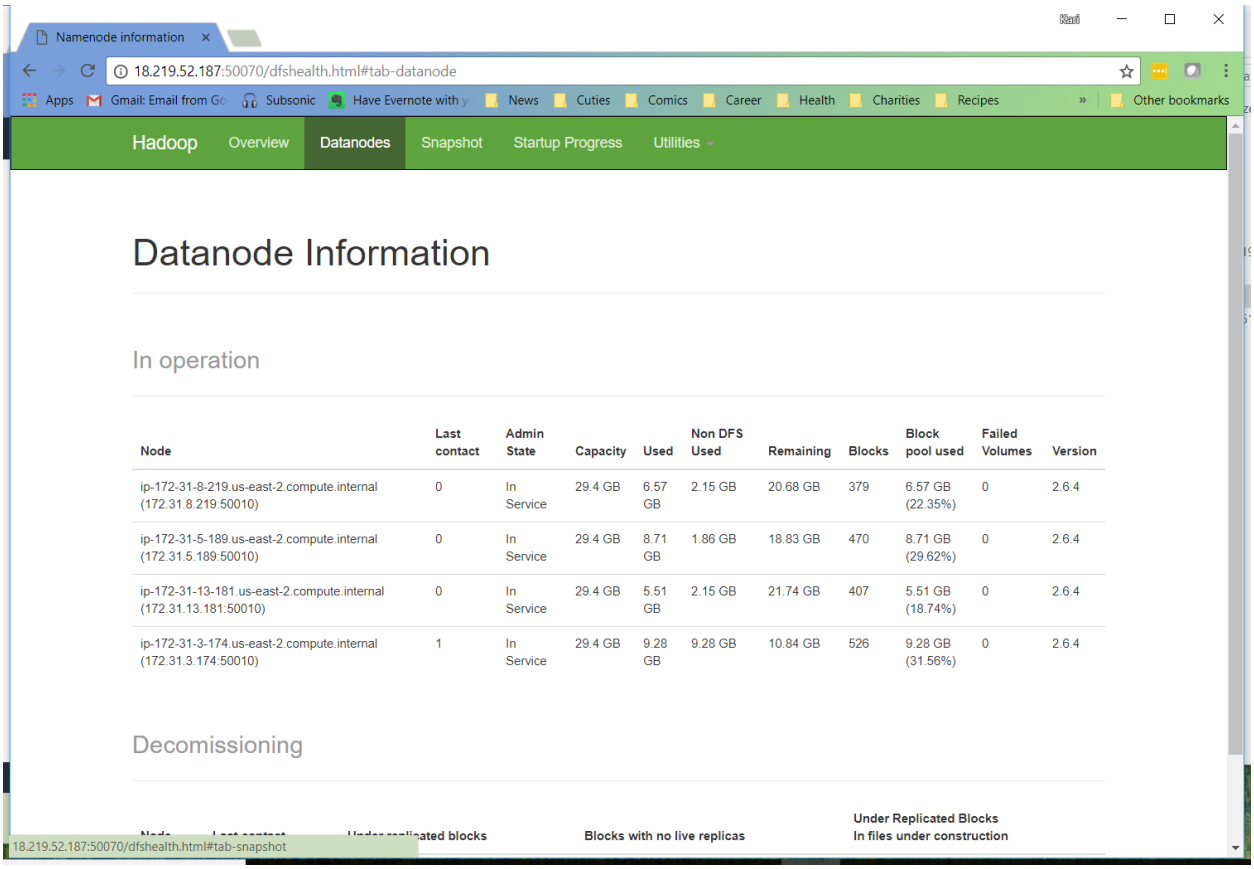
Kari Palmier
CSC 555 Winter 2018
Project Phase 2

I used my 4-node cluster from assignment 5 to perform all of the steps for parts 1 through 3. I used wget to get all of the scale 4 tables from the website given in the project instructions. Before doing this, I deleted any existing tables from the assignments or the project phase 1. I copied the scale 4 tables into HDFS using the hadoop fs -put filename /user/ec2-user command. All instances in the cluster were t2 medium instances with 30 GB of hard drive space each. The HDFS replication factor was set to 2 in the cluster.

4-node cluster screenshot:



Part 1

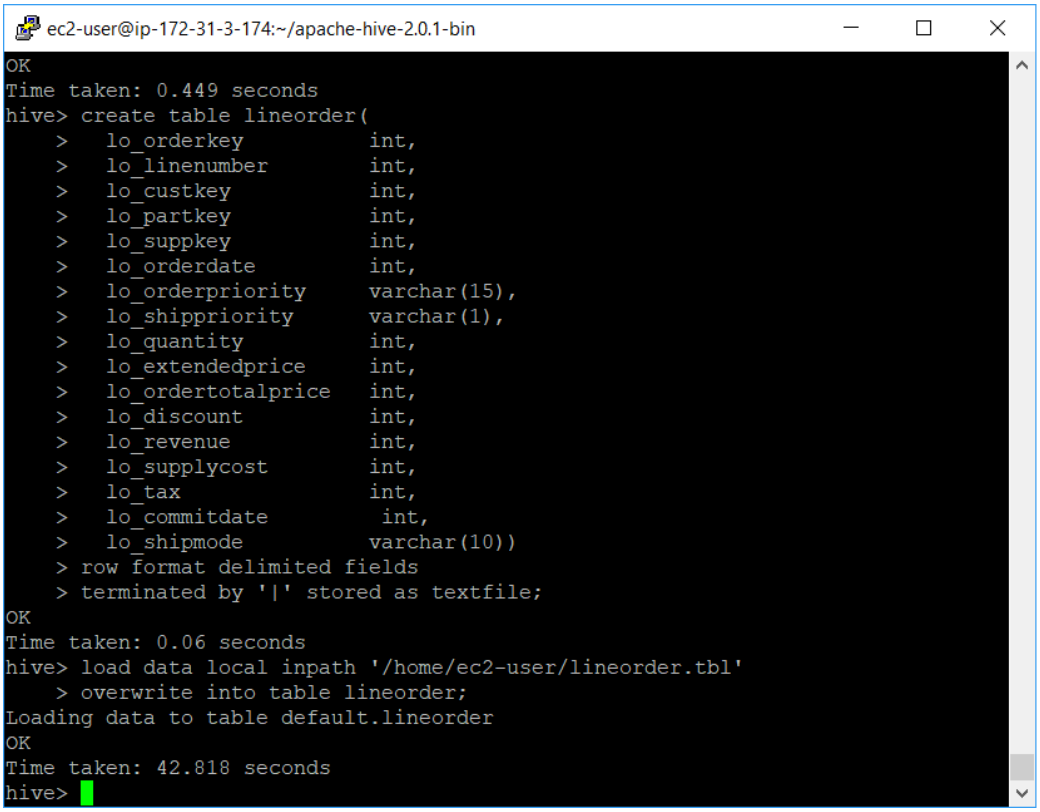
A)

Hive lineorder.tbl Transformation From | Separated to CSV

Hive Table Creation Code:

```
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;
```

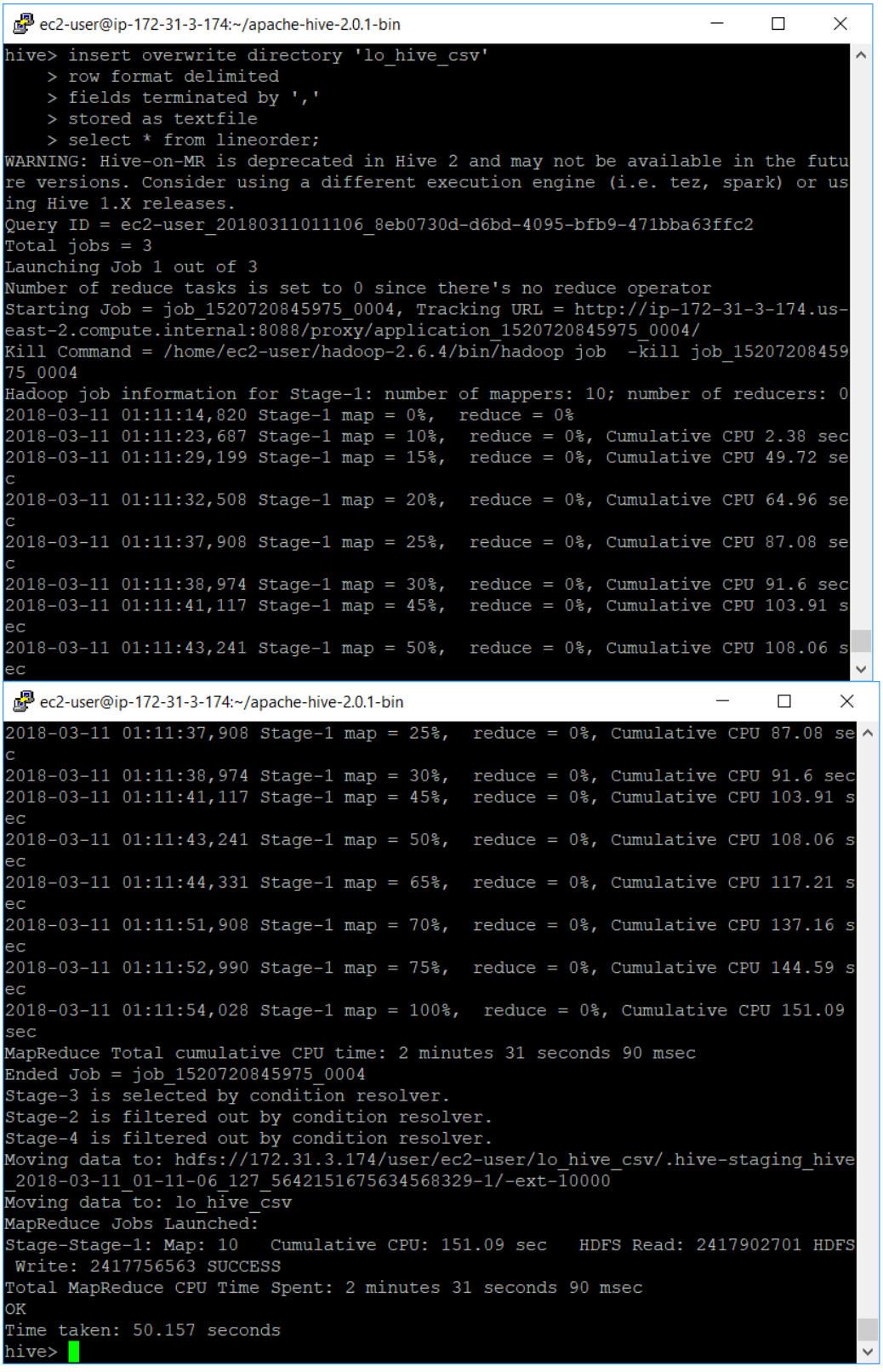
Hive Table Creation Execution:



Hive Transformation Code:

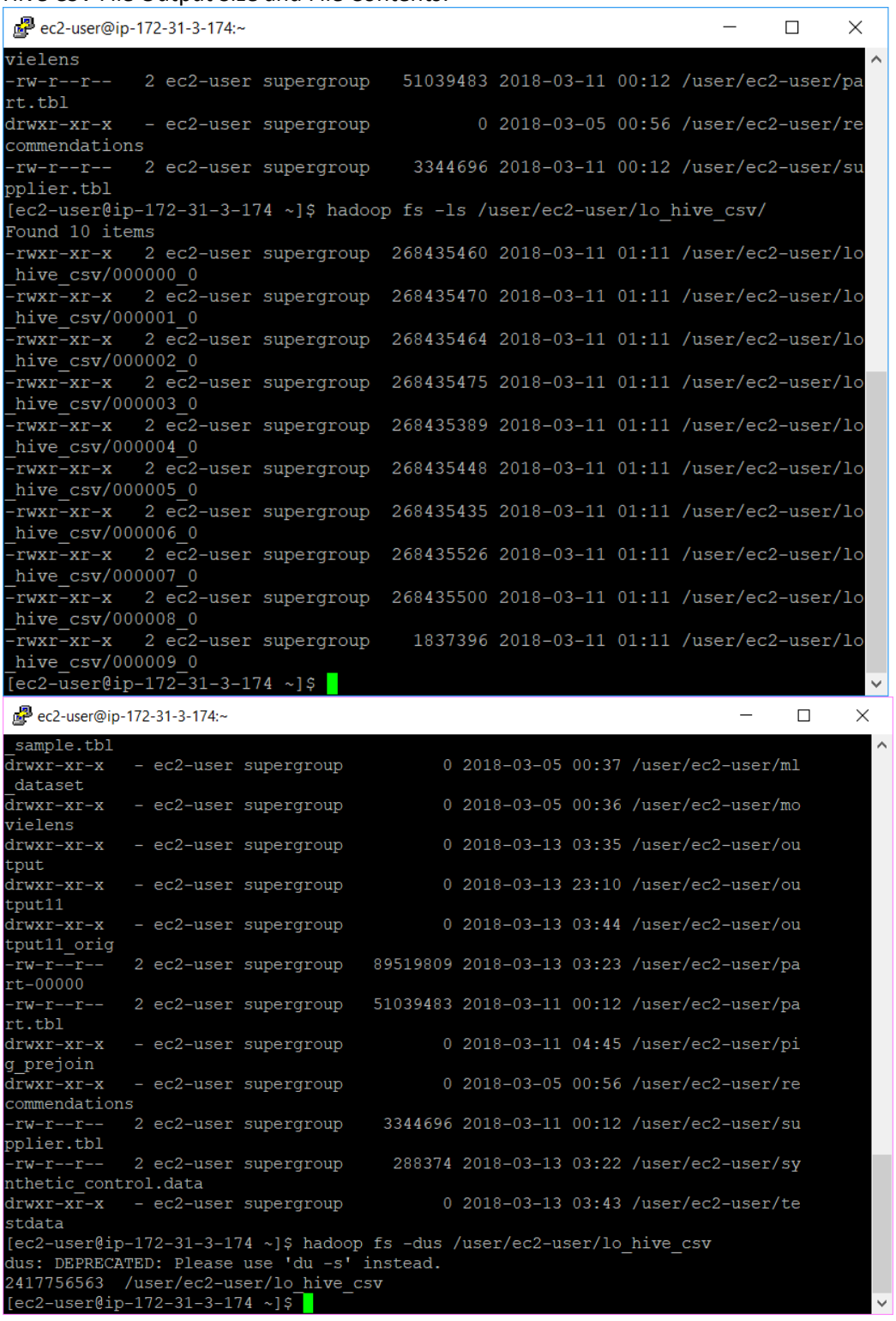
```
insert overwrite directory 'lo_hive_csv'
row format delimited
fields terminated by ','
stored as textfile
select * from lineorder;
```

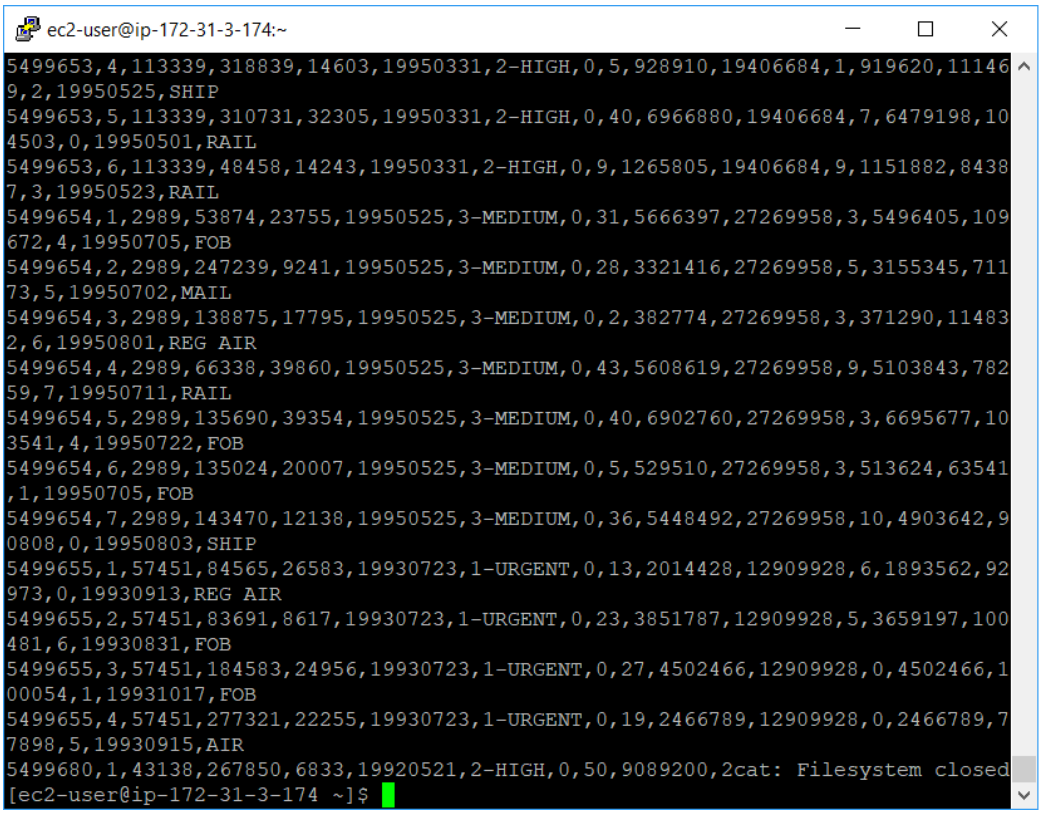
Hive Transformation Execution:



The Hive transformation code took 50.157 secs to execute on the 4-node cluster.

Hive CSV File Output Size and File Contents:

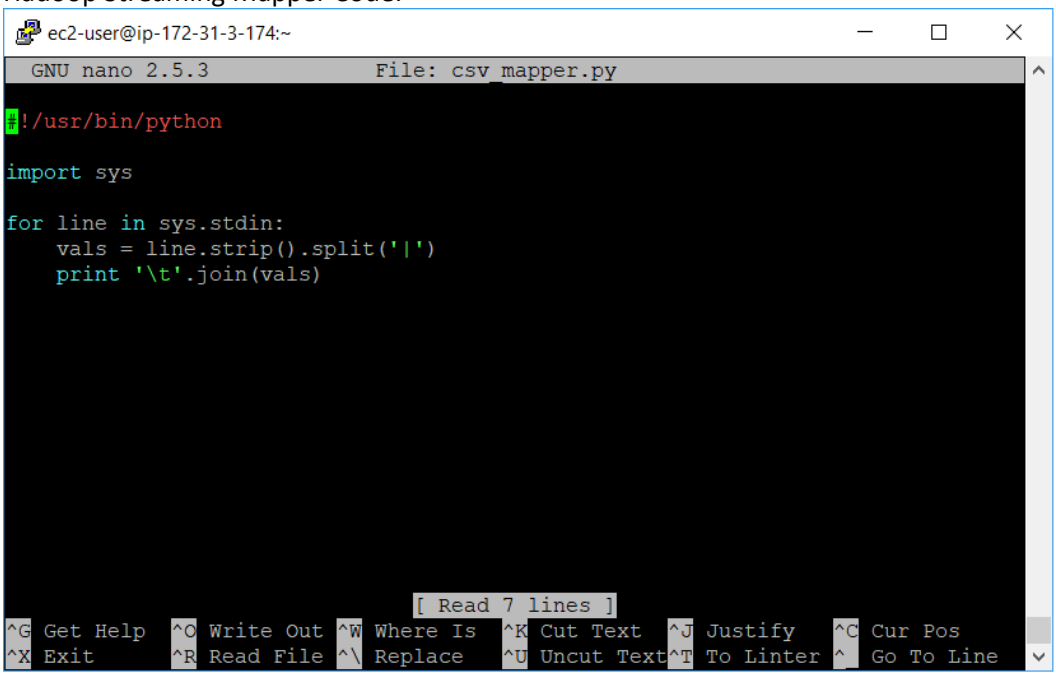




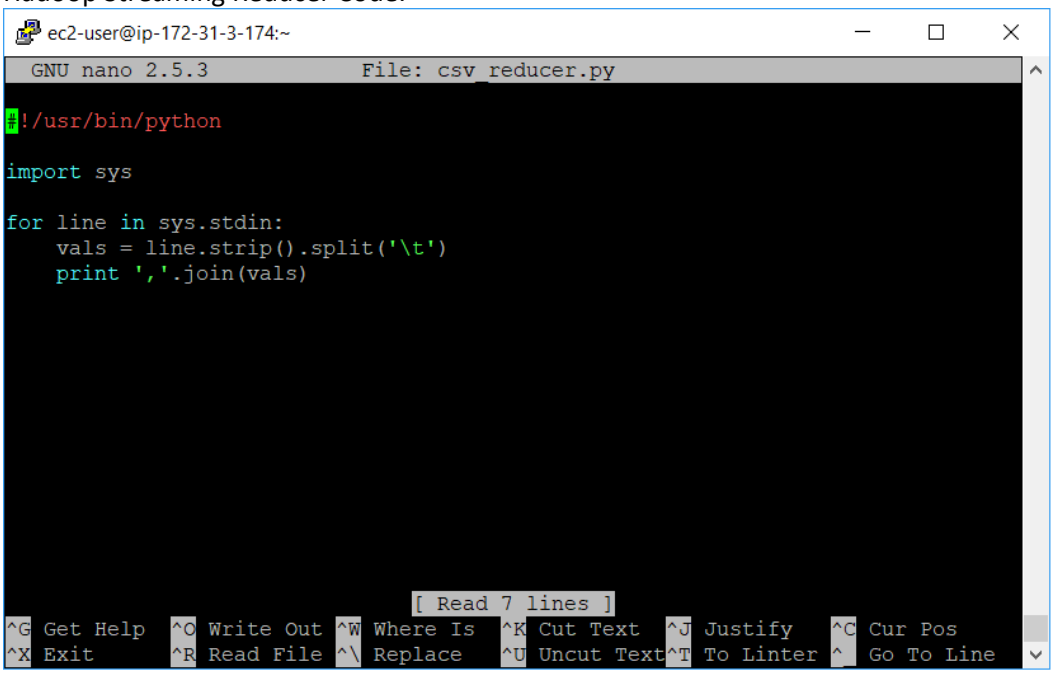
The size of the Hive CSV output directory is 2417756563 bytes or approximately 2.418 GB.

Hadoop Streaming lineorder.tbl Transformation From | Separated to CSV

Hadoop Streaming Mapper Code:



Hadoop Streaming Reducer Code:

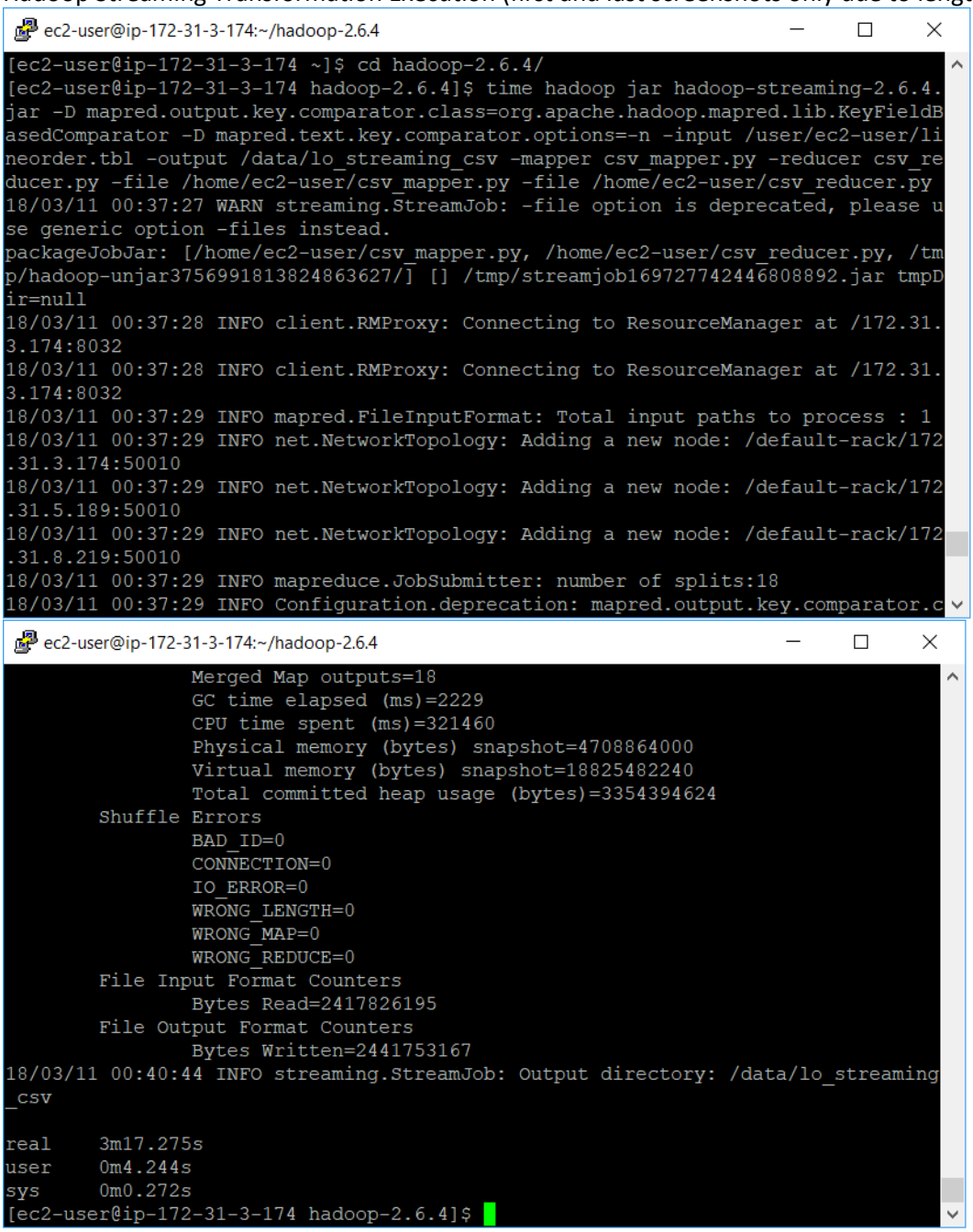


Hadoop Streaming Command Used:

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/lo_streaming_csv -mapper csv_mapper.py -reducer csv_reducer.py -file /home/ec2-user/csv_mapper.py -file /home/ec2-user/csv_reducer.py

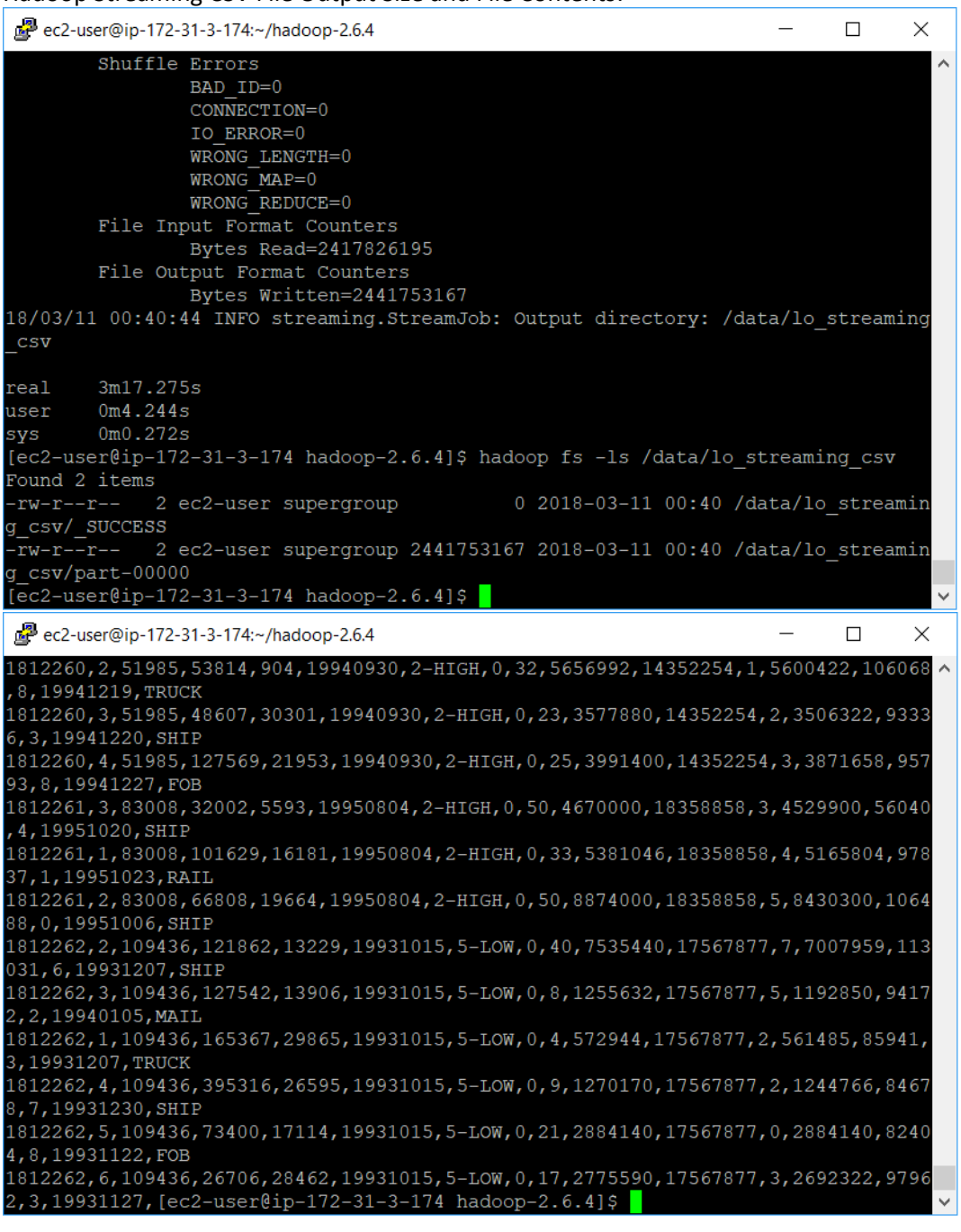
The mapper and reducer keys are set to the lo_orderkey value of the lineorder table, which is an integer. Note that I used the KeyFileBasedComparator and comparator.options fields so the key from the mapper would be interpreted as a number instead of a string.

Hadoop Streaming Transformation Execution (first and last screenshots only due to length):



The Hadoop Streaming transformation code took 3 min and 17.27 secs (or 197.27 secs) to execute on the 4-node cluster.

Hadoop Streaming CSV File Output Size and File Contents:



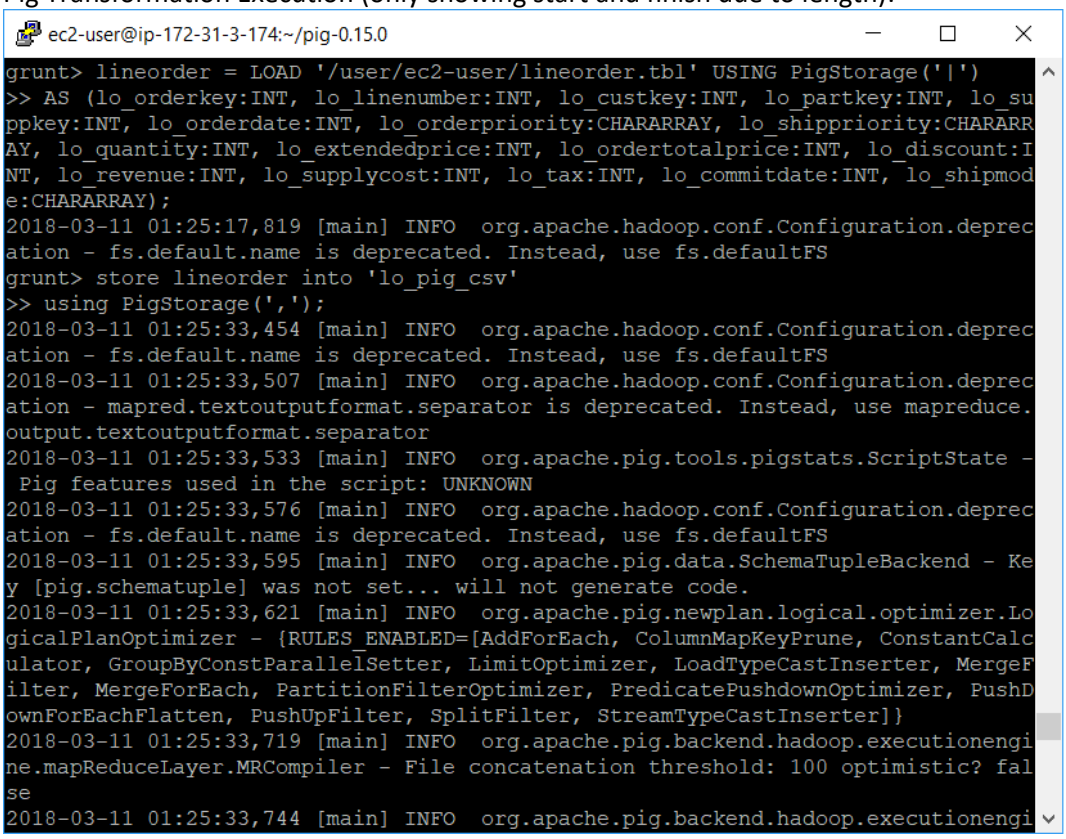
The size of the Hadoop Streaming CSV output file is 2441753167 bytes or approximately 2.442 GB.

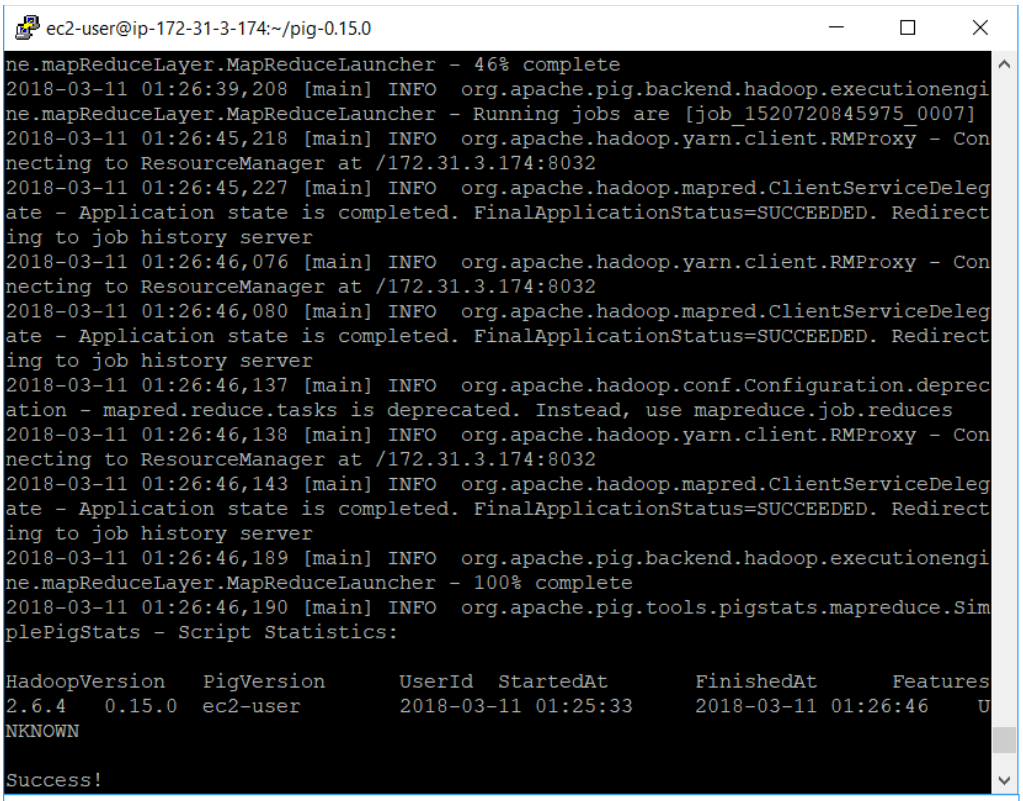
Pig lineorder.tbl Transformation From | Separated to CSV

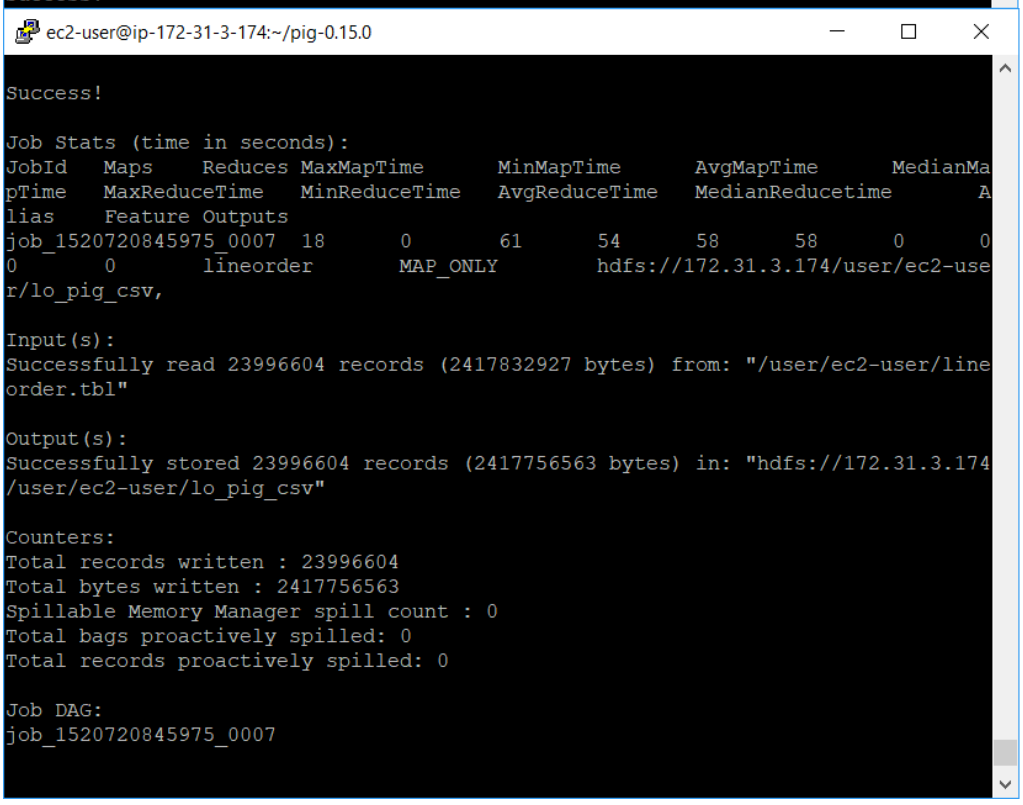
Pig Transformation Code:

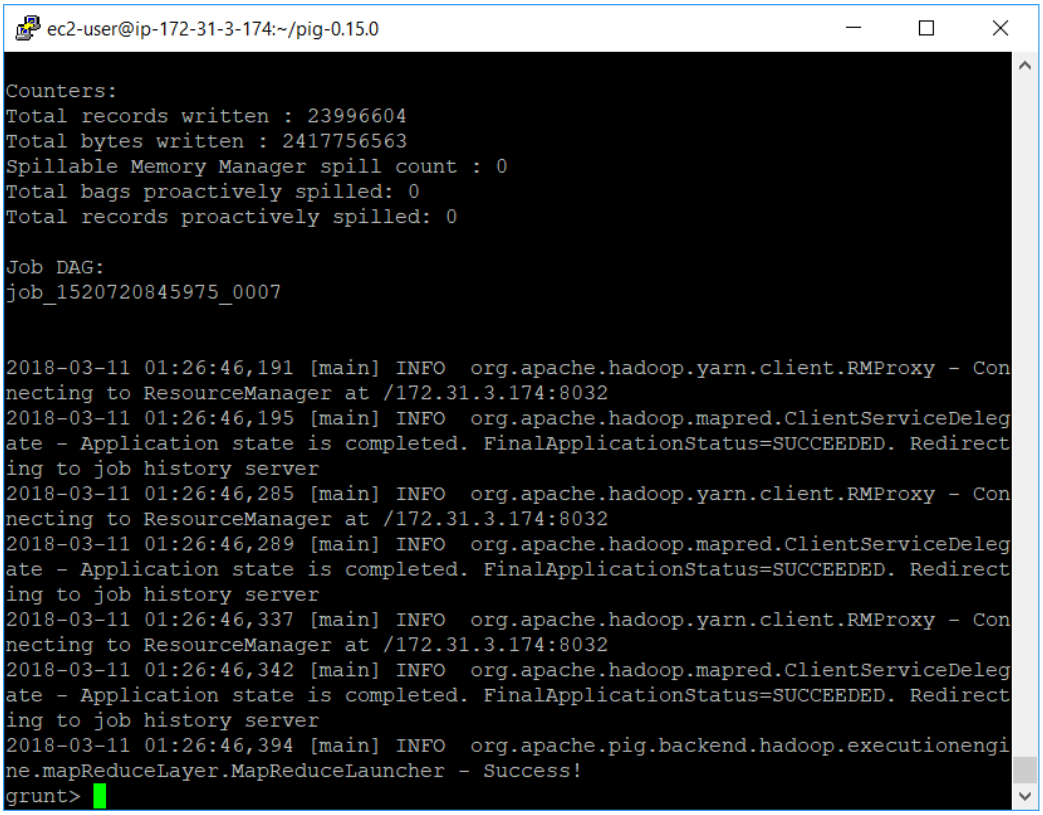
```
lineorder = LOAD '/user/ec2-user/lineorder.tbl' USING PigStorage('|')
AS (lo_orderkey:INT, lo_linenumber:INT, lo_custkey:INT, lo_partkey:INT, lo_suppkey:INT,
lo_orderdate:INT, lo_orderpriority:CHARARRAY, lo_shippriority:CHARARRAY, lo_quantity:INT,
lo_extendedprice:INT, lo_ordertotalprice:INT, lo_discount:INT,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

store lineorder into 'lo_pig_csv'
using PigStorage(',');
```

Pig Transformation Execution (only showing start and finish due to length):



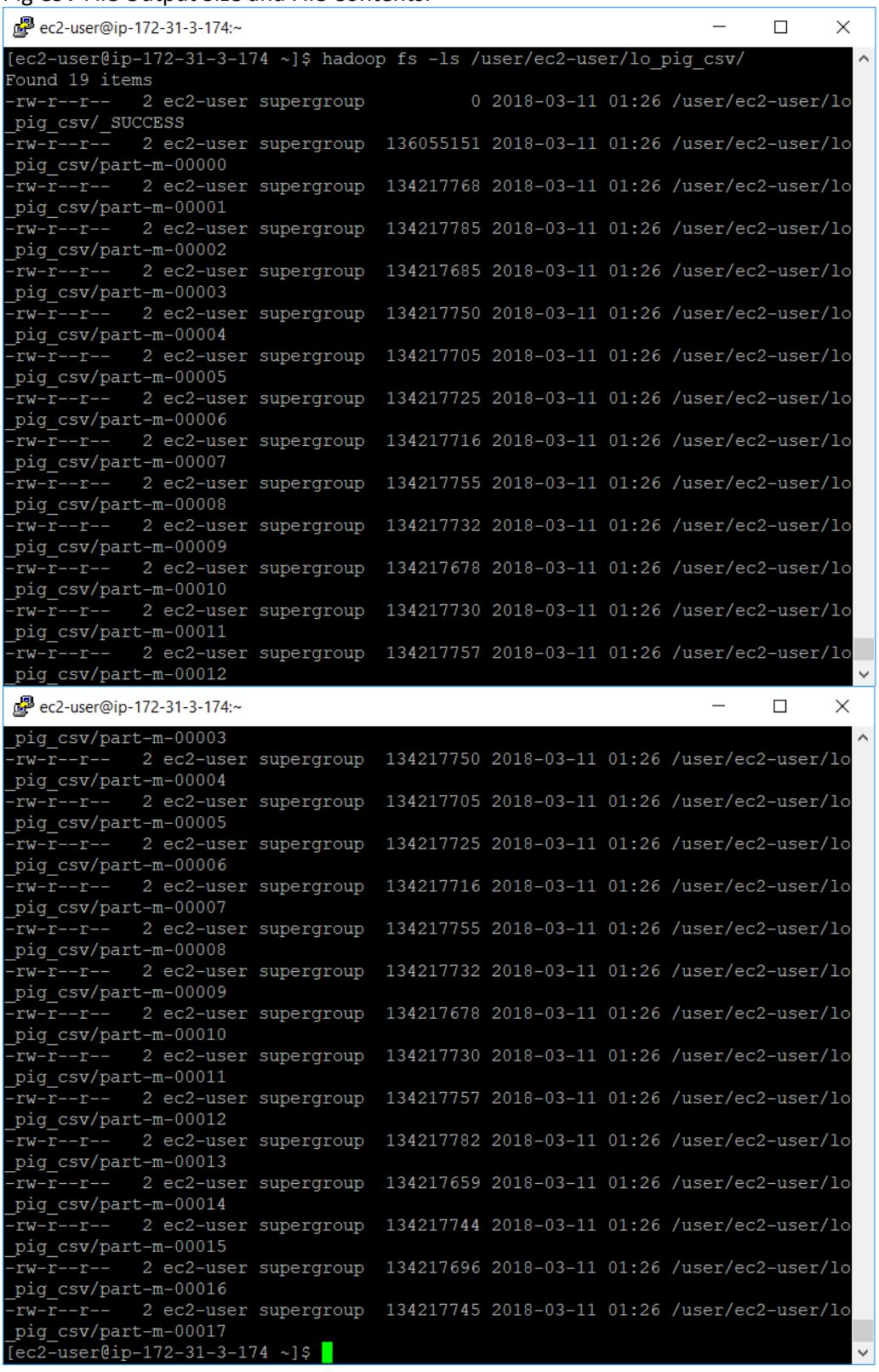

ec2-user@ip-172-31-3-174:~/pig-0.15.0
ne.mapReduceLayer.MapReduceLauncher - 46% complete
2018-03-11 01:26:39,208 [main] INFO org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Running jobs are [job_1520720845975_0007]
2018-03-11 01:26:45,218 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Con
necting to ResourceManager at /172.31.3.174:8032
2018-03-11 01:26:45,227 [main] INFO org.apache.hadoop.mapred.ClientServiceDeleg
ate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirect
ing to job history server
2018-03-11 01:26:46,076 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Con
necting to ResourceManager at /172.31.3.174:8032
2018-03-11 01:26:46,080 [main] INFO org.apache.hadoop.mapred.ClientServiceDeleg
ate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirect
ing to job history server
2018-03-11 01:26:46,137 [main] INFO org.apache.hadoop.conf.Configuration.deprec
ation - mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces
2018-03-11 01:26:46,138 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Con
necting to ResourceManager at /172.31.3.174:8032
2018-03-11 01:26:46,143 [main] INFO org.apache.hadoop.mapred.ClientServiceDeleg
ate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirect
ing to job history server
2018-03-11 01:26:46,189 [main] INFO org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - 100% complete
2018-03-11 01:26:46,190 [main] INFO org.apache.pig.tools.pigstats.mapreduce.Sim
plePigStats - Script Statistics:
HadoopVersion PigVersion UserId StartedAt FinishedAt Features
2.6.4 0.15.0 ec2-user 2018-03-11 01:25:33 2018-03-11 01:26:46 U
NKNOWN
Success!

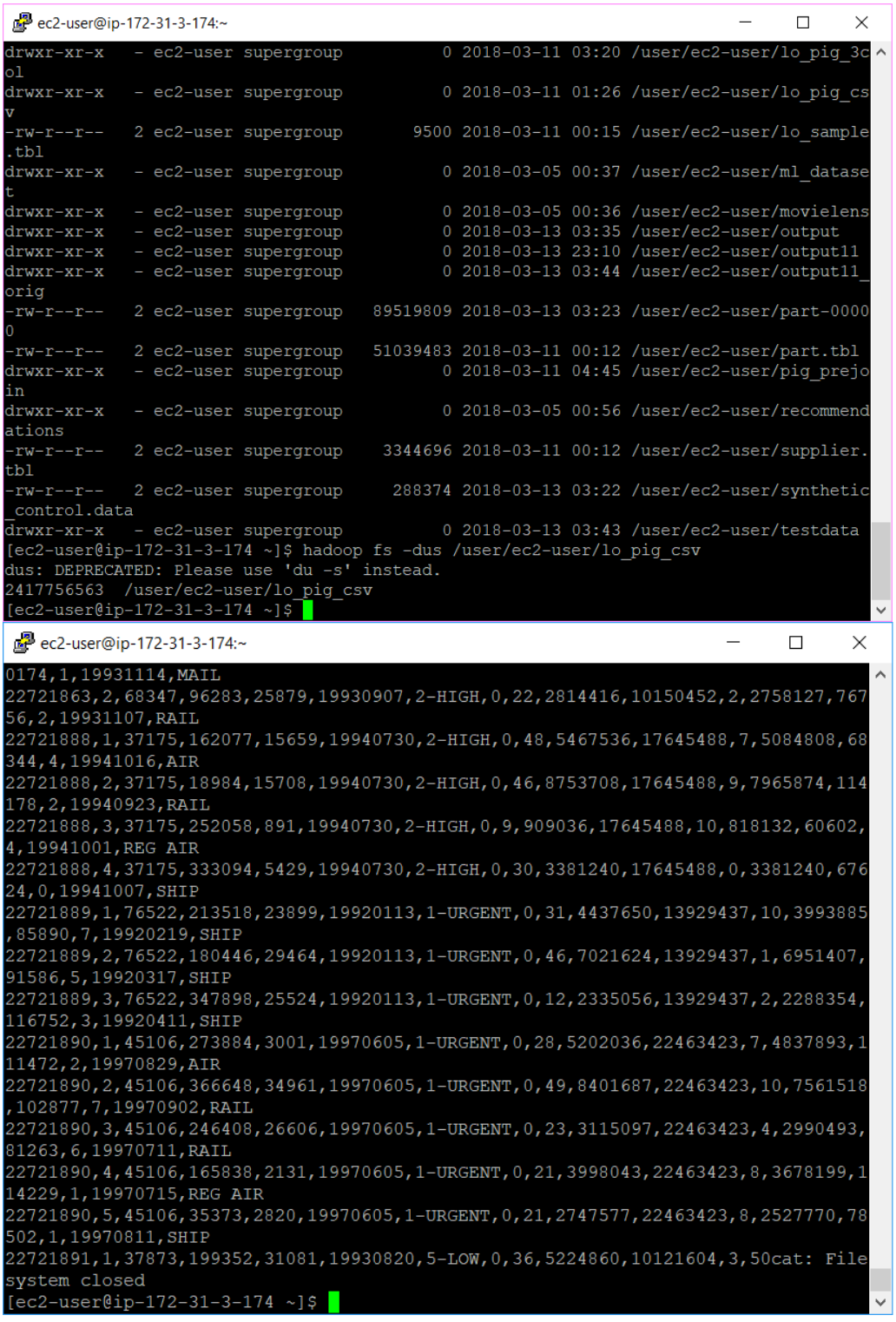

ec2-user@ip-172-31-3-174:~/pig-0.15.0
Success!
Job Stats (time in seconds):
JobId Maps Reduces MaxMapTime MinMapTime AvgMapTime MedianMa
pTime MaxReduceTime MinReduceTime AvgReduceTime MedianReducetime A
lias Feature Outputs
job_1520720845975_0007 18 0 61 54 58 58 0 0
0 0 lineorder MAP_ONLY hdfs://172.31.3.174/user/ec2-use
r/lo_pig_csv,
Input(s):
Successfully read 23996604 records (2417832927 bytes) from: "/user/ec2-user/line
order.tbl"
Output(s):
Successfully stored 23996604 records (2417756563 bytes) in: "hdfs://172.31.3.174
/user/ec2-user/lo_pig_csv"
Counters:
Total records written : 23996604
Total bytes written : 2417756563
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
Job DAG:
job_1520720845975_0007



The Pig transformation code started at 1:25:33 and ended at 1:26:46, so it took 1 min 13 secs (or 73 secs) to execute on the 4-node cluster.

Pig CSV File Output Size and File Contents:





The size of the Pig CSV output directory is 2417756563 bytes or approximately 2.418 GB.

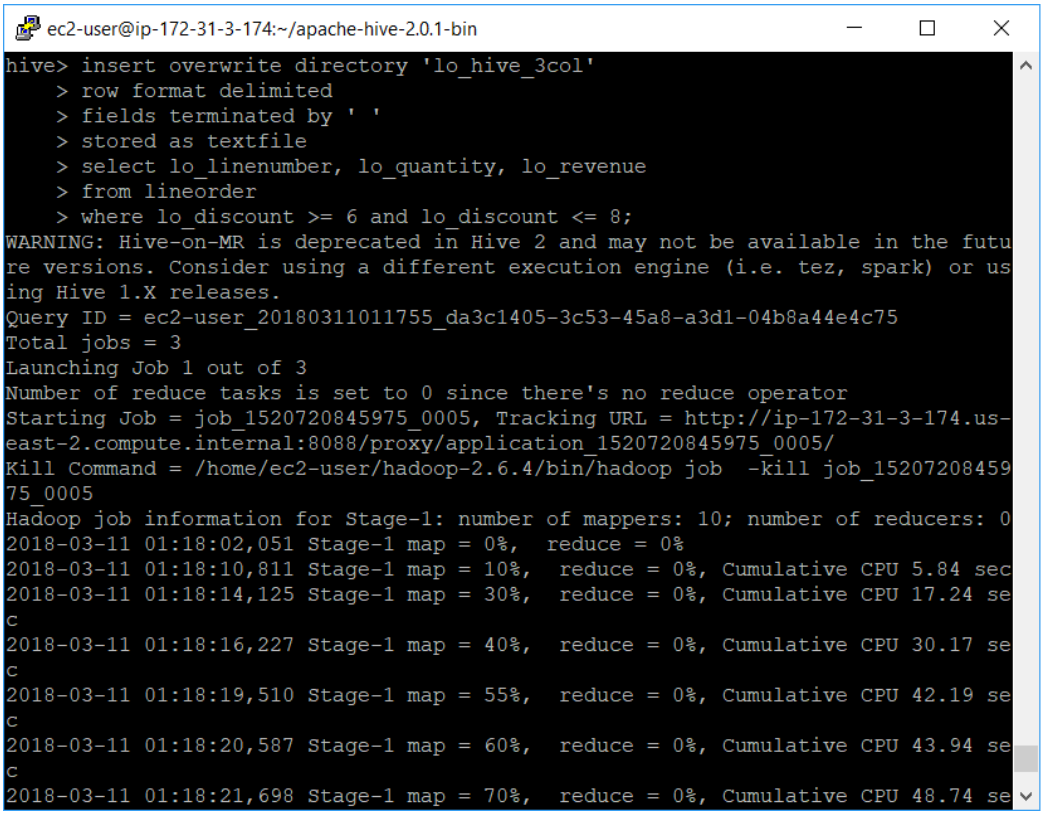
B)

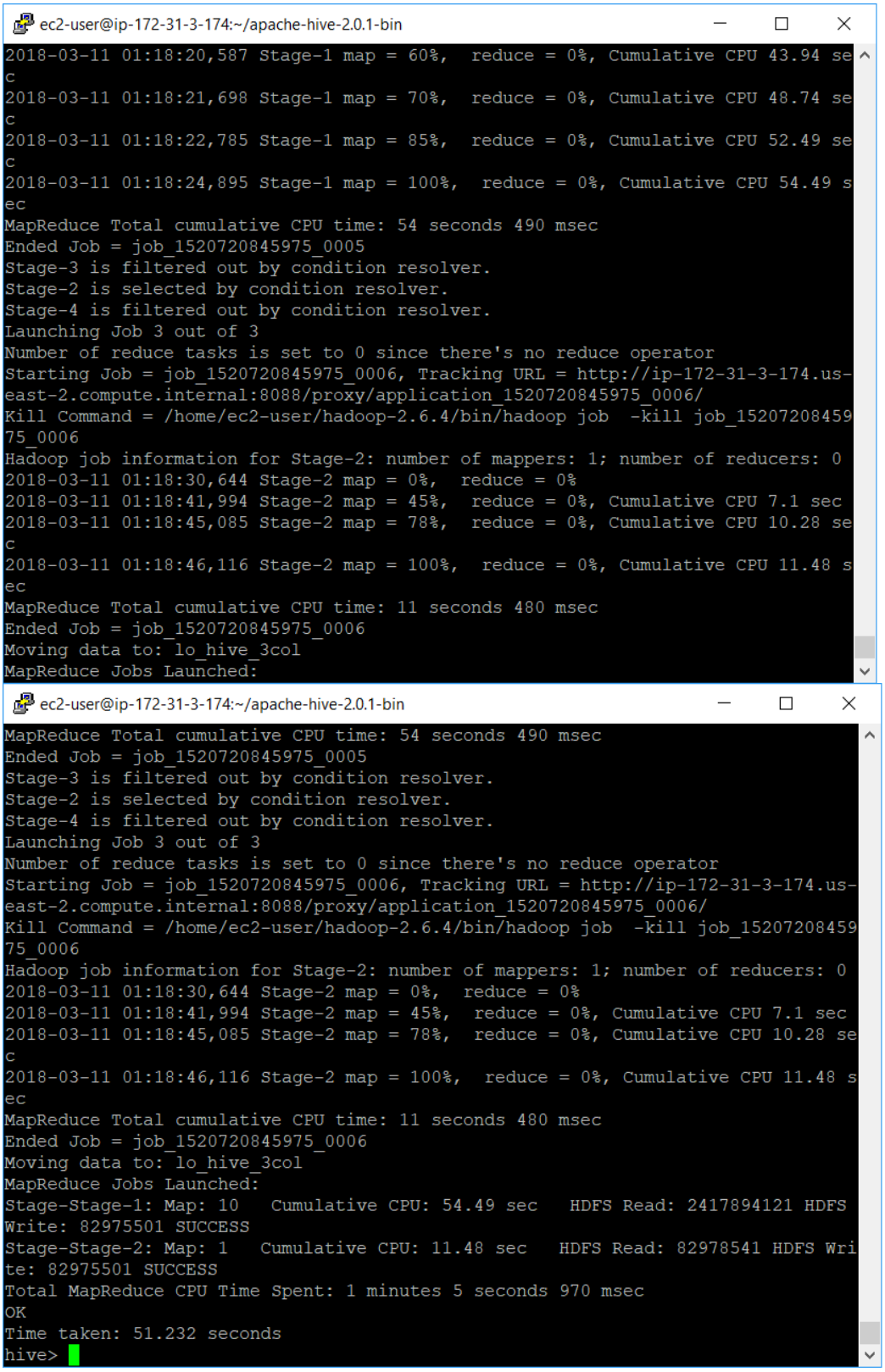
Hive lineorder.tbl 3 Column Extraction

Hive 3 Column Extraction Code:

```
insert overwrite directory 'lo_hive_3col'
row format delimited
fields terminated by ' '
stored as textfile
select lo_linenumber, lo_quantity, lo_revenue
from lineorder
where lo_discount >= 6 and lo_discount <= 8;
```

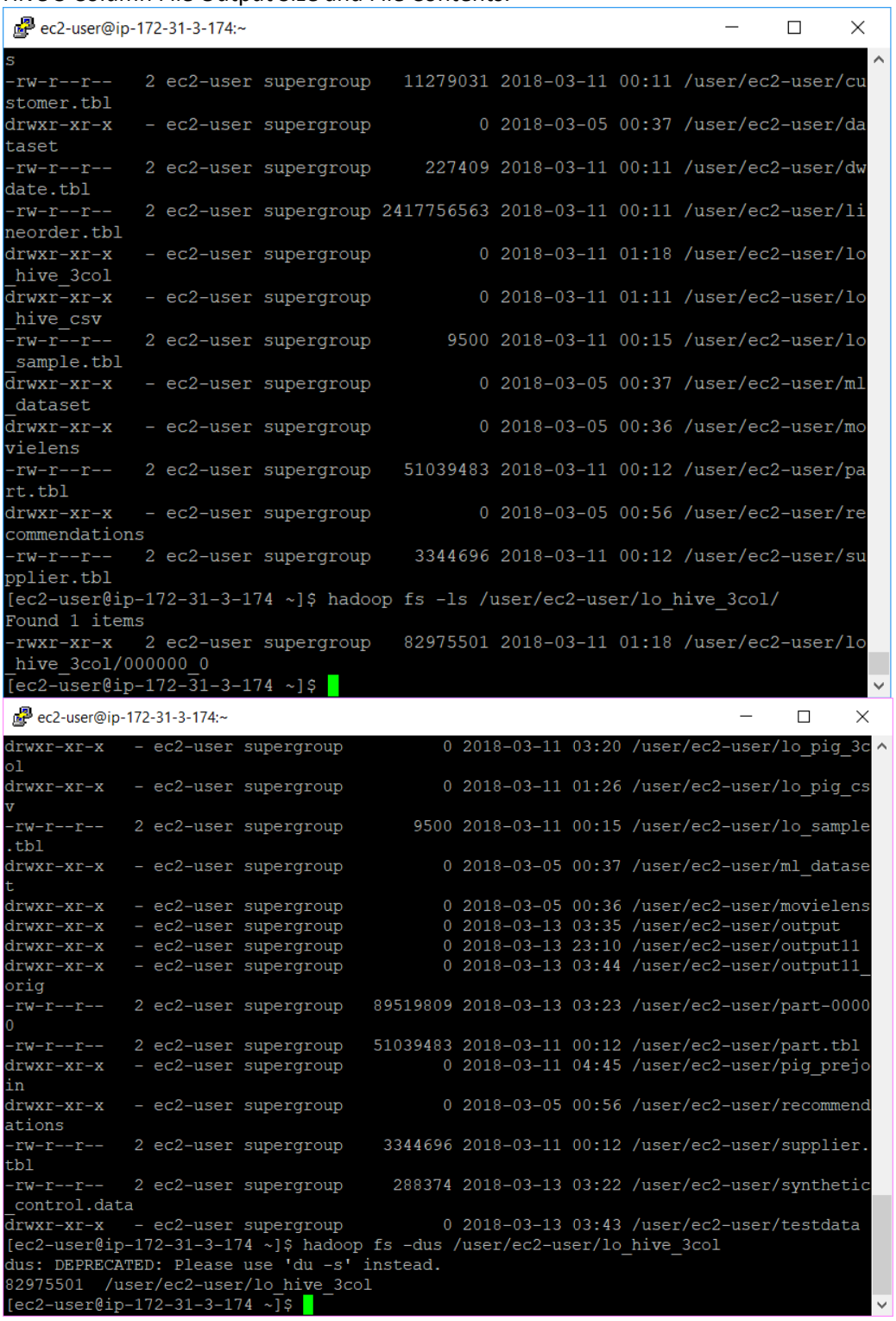
Hive 3 Column Extraction Execution:

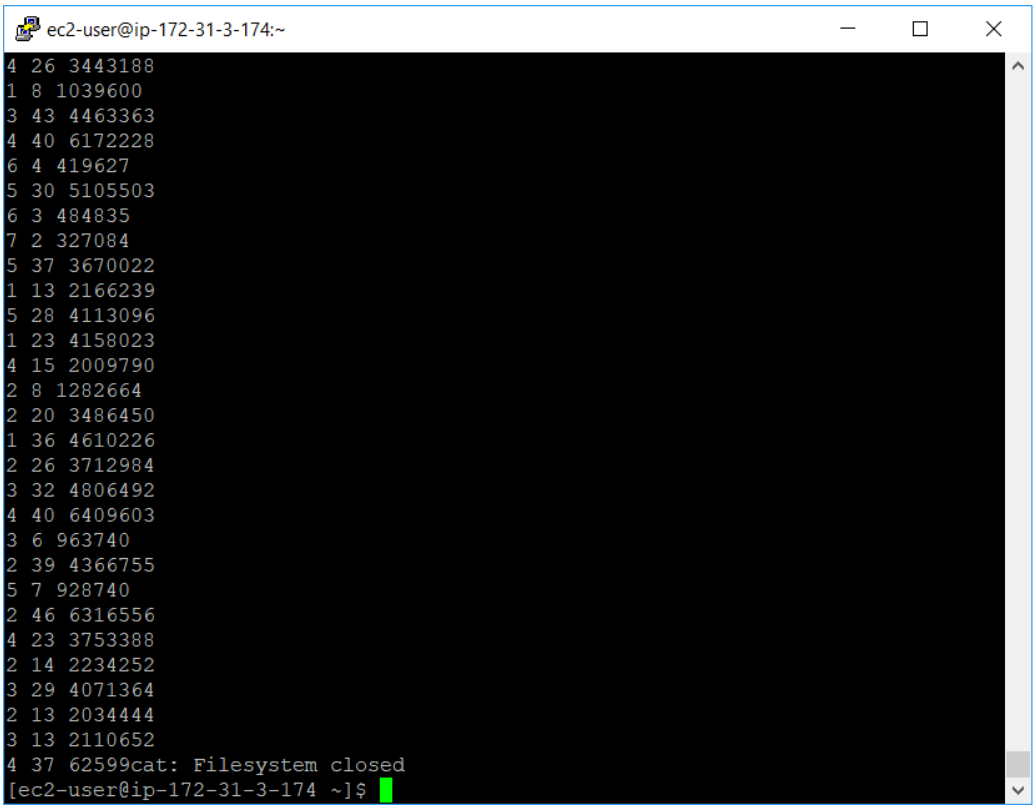




The Hive 3 column extraction code took 51.232 secs to execute on the 4-node cluster.

Hive 3 Column File Output Size and File Contents:

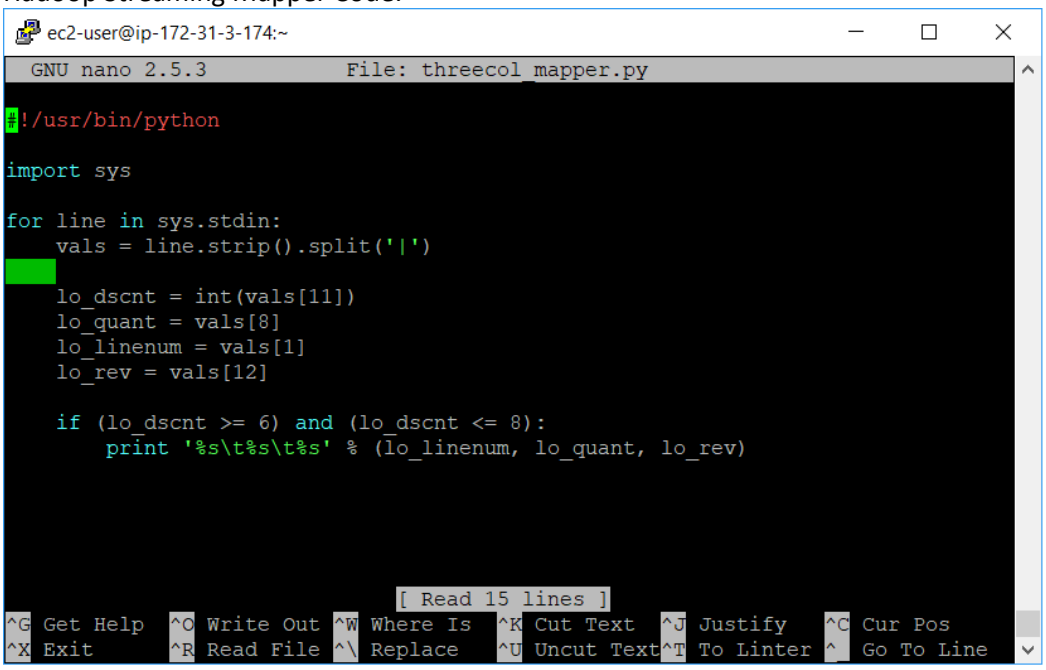




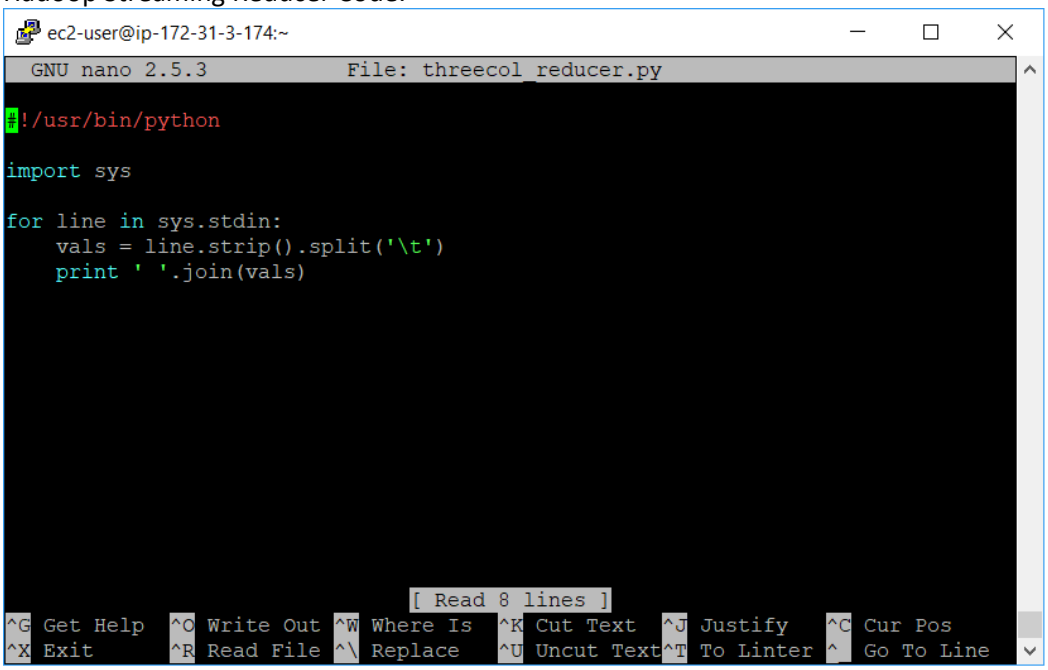
The size of the Hive 3 column output directory is 82975501 bytes or approximately 82.976 MB.

Hadoop Streaming lineorder.tbl 3 Column Extraction

Hadoop Streaming Mapper Code:



Hadoop Streaming Reducer Code:

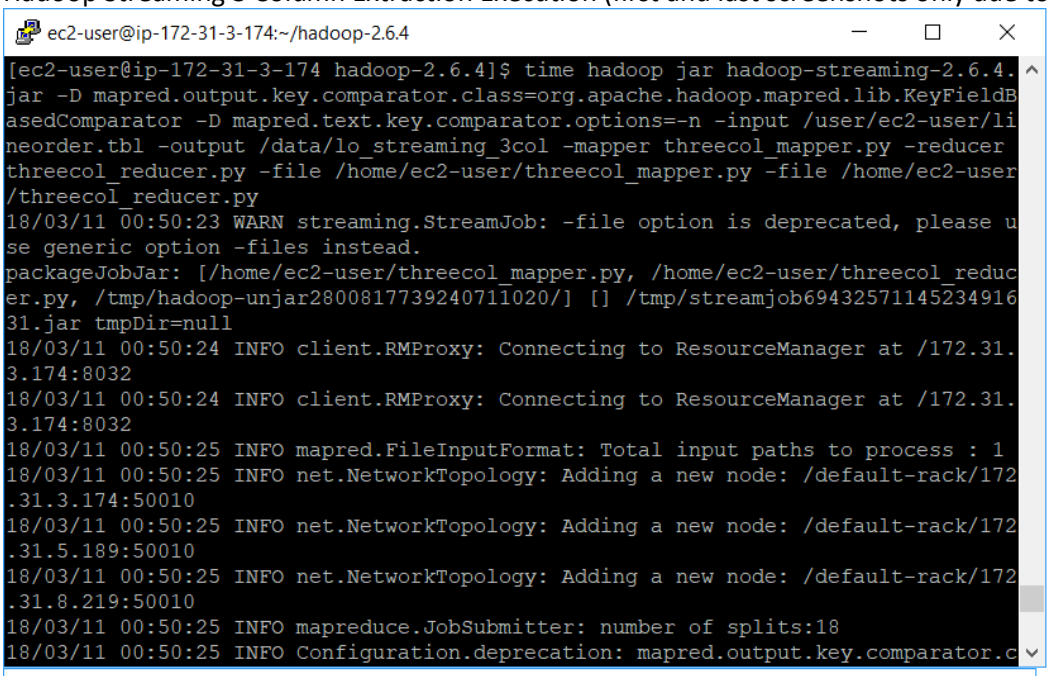


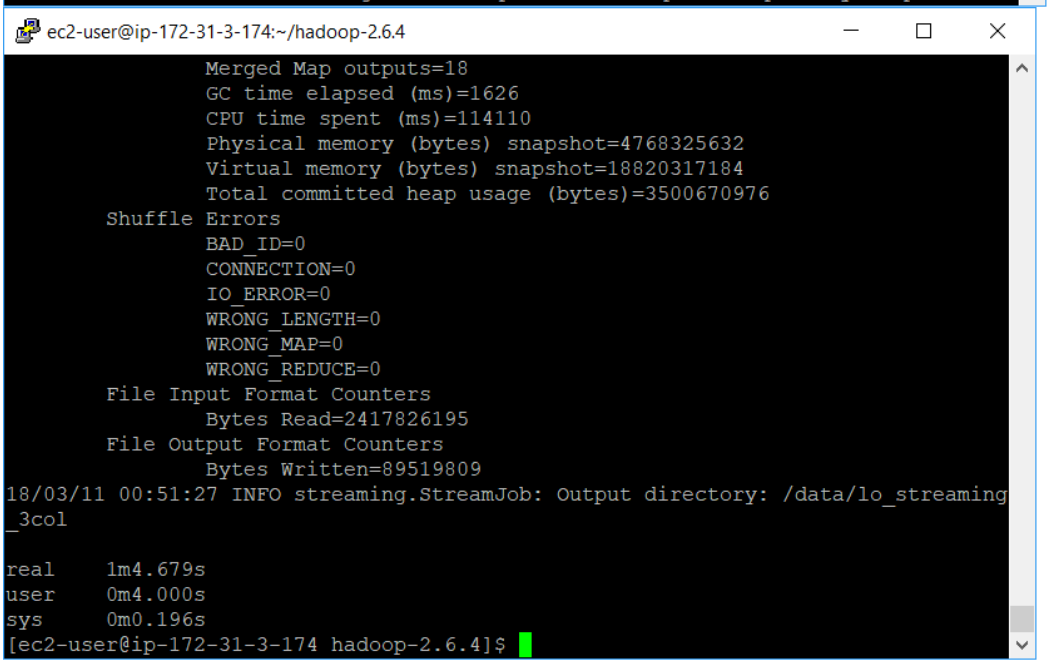
Hadoop Streaming Command Used:

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/lo_streaming_3col -mapper threecol_mapper.py -reducer threecol_reducer.py -file /home/ec2-user/threecol_mapper.py -file /home/ec2-user/threecol_reducer.py

The mapper key is set to the lo_linenumber value of the lineorder table, which is an integer. There is no key set in the reducer output because the goal is to have the columns separated by spaces (a key would induce a tab). Note that I used the KeyFileBasedComparator and comparator.options fields so the key from the mapper would be interpreted as a number instead of a string.

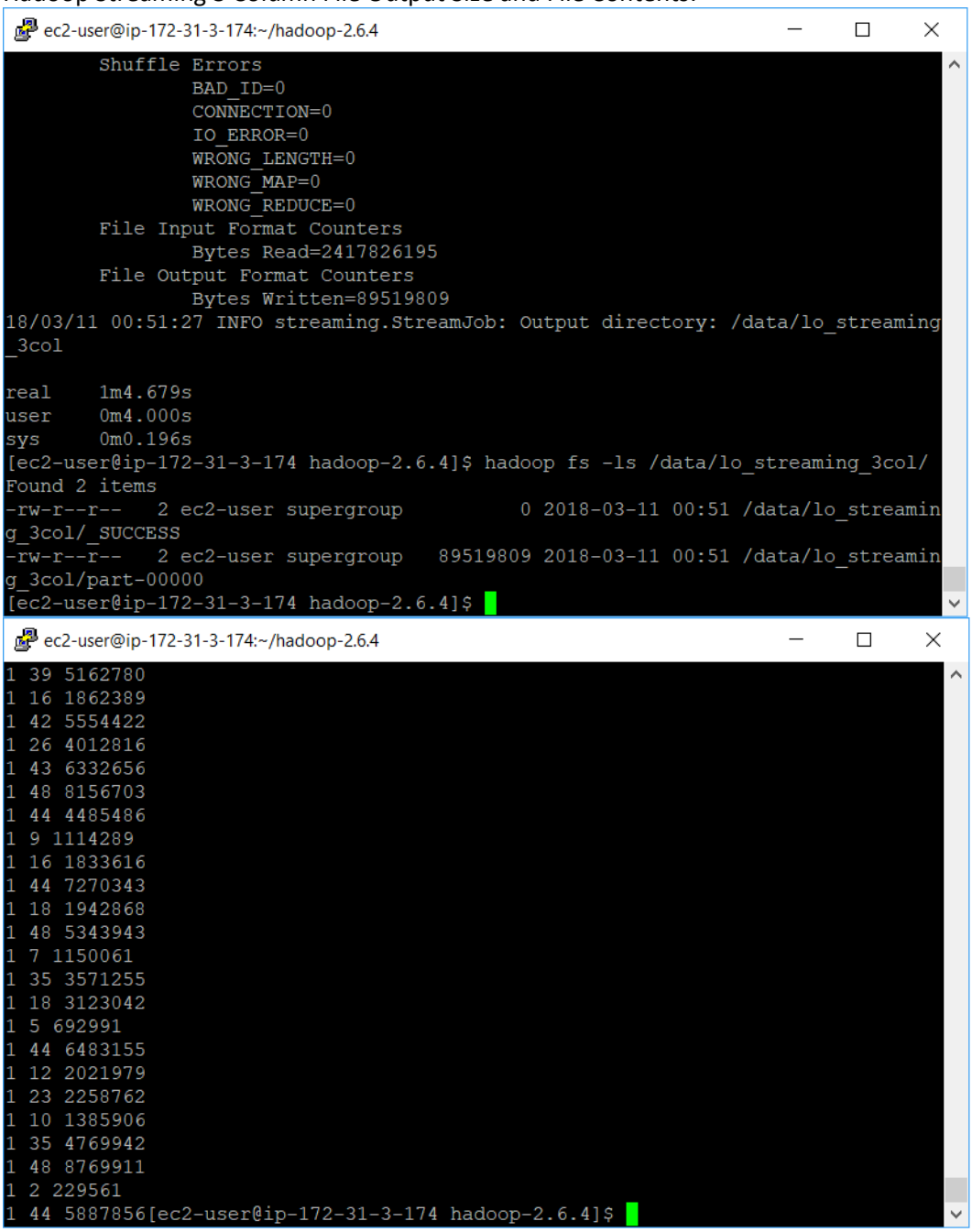
Hadoop Streaming 3 Column Extraction Execution (first and last screenshots only due to length):





The Hadoop Streaming transformation code took 1 min and 4.679 secs (or 64.679 secs) to execute on the 4-node cluster

Hadoop Streaming 3 Column File Output Size and File Contents:



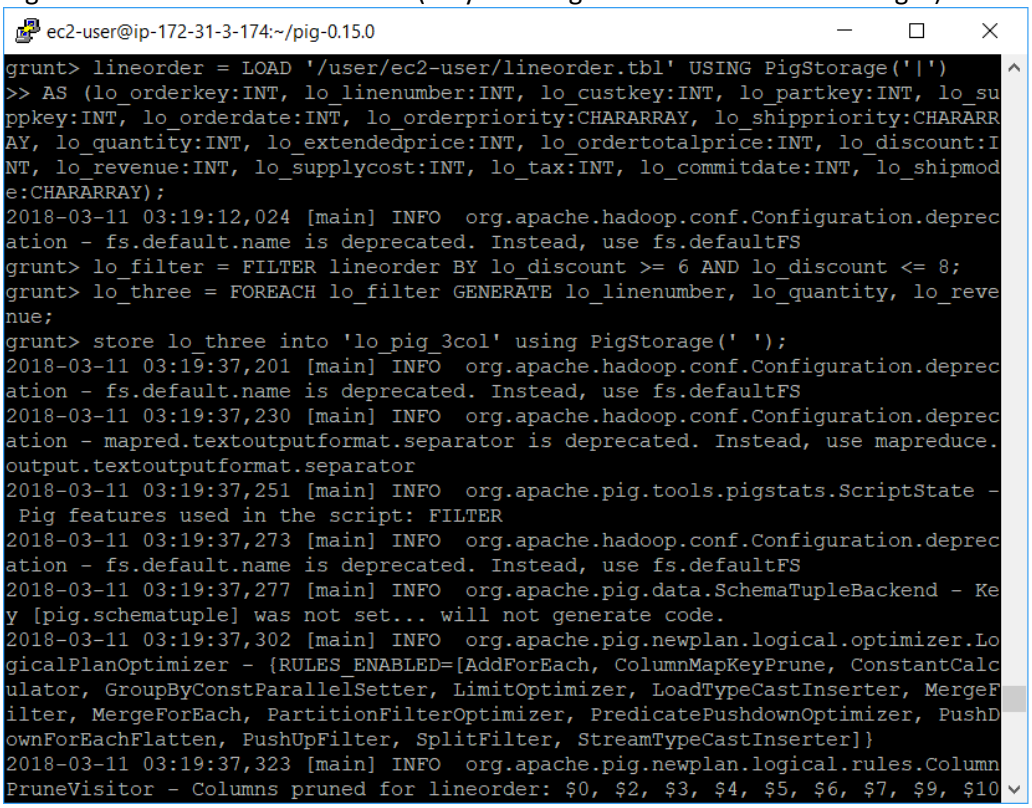
The size of the Hadoop Streaming 3 column output file is 89519809 bytes or approximately 89.52 MB.

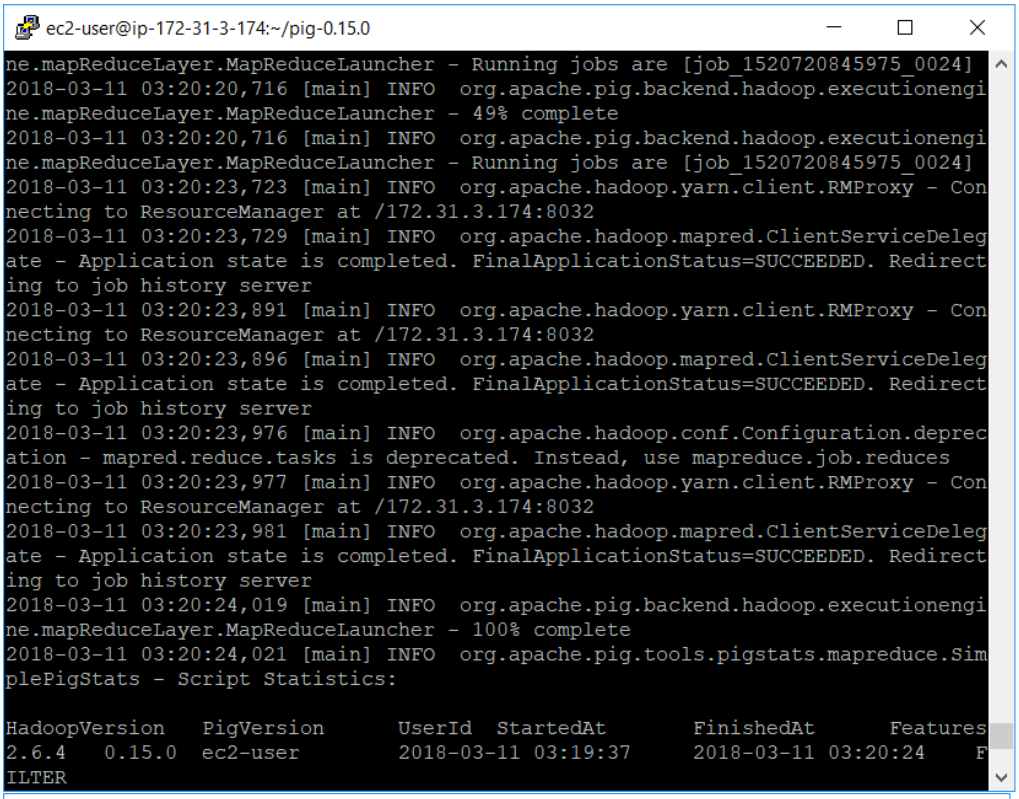
Pig lineorder.tbl 3 Column Extraction

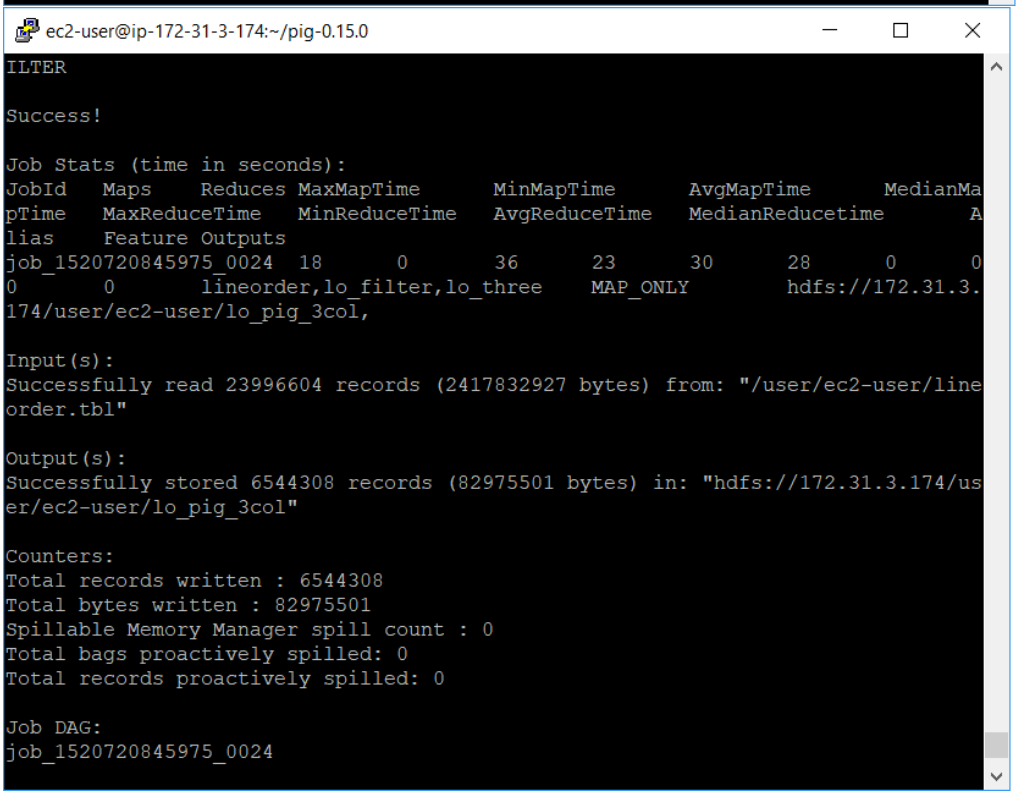
Pig 3 Column Extraction Code:

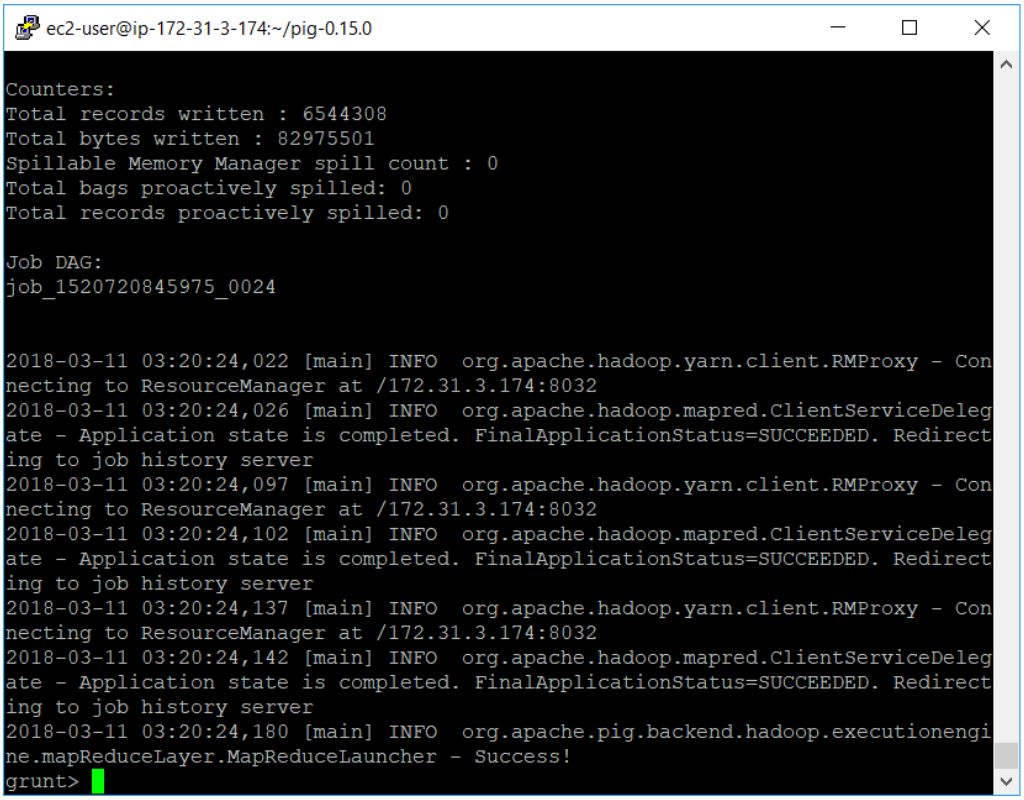
```
lineorder = LOAD '/user/ec2-user/lineorder.tbl' USING PigStorage('|')
AS (lo_orderkey:INT, lo_linenumber:INT, lo_custkey:INT, lo_partkey:INT, lo_suppkey:INT,
lo_orderdate:INT, lo_orderpriority:CHARARRAY, lo_shippriority:CHARARRAY, lo_quantity:INT,
lo_extendedprice:INT, lo_ordertotalprice:INT, lo_discount:INT,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

lo_filter = FILTER lineorder BY lo_discount >= 6 AND lo_discount <= 8;
lo_three = FOREACH lo_filter GENERATE lo_linenumber, lo_quantity, lo_revenue;
store lo_three into 'lo_pig_3col' using PigStorage(' ');
```

Pig 3 Column Extraction Execution (only showing start and finish due to length):



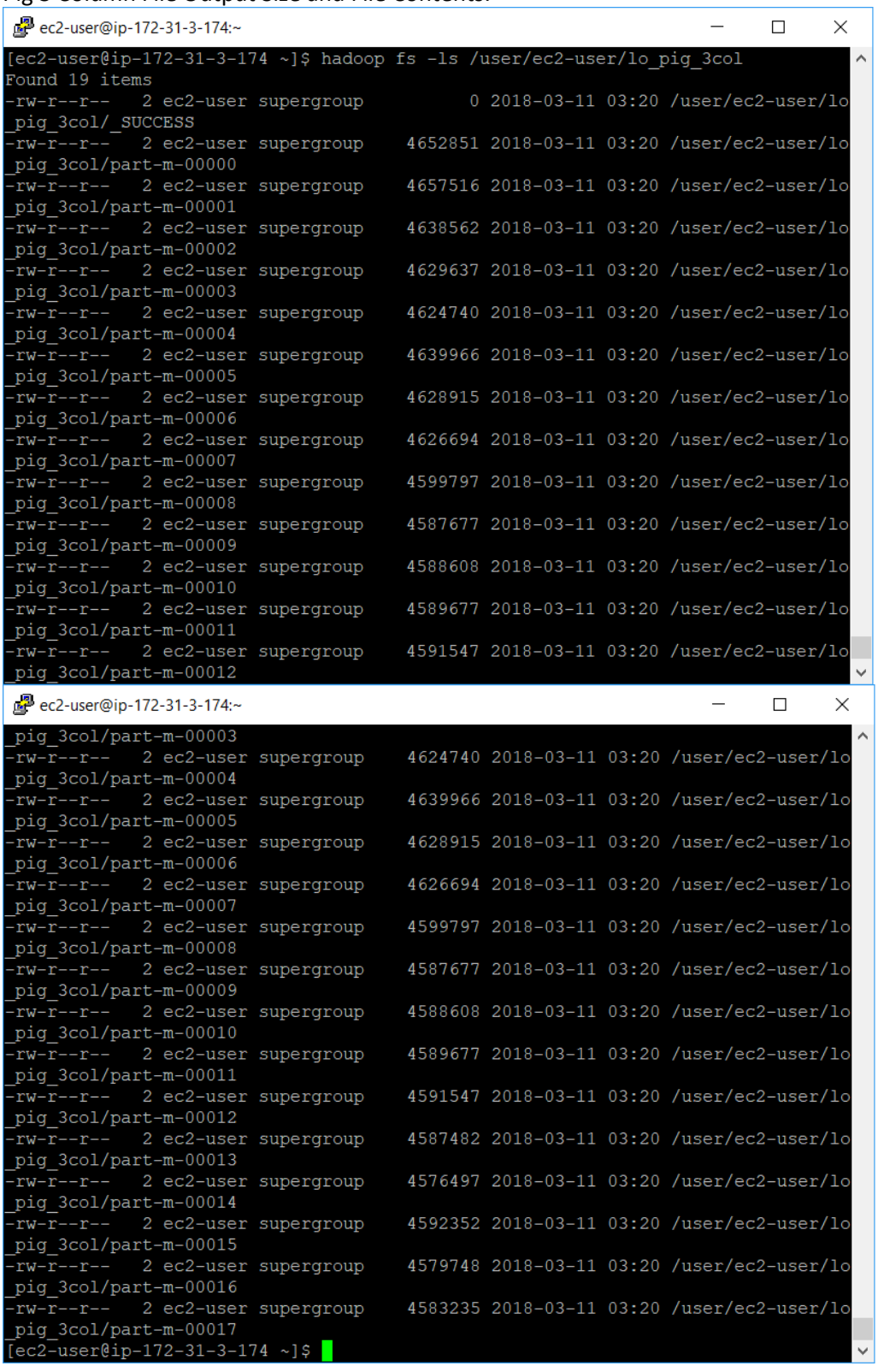

ec2-user@ip-172-31-3-174:~/pig-0.15.0
ne.mapReduceLayer.MapReduceLauncher - Running jobs are [job_1520720845975_0024]
2018-03-11 03:20:20,716 [main] INFO org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - 49% complete
2018-03-11 03:20:20,716 [main] INFO org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Running jobs are [job_1520720845975_0024]
2018-03-11 03:20:23,723 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Con
necting to ResourceManager at /172.31.3.174:8032
2018-03-11 03:20:23,729 [main] INFO org.apache.hadoop.mapred.ClientServiceDeleg
ate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirect
ing to job history server
2018-03-11 03:20:23,891 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Con
necting to ResourceManager at /172.31.3.174:8032
2018-03-11 03:20:23,896 [main] INFO org.apache.hadoop.mapred.ClientServiceDeleg
ate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirect
ing to job history server
2018-03-11 03:20:23,976 [main] INFO org.apache.hadoop.conf.Configuration.deprec
ation - mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces
2018-03-11 03:20:23,977 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Con
necting to ResourceManager at /172.31.3.174:8032
2018-03-11 03:20:23,981 [main] INFO org.apache.hadoop.mapred.ClientServiceDeleg
ate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirect
ing to job history server
2018-03-11 03:20:24,019 [main] INFO org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - 100% complete
2018-03-11 03:20:24,021 [main] INFO org.apache.pig.tools.pigstats.mapreduce.Sim
plePigStats - Script Statistics:
HadoopVersion PigVersion UserId StartedAt FinishedAt Features
2.6.4 0.15.0 ec2-user 2018-03-11 03:19:37 2018-03-11 03:20:24 F
ILTER

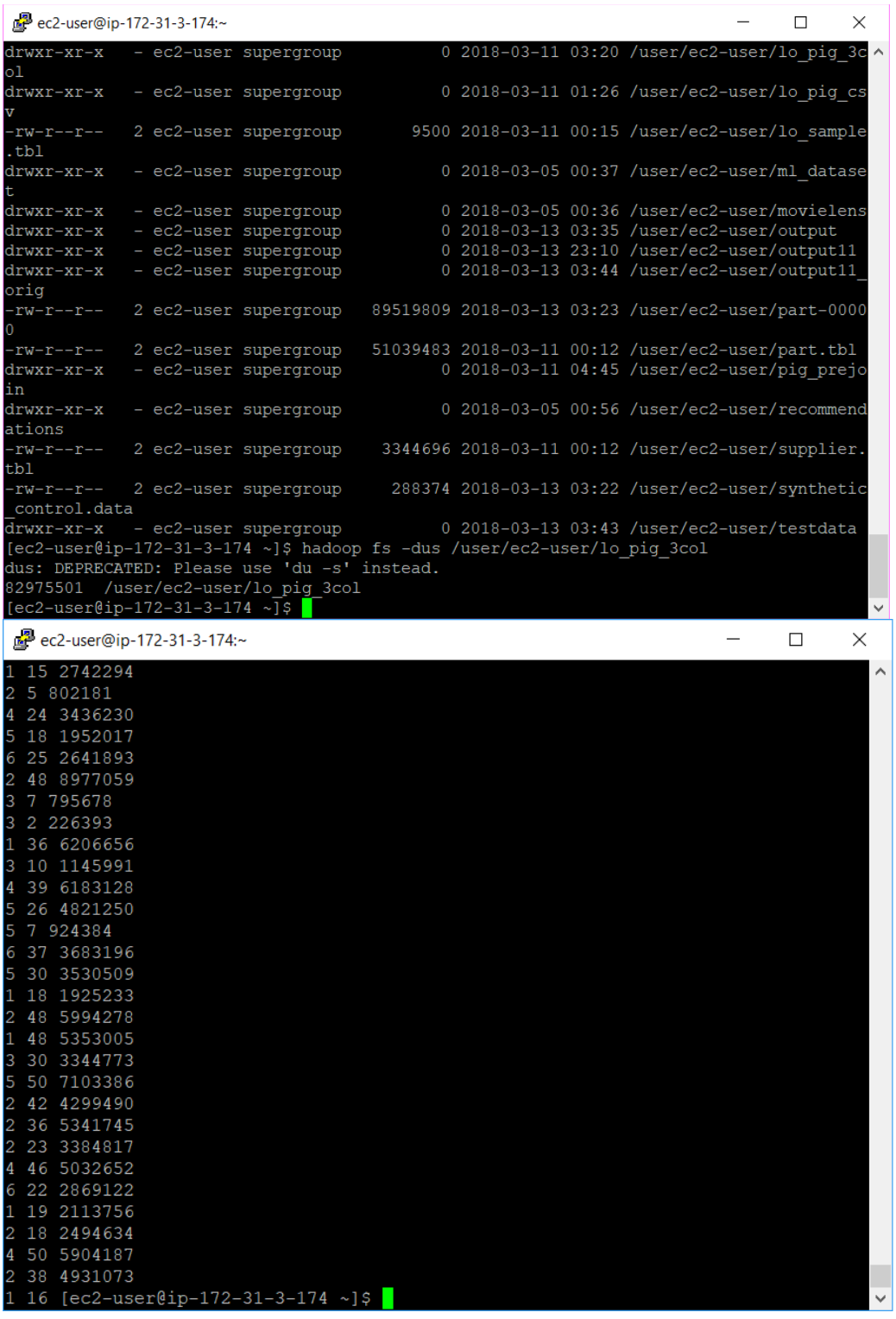

ec2-user@ip-172-31-3-174:~/pig-0.15.0
ILTER
Success!
Job Stats (time in seconds):
JobId Maps Reduces MaxMapTime MinMapTime AvgMapTime MedianMa
pTime MaxReduceTime MinReduceTime AvgReduceTime MedianReducetime A
lias Feature Outputs
job_1520720845975_0024 18 0 36 23 30 28 0 0
0 0 lineorder,lo_filter,lo_three MAP_ONLY hdfs://172.31.3.
174/user/ec2-user/lo_pig_3col,
Input(s):
Successfully read 23996604 records (2417832927 bytes) from: "/user/ec2-user/line
order.tbl"
Output(s):
Successfully stored 6544308 records (82975501 bytes) in: "hdfs://172.31.3.174/us
er/ec2-user/lo_pig_3col"
Counters:
Total records written : 6544308
Total bytes written : 82975501
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
Job DAG:
job_1520720845975_0024



The Pig transformation code started at 3:19:37 and ended at 3:20:24, so it took 53 secs to execute on the 4-node cluster.

Pig 3 Column File Output Size and File Contents:





The size of the Pig 3 column output directory is 82975501 bytes or approximately 82.976 MB.

Part 2

A)

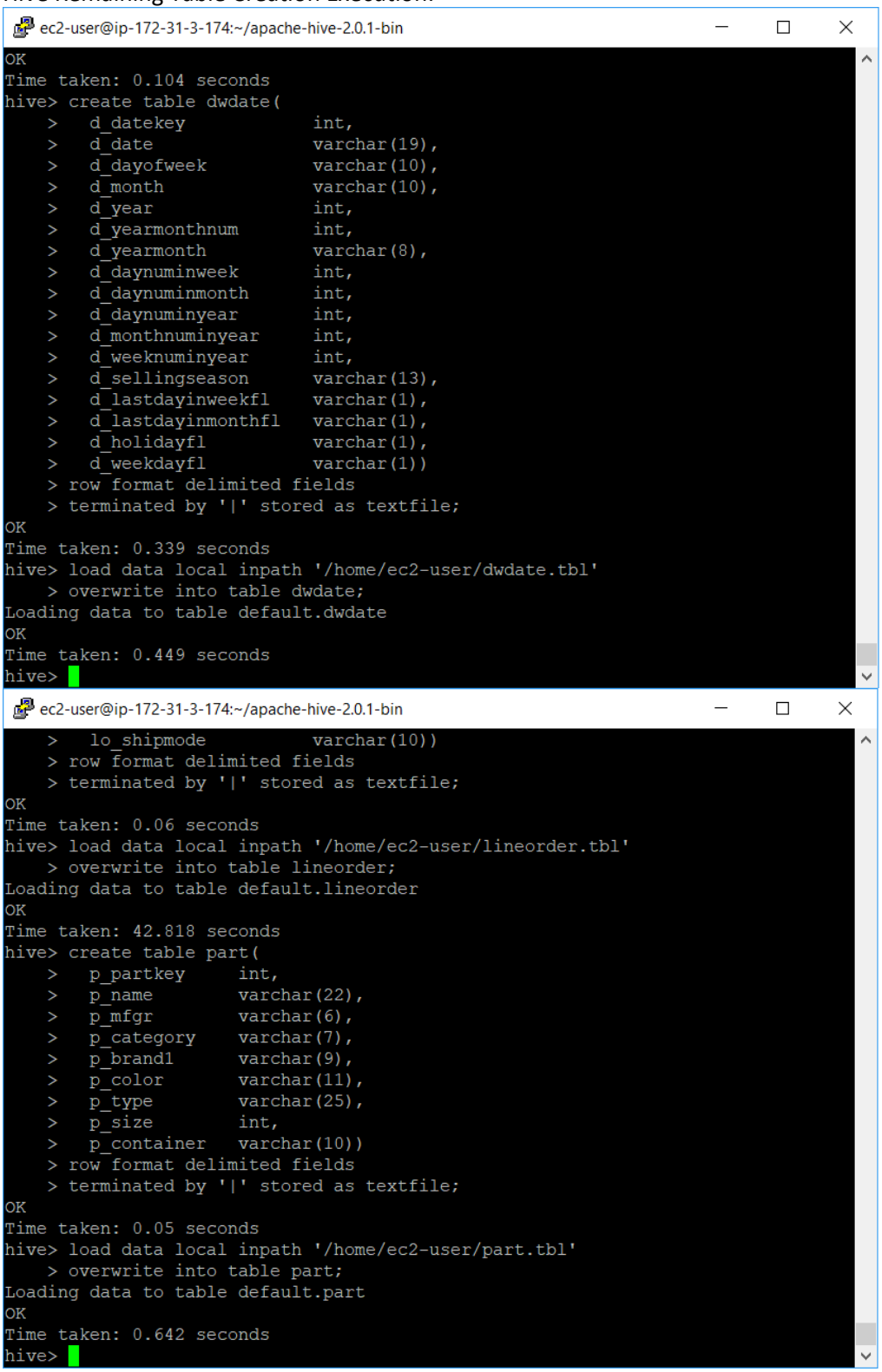
Hive Query 2.1

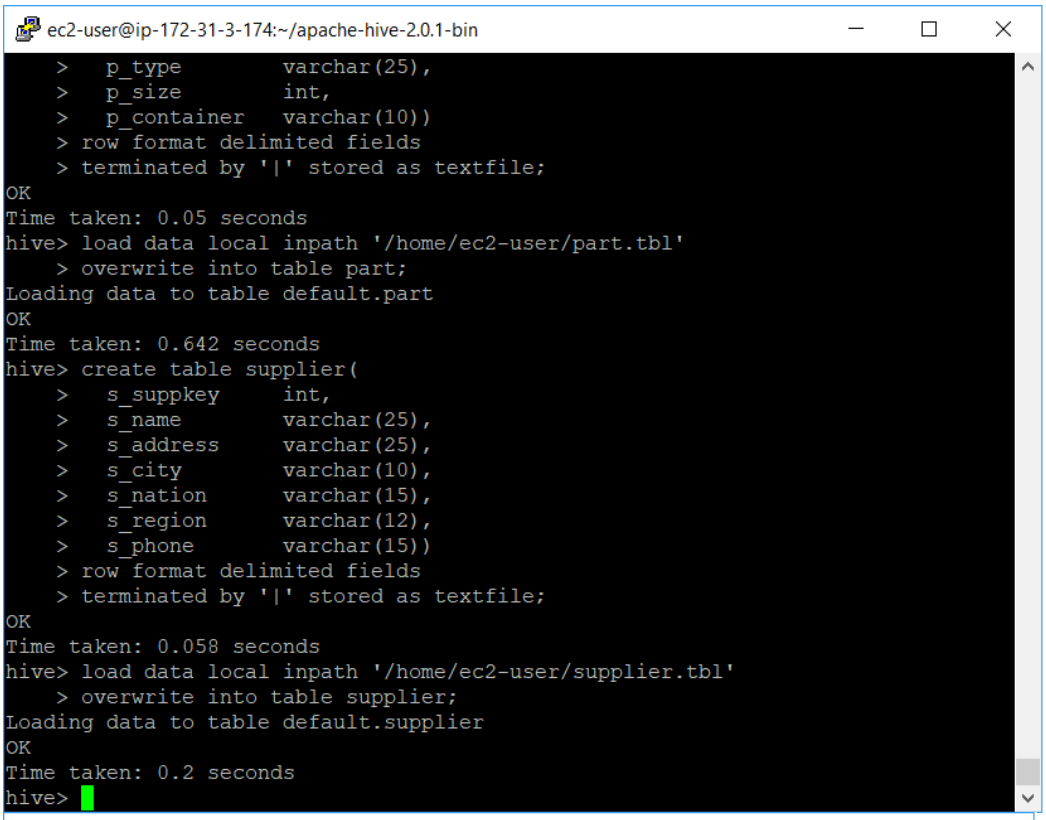
Hive Remaining Table Creation Code:

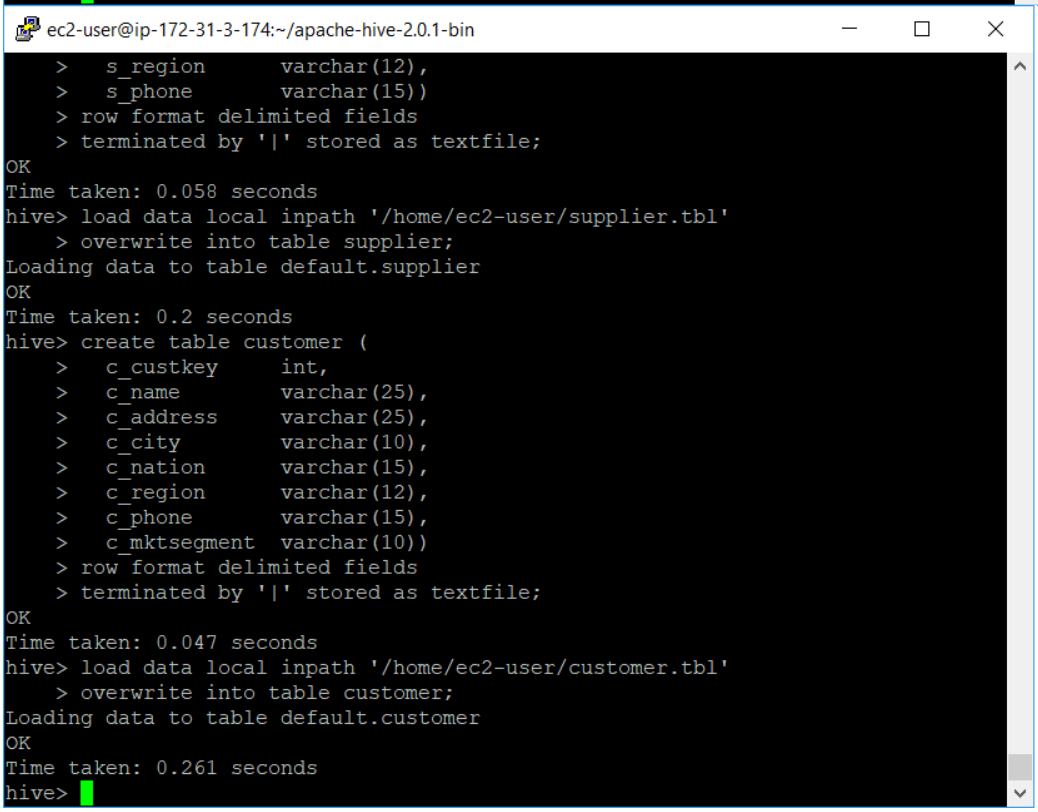
```
create table dwdate(
  d_datekey          int,
  d_date             varchar(19),
  d_dayofweek        varchar(10),
  d_month            varchar(10),
  d_year             int,
  d_yearmonthnum     int,
  d_yearmonth        varchar(8),
  d_daynuminweek     int,
  d_daynuminmonth    int,
  d_daynuminyear     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;


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;
```

```
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;


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;
```

Hive Remaining Table Creation Execution:



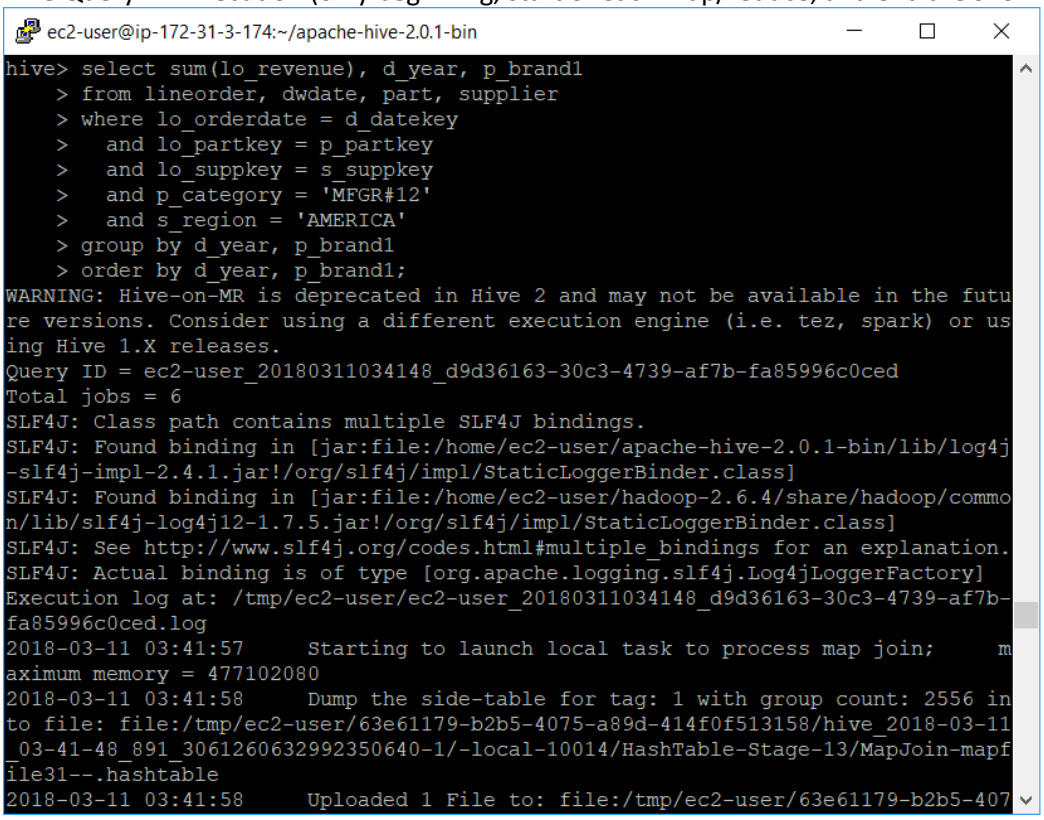

ec2-user@ip-172-31-3-174:~/apache-hive-2.0.1-bin
> p_type varchar(25),
> p_size int,
> p_container varchar(10))
> row format delimited fields
> terminated by '|' stored as textfile;
OK
Time taken: 0.05 seconds
hive> load data local inpath '/home/ec2-user/part.tbl'
> overwrite into table part;
Loading data to table default.part
OK
Time taken: 0.642 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: 0.058 seconds
hive> load data local inpath '/home/ec2-user/supplier.tbl'
> overwrite into table supplier;
Loading data to table default.supplier
OK
Time taken: 0.2 seconds
hive>

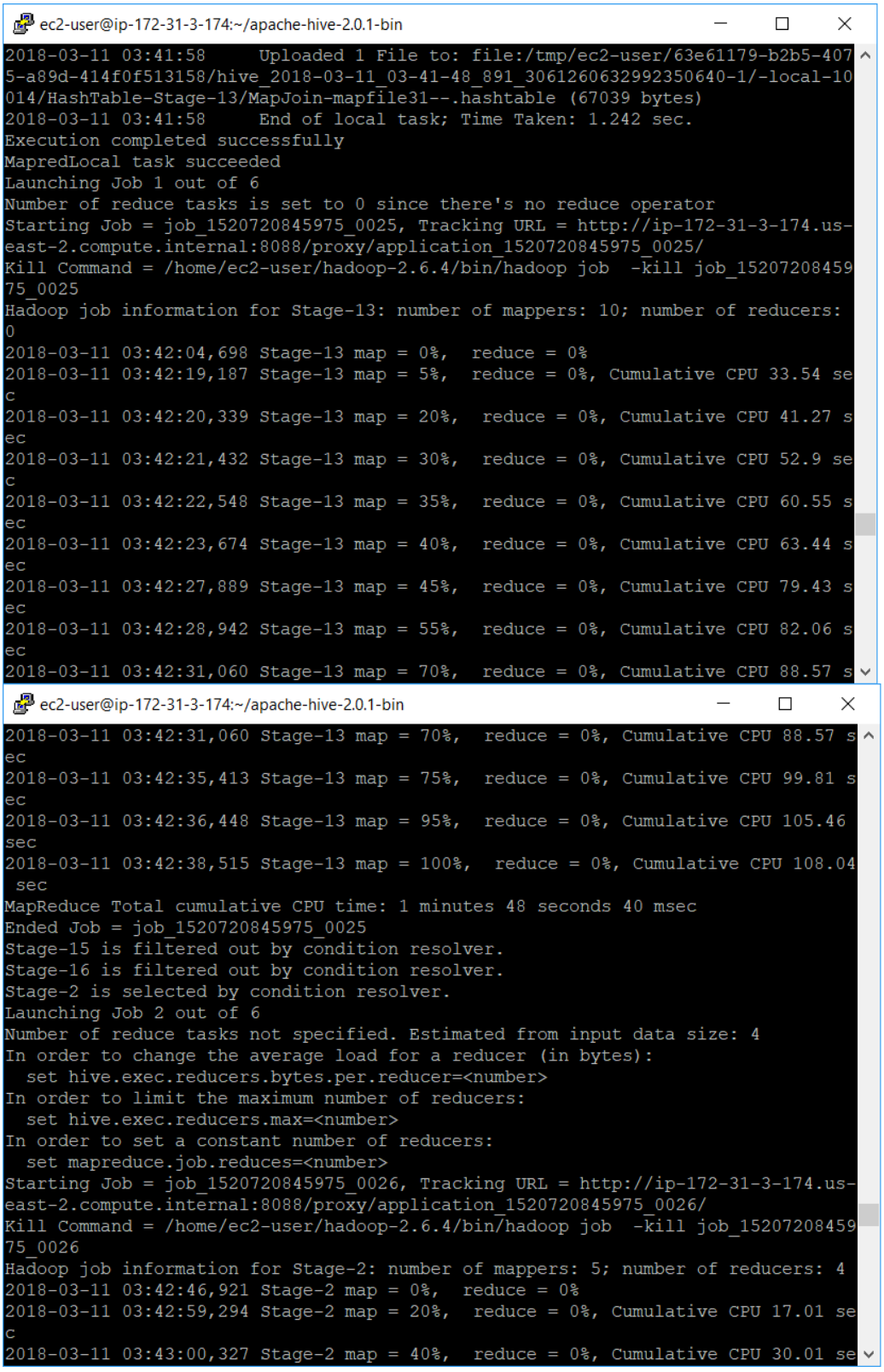

ec2-user@ip-172-31-3-174:~/apache-hive-2.0.1-bin
> s_region varchar(12),
> s_phone varchar(15))
> row format delimited fields
> terminated by '|' stored as textfile;
OK
Time taken: 0.058 seconds
hive> load data local inpath '/home/ec2-user/supplier.tbl'
> overwrite into table supplier;
Loading data to table default.supplier
OK
Time taken: 0.2 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: 0.047 seconds
hive> load data local inpath '/home/ec2-user/customer.tbl'
> overwrite into table customer;
Loading data to table default.customer
OK
Time taken: 0.261 seconds
hive>

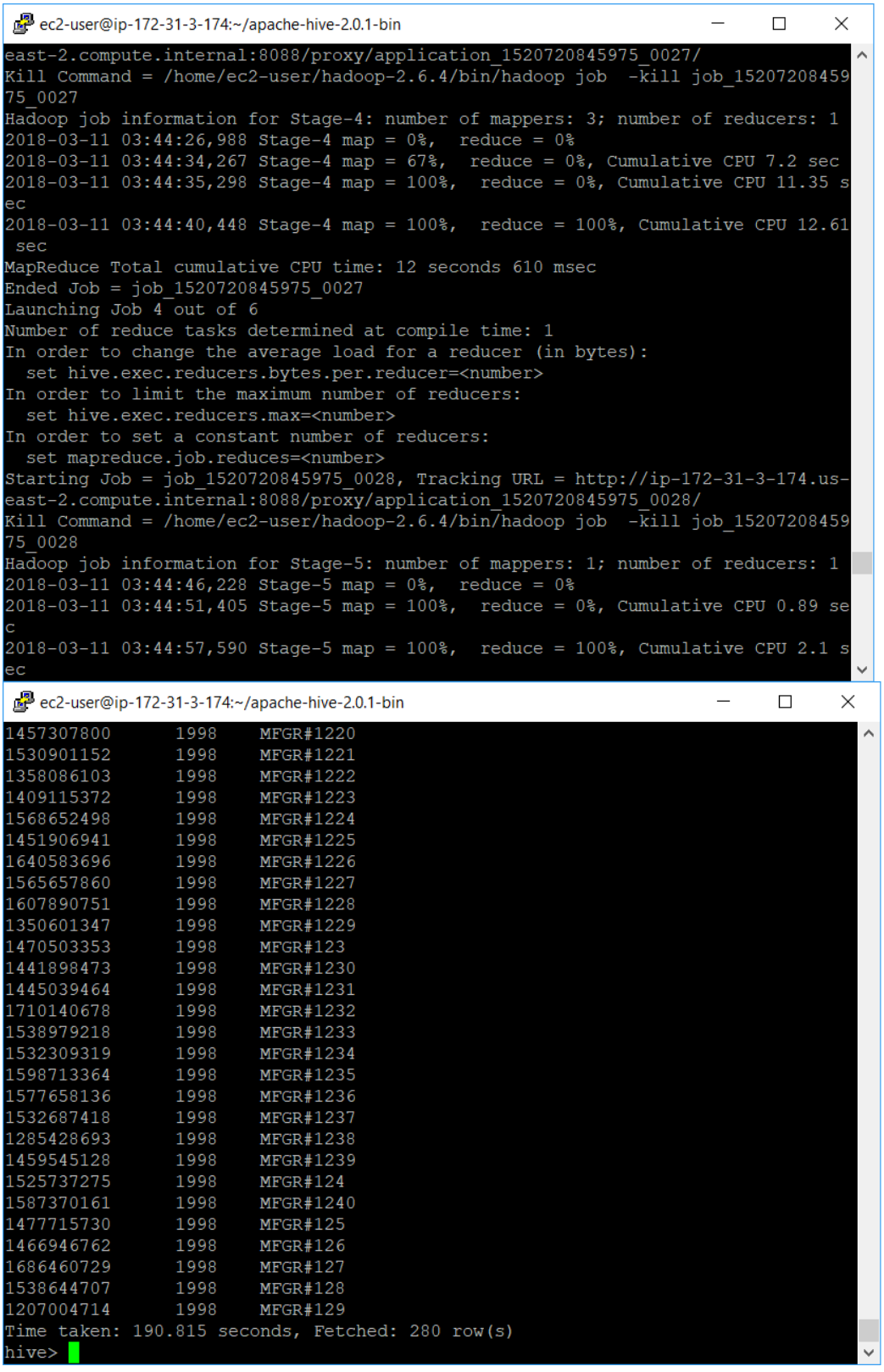
Hive Query 2.1 Code:

```
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 2.1 Execution (only beginning, start of each map/reduce, and end are shown):




ec2-user@ip-172-31-3-174:~/apache-hive-2.0.1-bin
2018-03-11 03:41:58 Uploaded 1 File to: file:/tmp/ec2-user/63e61179-b2b5-407
5-a89d-414f0f513158/hive_2018-03-11_03-41-48_891_3061260632992350640-1/-local-10
014/HashTable-Stage-13/MapJoin-mapfile31--.hashtable (67039 bytes)
2018-03-11 03:41:58 End of local task; Time Taken: 1.242 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 6
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1520720845975_0025, Tracking URL = http://ip-172-31-3-174.us-
east-2.compute.internal:8088/proxy/application_1520720845975_0025/
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job -kill job_15207208459
75_0025
Hadoop job information for Stage-13: number of mappers: 10; number of reducers:
0
2018-03-11 03:42:04,698 Stage-13 map = 0%, reduce = 0%
2018-03-11 03:42:19,187 Stage-13 map = 5%, reduce = 0%, Cumulative CPU 33.54 se
c
2018-03-11 03:42:20,339 Stage-13 map = 20%, reduce = 0%, Cumulative CPU 41.27 s
ec
2018-03-11 03:42:21,432 Stage-13 map = 30%, reduce = 0%, Cumulative CPU 52.9 se
c
2018-03-11 03:42:22,548 Stage-13 map = 35%, reduce = 0%, Cumulative CPU 60.55 s
ec
2018-03-11 03:42:23,674 Stage-13 map = 40%, reduce = 0%, Cumulative CPU 63.44 s
ec
2018-03-11 03:42:27,889 Stage-13 map = 45%, reduce = 0%, Cumulative CPU 79.43 s
ec
2018-03-11 03:42:28,942 Stage-13 map = 55%, reduce = 0%, Cumulative CPU 82.06 s
ec
2018-03-11 03:42:31,060 Stage-13 map = 70%, reduce = 0%, Cumulative CPU 88.57 s
ec
2018-03-11 03:42:35,413 Stage-13 map = 75%, reduce = 0%, Cumulative CPU 99.81 s
ec
2018-03-11 03:42:36,448 Stage-13 map = 95%, reduce = 0%, Cumulative CPU 105.46
sec
2018-03-11 03:42:38,515 Stage-13 map = 100%, reduce = 0%, Cumulative CPU 108.04
sec
MapReduce Total cumulative CPU time: 1 minutes 48 seconds 40 msec
Ended Job = job_1520720845975_0025
Stage-15 is filtered out by condition resolver.
Stage-16 is filtered out by condition resolver.
Stage-2 is selected by condition resolver.
Launching Job 2 out of 6
Number of reduce tasks not specified. Estimated from input data size: 4
In order to change the average load for a reducer (in bytes):
set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
set mapreduce.job.reduces=<number>
Starting Job = job_1520720845975_0026, Tracking URL = http://ip-172-31-3-174.us-
east-2.compute.internal:8088/proxy/application_1520720845975_0026/
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job -kill job_15207208459
75_0026
Hadoop job information for Stage-2: number of mappers: 5; number of reducers: 4
2018-03-11 03:42:46,921 Stage-2 map = 0%, reduce = 0%
2018-03-11 03:42:59,294 Stage-2 map = 20%, reduce = 0%, Cumulative CPU 17.01 se
c
2018-03-11 03:43:00,327 Stage-2 map = 40%, reduce = 0%, Cumulative CPU 30.01 se



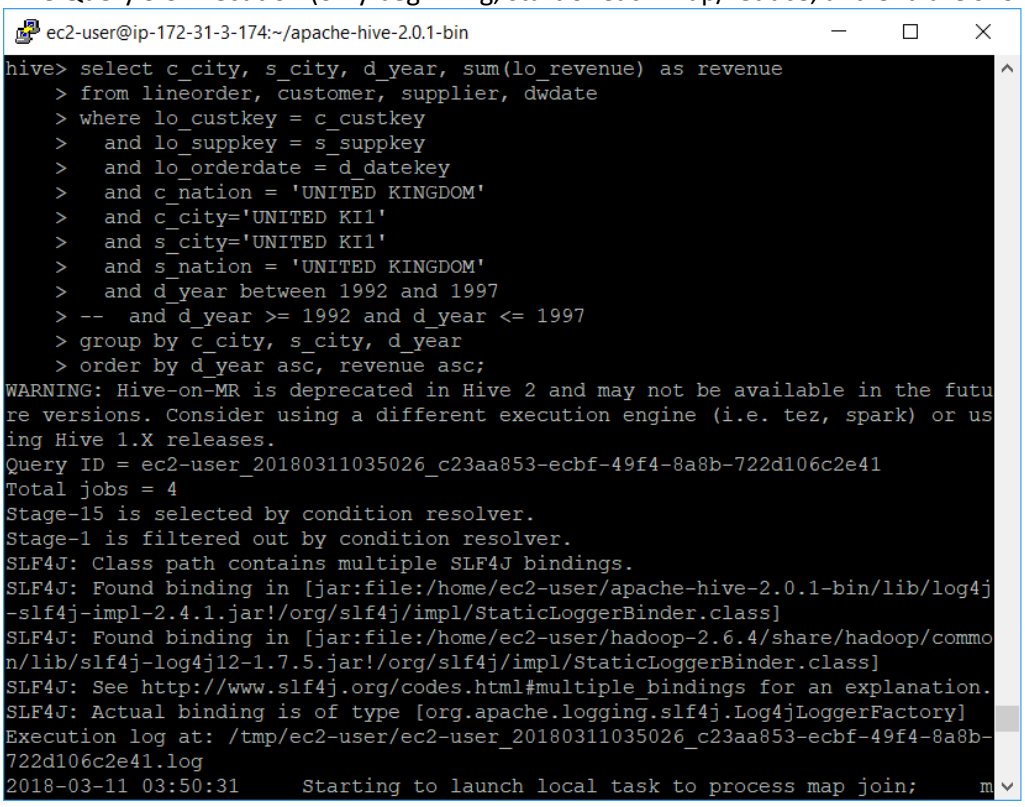
The Hive Query 2.1 took 190.815 secs to run on a 4-node cluster. Note that 4 Map/Reduce passes were executed.

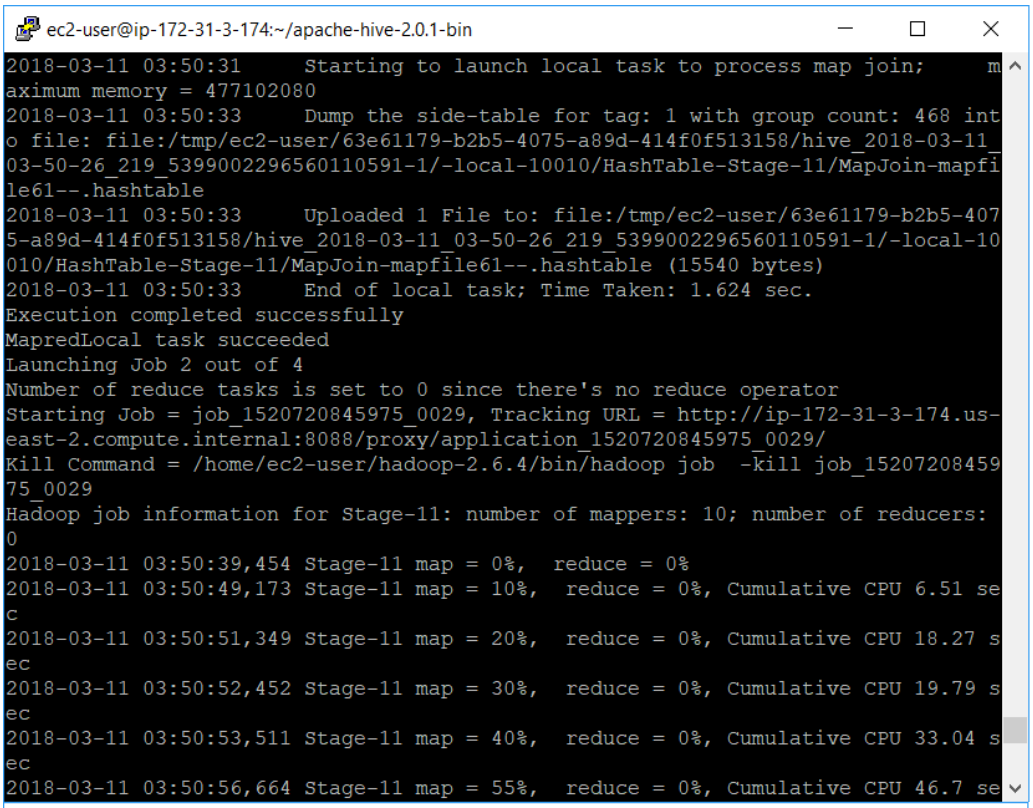
Hive Query 3.3

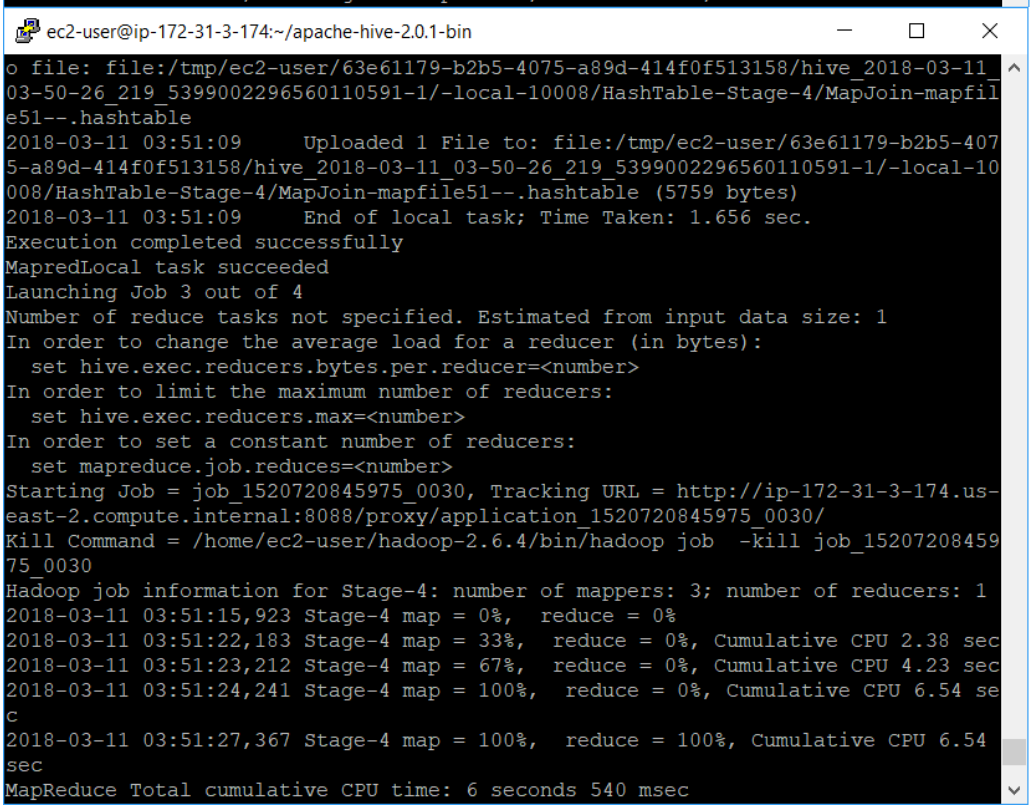
Hive Query 3.3 Code:

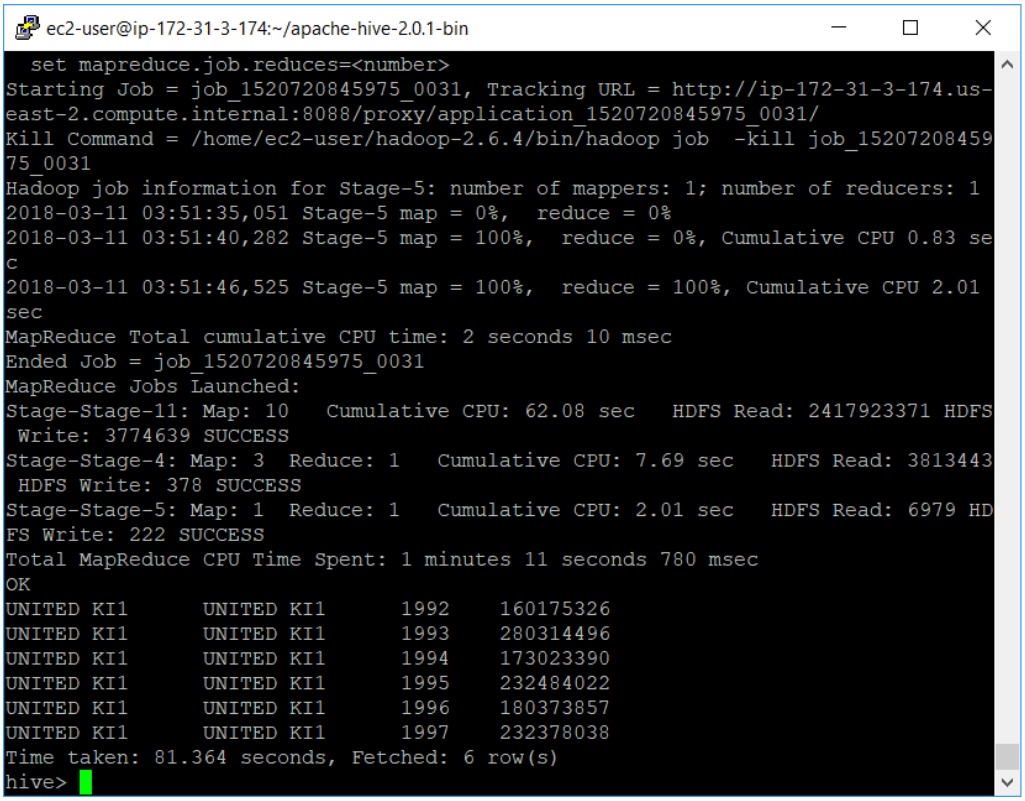
```
select c_city, s_city, d_year, sum(lo_revenue) as revenue
from lineorder, customer, supplier, dwdate
where lo_custkey = c_custkey
  and lo_suppkey = s_suppkey
  and lo_orderdate = d_datekey
  and c_nation = 'UNITED KINGDOM'
  and c_city='UNITED KI1'
  and s_city='UNITED KI1'
  and s_nation = 'UNITED KINGDOM'
  and d_year between 1992 and 1997
--  and d_year >= 1992 and d_year <= 1997
group by c_city, s_city, d_year
order by d_year asc, revenue asc;
```

Hive Query 3.3 Execution (only beginning, start of each map/reduce, and end are shown):




ec2-user@ip-172-31-3-174:~/apache-hive-2.0.1-bin
2018-03-11 03:50:31 Starting to launch local task to process map join; m
aximum memory = 477102080
2018-03-11 03:50:33 Dump the side-table for tag: 1 with group count: 468 int
o file: file:/tmp/ec2-user/63e61179-b2b5-4075-a89d-414f0f513158/hive_2018-03-11_
03-50-26_219_5399002296560110591-1/-local-10010/HashTable-Stage-11/MapJoin-mapfi
le61--.hashtable
2018-03-11 03:50:33 Uploaded 1 File to: file:/tmp/ec2-user/63e61179-b2b5-407
5-a89d-414f0f513158/hive_2018-03-11_03-50-26_219_5399002296560110591-1/-local-10
010/HashTable-Stage-11/MapJoin-mapfile61--.hashtable (15540 bytes)
2018-03-11 03:50:33 End of local task; Time Taken: 1.624 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 2 out of 4
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1520720845975_0029, Tracking URL = http://ip-172-31-3-174.us-
east-2.compute.internal:8088/proxy/application_1520720845975_0029/
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job -kill job_15207208459
75_0029
Hadoop job information for Stage-11: number of mappers: 10; number of reducers:
0
2018-03-11 03:50:39,454 Stage-11 map = 0%, reduce = 0%
2018-03-11 03:50:49,173 Stage-11 map = 10%, reduce = 0%, Cumulative CPU 6.51 se
c
2018-03-11 03:50:51,349 Stage-11 map = 20%, reduce = 0%, Cumulative CPU 18.27 s
ec
2018-03-11 03:50:52,452 Stage-11 map = 30%, reduce = 0%, Cumulative CPU 19.79 s
ec
2018-03-11 03:50:53,511 Stage-11 map = 40%, reduce = 0%, Cumulative CPU 33.04 s
ec
2018-03-11 03:50:56,664 Stage-11 map = 55%, reduce = 0%, Cumulative CPU 46.7 se


ec2-user@ip-172-31-3-174:~/apache-hive-2.0.1-bin
o file: file:/tmp/ec2-user/63e61179-b2b5-4075-a89d-414f0f513158/hive_2018-03-11_
03-50-26_219_5399002296560110591-1/-local-10008/HashTable-Stage-4/MapJoin-mapfil
e51--.hashtable
2018-03-11 03:51:09 Uploaded 1 File to: file:/tmp/ec2-user/63e61179-b2b5-407
5-a89d-414f0f513158/hive_2018-03-11_03-50-26_219_5399002296560110591-1/-local-10
008/HashTable-Stage-4/MapJoin-mapfile51--.hashtable (5759 bytes)
2018-03-11 03:51:09 End of local task; Time Taken: 1.656 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 3 out of 4
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
set mapreduce.job.reduces=<number>
Starting Job = job_1520720845975_0030, Tracking URL = http://ip-172-31-3-174.us-
east-2.compute.internal:8088/proxy/application_1520720845975_0030/
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job -kill job_15207208459
75_0030
Hadoop job information for Stage-4: number of mappers: 3; number of reducers: 1
2018-03-11 03:51:15,923 Stage-4 map = 0%, reduce = 0%
2018-03-11 03:51:22,183 Stage-4 map = 33%, reduce = 0%, Cumulative CPU 2.38 sec
2018-03-11 03:51:23,212 Stage-4 map = 67%, reduce = 0%, Cumulative CPU 4.23 sec
2018-03-11 03:51:24,241 Stage-4 map = 100%, reduce = 0%, Cumulative CPU 6.54 se
c
2018-03-11 03:51:27,367 Stage-4 map = 100%, reduce = 100%, Cumulative CPU 6.54
sec
MapReduce Total cumulative CPU time: 6 seconds 540 msec



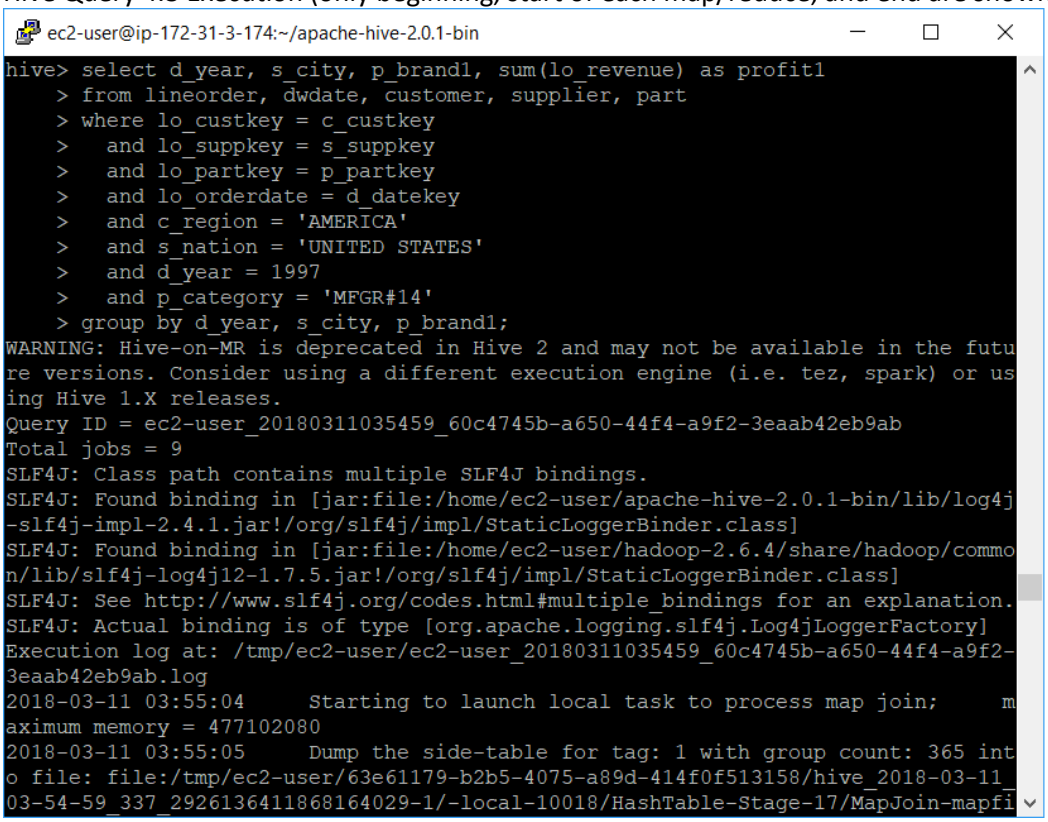
The Hive Query 3.3 took 81.364 secs to run on a 4-node cluster. Note that 2 Map/Reduce passes were executed.

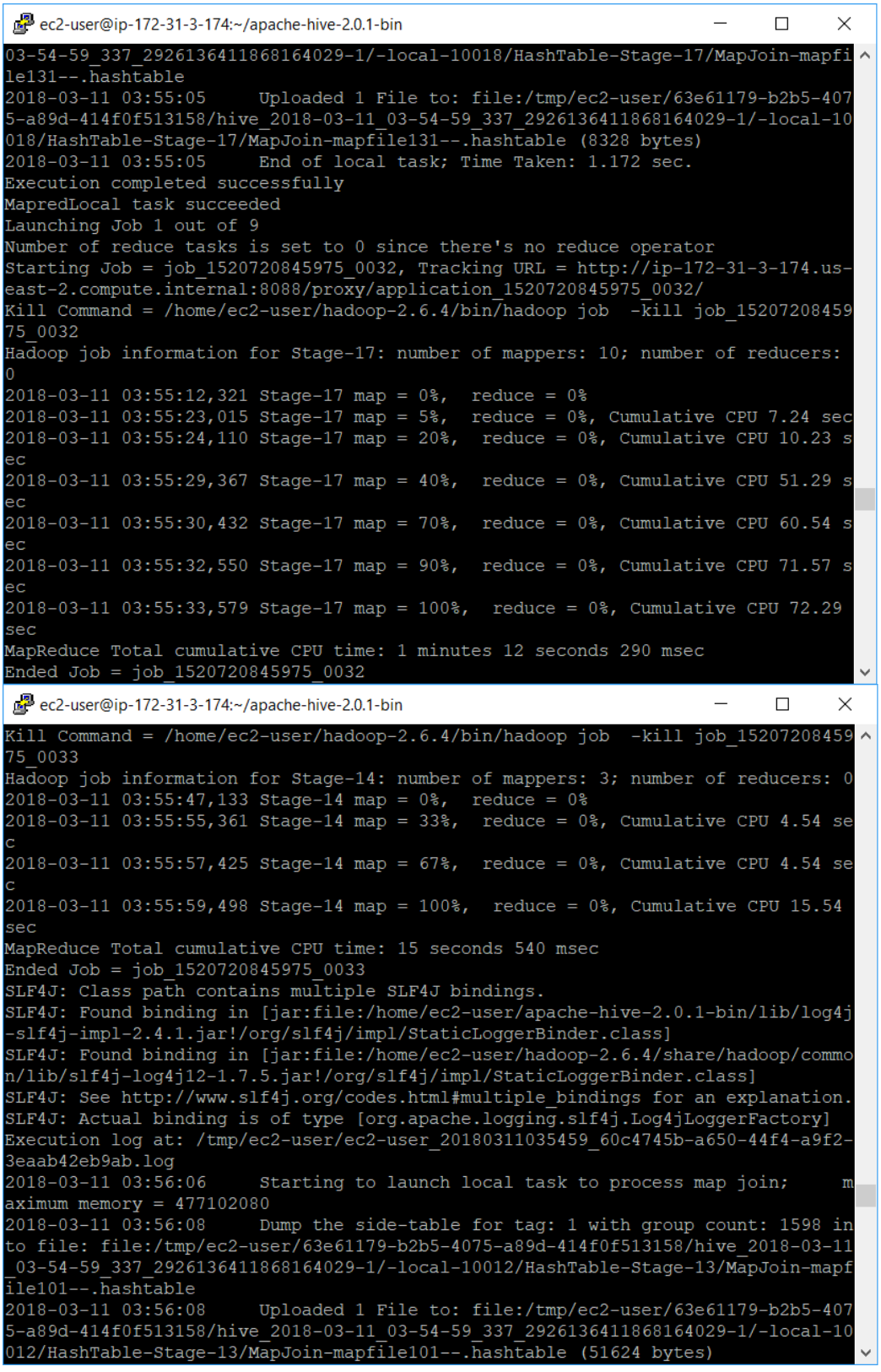
Hive Query 4.3

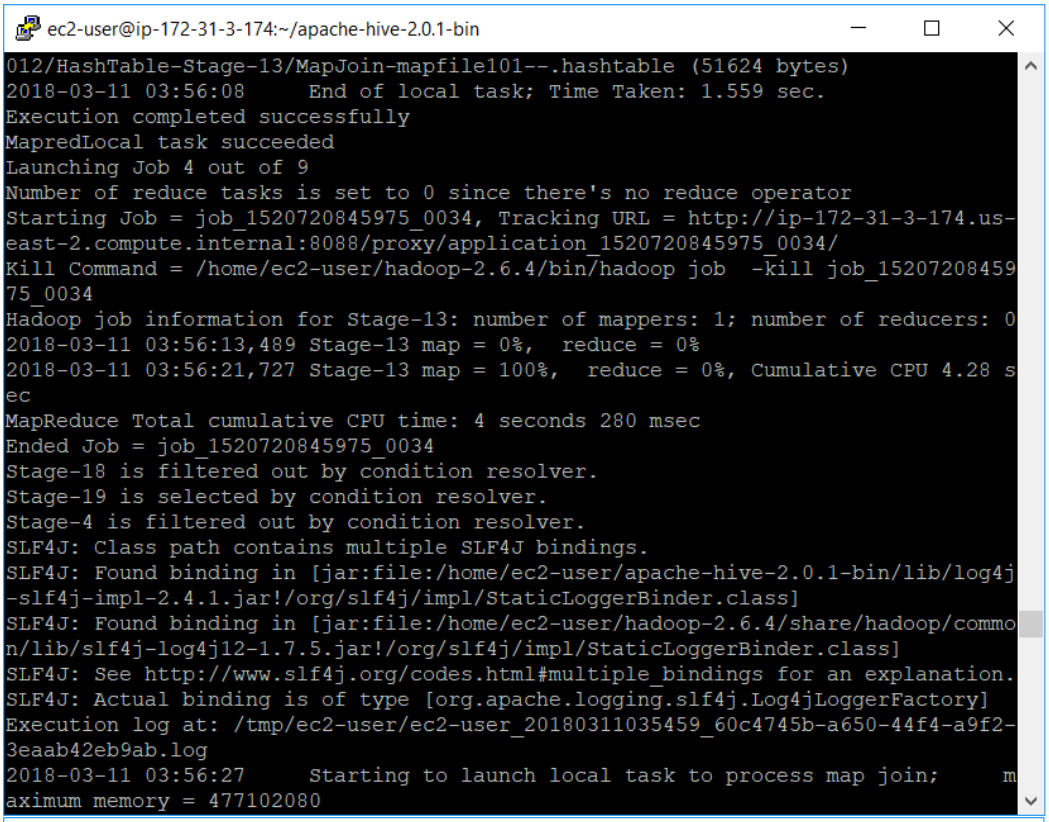
Hive Query 4.3 Code:

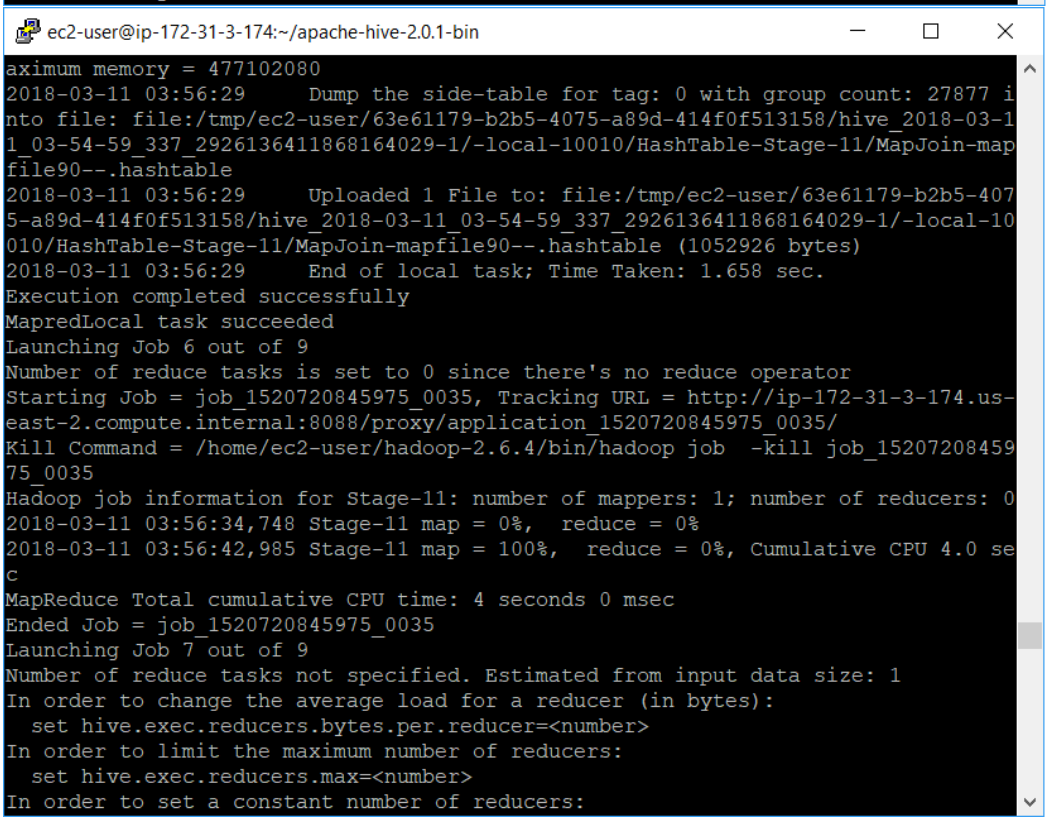
```
select d_year, s_city, p_brand1, sum(lo_revenue) as profit1
from lineorder, dwdate, customer, supplier, part
where lo_custkey = c_custkey
  and lo_suppkey = s_suppkey
  and lo_partkey = p_partkey
  and lo_orderdate = d_datekey
  and c_region = 'AMERICA'
  and s_nation = 'UNITED STATES'
  and d_year = 1997
  and p_category = 'MFGR#14'
group by d_year, s_city, p_brand1;
```

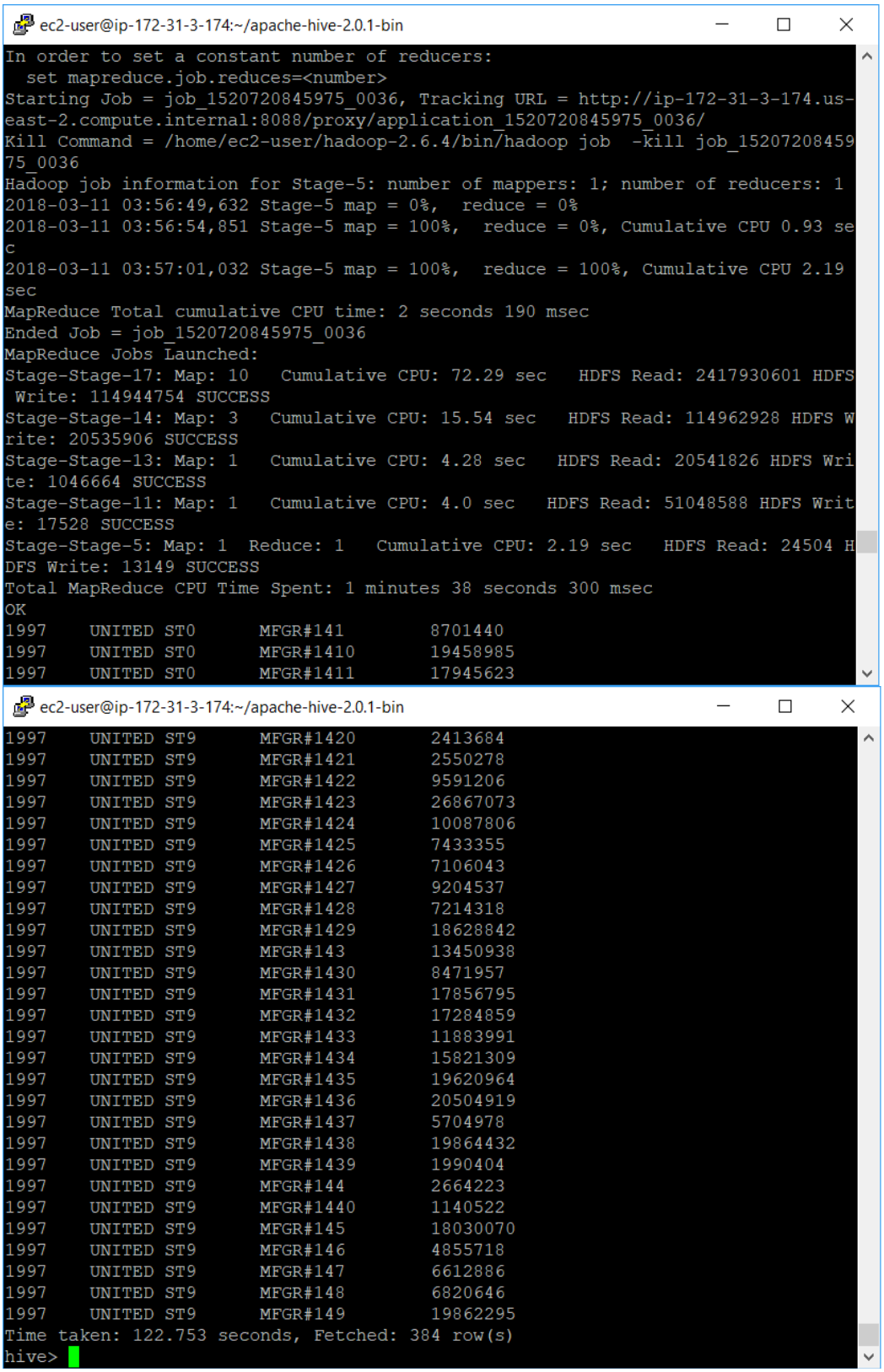
Hive Query 4.3 Execution (only beginning, start of each map/reduce, and end are shown):




ec2-user@ip-172-31-3-174:~/apache-hive-2.0.1-bin
03-54-59_337_2926136411868164029-1/-local-10018/HashTable-Stage-17/MapJoin-mapfile131--.hashtable
2018-03-11 03:55:05 Uploaded 1 File to: file:/tmp/ec2-user/63e61179-b2b5-4075-a89d-414f0f513158/hive_2018-03-11_03-54-59_337_2926136411868164029-1/-local-10018/HashTable-Stage-17/MapJoin-mapfile131--.hashtable (8328 bytes)
2018-03-11 03:55:05 End of local task; Time Taken: 1.172 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 9
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1520720845975_0032, Tracking URL = http://ip-172-31-3-174.us-east-2.compute.internal:8088/proxy/application_1520720845975_0032/
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job -kill job_1520720845975_0032
Hadoop job information for Stage-17: number of mappers: 10; number of reducers: 0
2018-03-11 03:55:12,321 Stage-17 map = 0%, reduce = 0%
2018-03-11 03:55:23,015 Stage-17 map = 5%, reduce = 0%, Cumulative CPU 7.24 sec
2018-03-11 03:55:24,110 Stage-17 map = 20%, reduce = 0%, Cumulative CPU 10.23 sec
2018-03-11 03:55:29,367 Stage-17 map = 40%, reduce = 0%, Cumulative CPU 51.29 sec
2018-03-11 03:55:30,432 Stage-17 map = 70%, reduce = 0%, Cumulative CPU 60.54 sec
2018-03-11 03:55:32,550 Stage-17 map = 90%, reduce = 0%, Cumulative CPU 71.57 sec
2018-03-11 03:55:33,579 Stage-17 map = 100%, reduce = 0%, Cumulative CPU 72.29 sec
MapReduce Total cumulative CPU time: 1 minutes 12 seconds 290 msec
Ended Job = job_1520720845975_0032
ec2-user@ip-172-31-3-174:~/apache-hive-2.0.1-bin
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job -kill job_1520720845975_0033
Hadoop job information for Stage-14: number of mappers: 3; number of reducers: 0
2018-03-11 03:55:47,133 Stage-14 map = 0%, reduce = 0%
2018-03-11 03:55:55,361 Stage-14 map = 33%, reduce = 0%, Cumulative CPU 4.54 sec
2018-03-11 03:55:57,425 Stage-14 map = 67%, reduce = 0%, Cumulative CPU 4.54 sec
2018-03-11 03:55:59,498 Stage-14 map = 100%, reduce = 0%, Cumulative CPU 15.54 sec
MapReduce Total cumulative CPU time: 15 seconds 540 msec
Ended Job = job_1520720845975_0033
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/ec2-user/apache-hive-2.0.1-bin/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/ec2-user/hadoop-2.6.4/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Execution log at: /tmp/ec2-user/ec2-user_20180311035459_60c4745b-a650-44f4-a9f2-3eaab42eb9ab.log
2018-03-11 03:56:06 Starting to launch local task to process map join; maximum memory = 477102080
2018-03-11 03:56:08 Dump the side-table for tag: 1 with group count: 1598 into file: file:/tmp/ec2-user/63e61179-b2b5-4075-a89d-414f0f513158/hive_2018-03-11_03-54-59_337_2926136411868164029-1/-local-10012/HashTable-Stage-13/MapJoin-mapfile101--.hashtable
2018-03-11 03:56:08 Uploaded 1 File to: file:/tmp/ec2-user/63e61179-b2b5-4075-a89d-414f0f513158/hive_2018-03-11_03-54-59_337_2926136411868164029-1/-local-10012/HashTable-Stage-13/MapJoin-mapfile101--.hashtable (51624 bytes)


ec2-user@ip-172-31-3-174:~/apache-hive-2.0.1-bin
012/HashTable-Stage-13/MapJoin-mapfile101--.hashtable (51624 bytes)
2018-03-11 03:56:08 End of local task; Time Taken: 1.559 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 4 out of 9
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1520720845975_0034, Tracking URL = http://ip-172-31-3-174.us-east-2.compute.internal:8088/proxy/application_1520720845975_0034/
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job -kill job_1520720845975_0034
Hadoop job information for Stage-13: number of mappers: 1; number of reducers: 0
2018-03-11 03:56:13,489 Stage-13 map = 0%, reduce = 0%
2018-03-11 03:56:21,727 Stage-13 map = 100%, reduce = 0%, Cumulative CPU 4.28 sec
MapReduce Total cumulative CPU time: 4 seconds 280 msec
Ended Job = job_1520720845975_0034
Stage-18 is filtered out by condition resolver.
Stage-19 is selected by condition resolver.
Stage-4 is filtered out by condition resolver.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/ec2-user/apache-hive-2.0.1-bin/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/ec2-user/hadoop-2.6.4/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Execution log at: /tmp/ec2-user/ec2-user_20180311035459_60c4745b-a650-44f4-a9f2-3eaab42eb9ab.log
2018-03-11 03:56:27 Starting to launch local task to process map join; m
aximum memory = 477102080


ec2-user@ip-172-31-3-174:~/apache-hive-2.0.1-bin
aximum memory = 477102080
2018-03-11 03:56:29 Dump the side-table for tag: 0 with group count: 27877 into file: file:/tmp/ec2-user/63e61179-b2b5-4075-a89d-414f0f513158/hive_2018-03-11_03-54-59_337_2926136411868164029-1/-local-10010/HashTable-Stage-11/MapJoin-mapfile90--.hashtable
2018-03-11 03:56:29 Uploaded 1 File to: file:/tmp/ec2-user/63e61179-b2b5-4075-a89d-414f0f513158/hive_2018-03-11_03-54-59_337_2926136411868164029-1/-local-10010/HashTable-Stage-11/MapJoin-mapfile90--.hashtable (1052926 bytes)
2018-03-11 03:56:29 End of local task; Time Taken: 1.658 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 6 out of 9
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1520720845975_0035, Tracking URL = http://ip-172-31-3-174.us-east-2.compute.internal:8088/proxy/application_1520720845975_0035/
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job -kill job_1520720845975_0035
Hadoop job information for Stage-11: number of mappers: 1; number of reducers: 0
2018-03-11 03:56:34,748 Stage-11 map = 0%, reduce = 0%
2018-03-11 03:56:42,985 Stage-11 map = 100%, reduce = 0%, Cumulative CPU 4.0 sec
MapReduce Total cumulative CPU time: 4 seconds 0 msec
Ended Job = job_1520720845975_0035
Launching Job 7 out of 9
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:



The Hive Query 4.3 took 122.753 secs to run on a 4-node cluster. Note that 5 Map/Reduce passes were executed.

B)

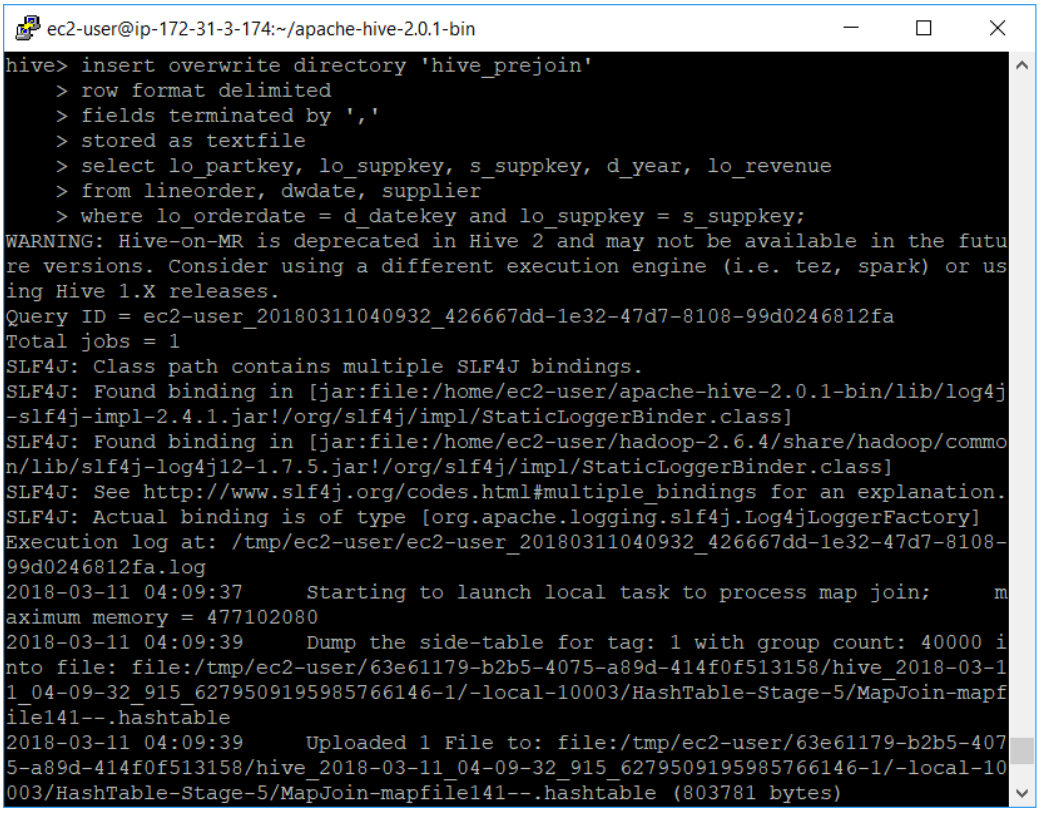
Note that since no delimiter was specified to be used for the pre-join file, I used a comma delimiter.

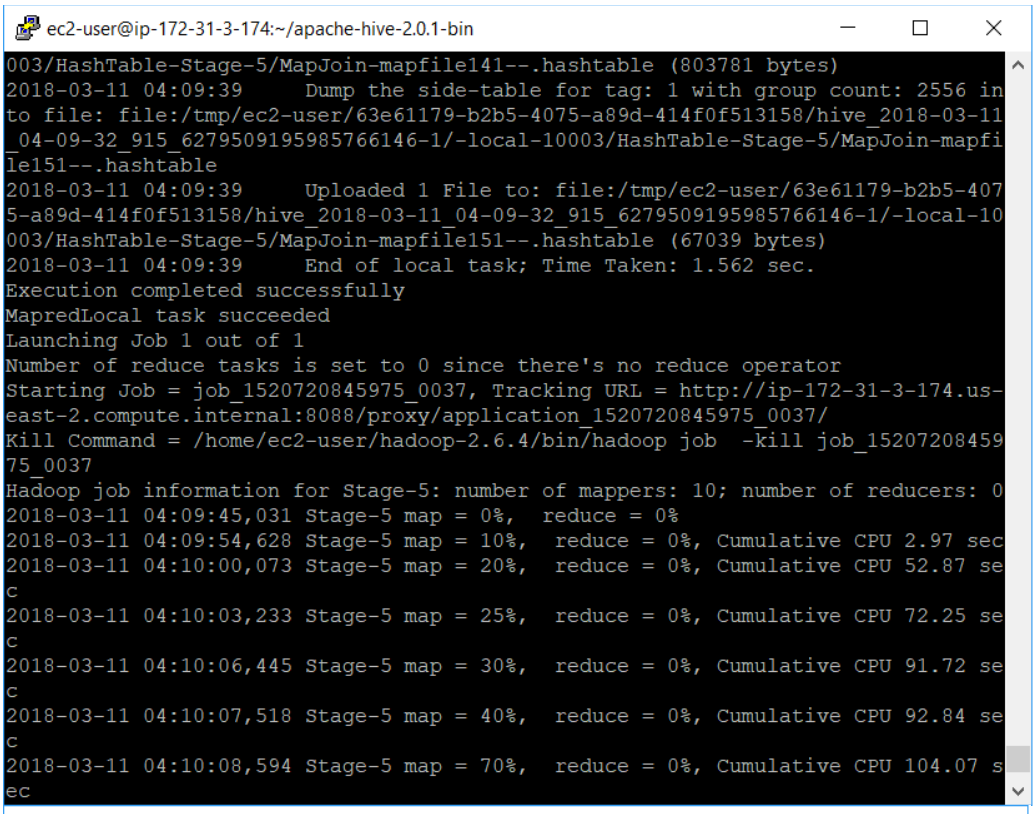
Hive Pre-Join

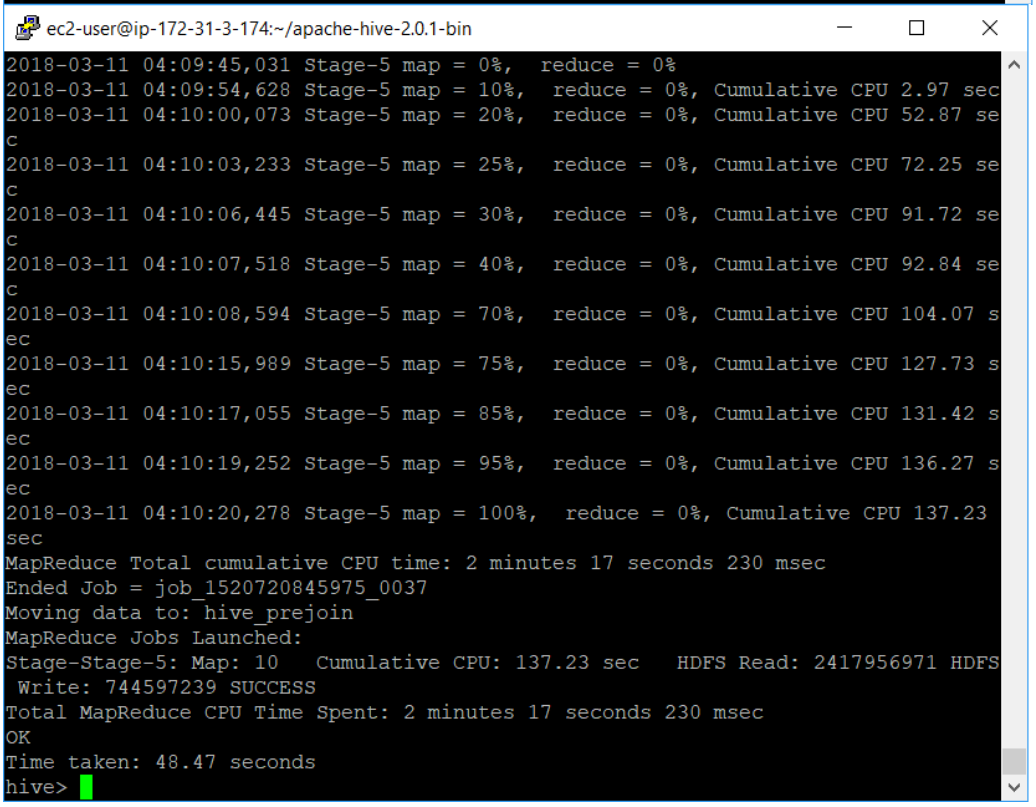
Hive Pre-Join Code:

```
insert overwrite directory 'hive_prejoin'
row format delimited
fields terminated by ','
stored as textfile
select lo_partkey, lo_suppkey, s_suppkey, d_year, lo_revenue
from lineorder, dwdate, supplier
where lo_orderdate = d_datekey and lo_suppkey = s_suppkey;
```

Hive Pre-Join Execution:

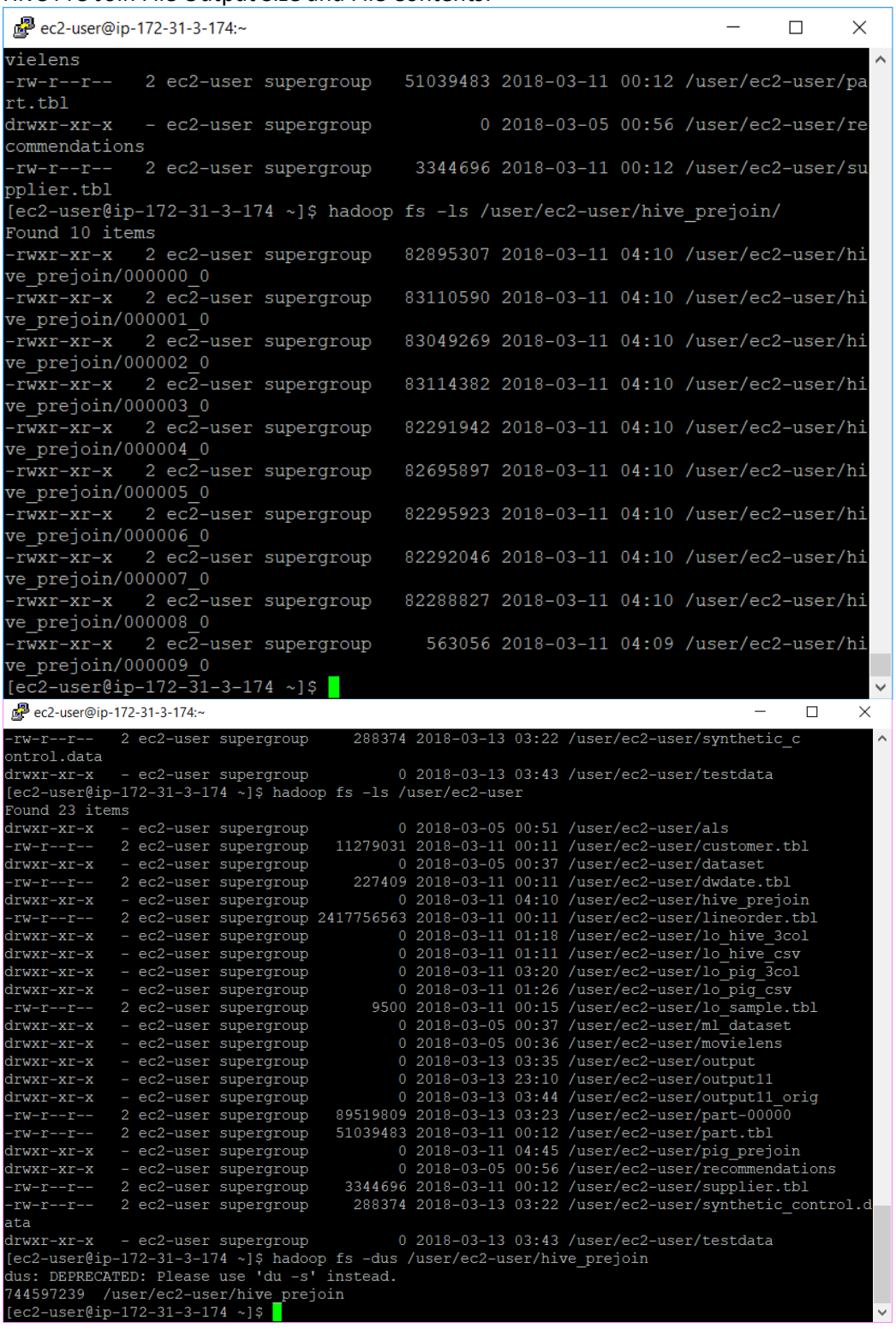


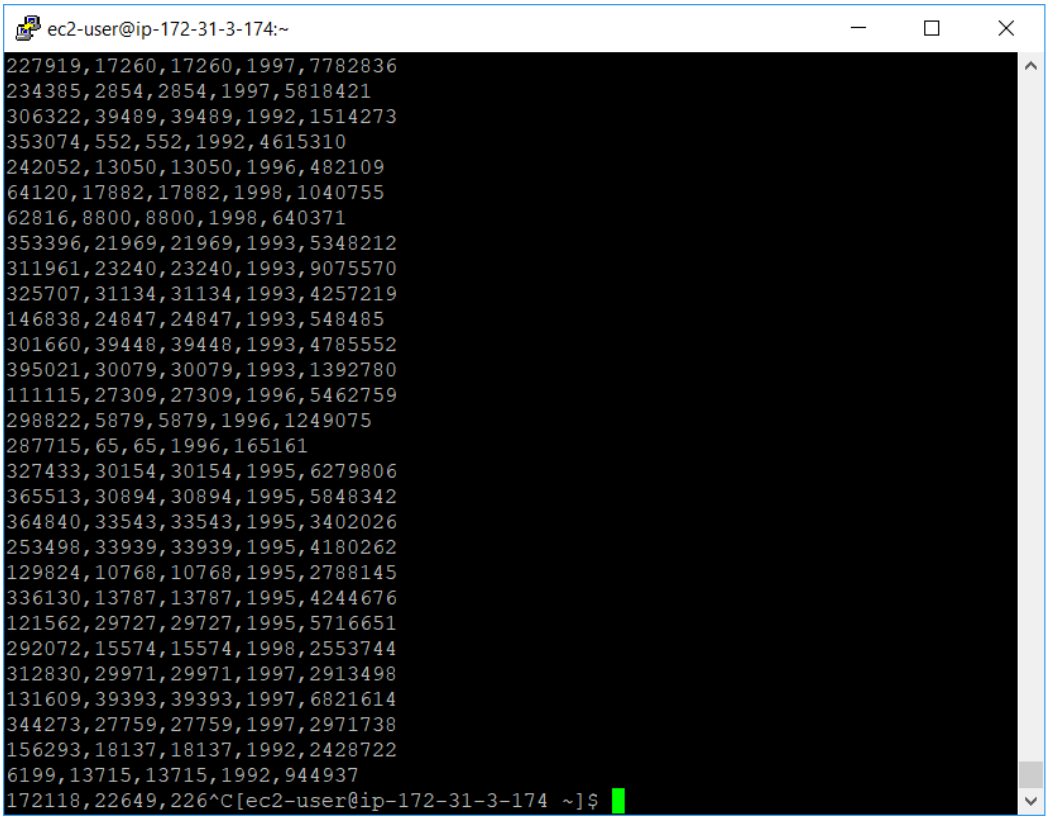




The Hive Pre-Join code took 48.47 secs to execute on a 4-node cluster.

Hive Pre-Join File Output Size and File Contents:





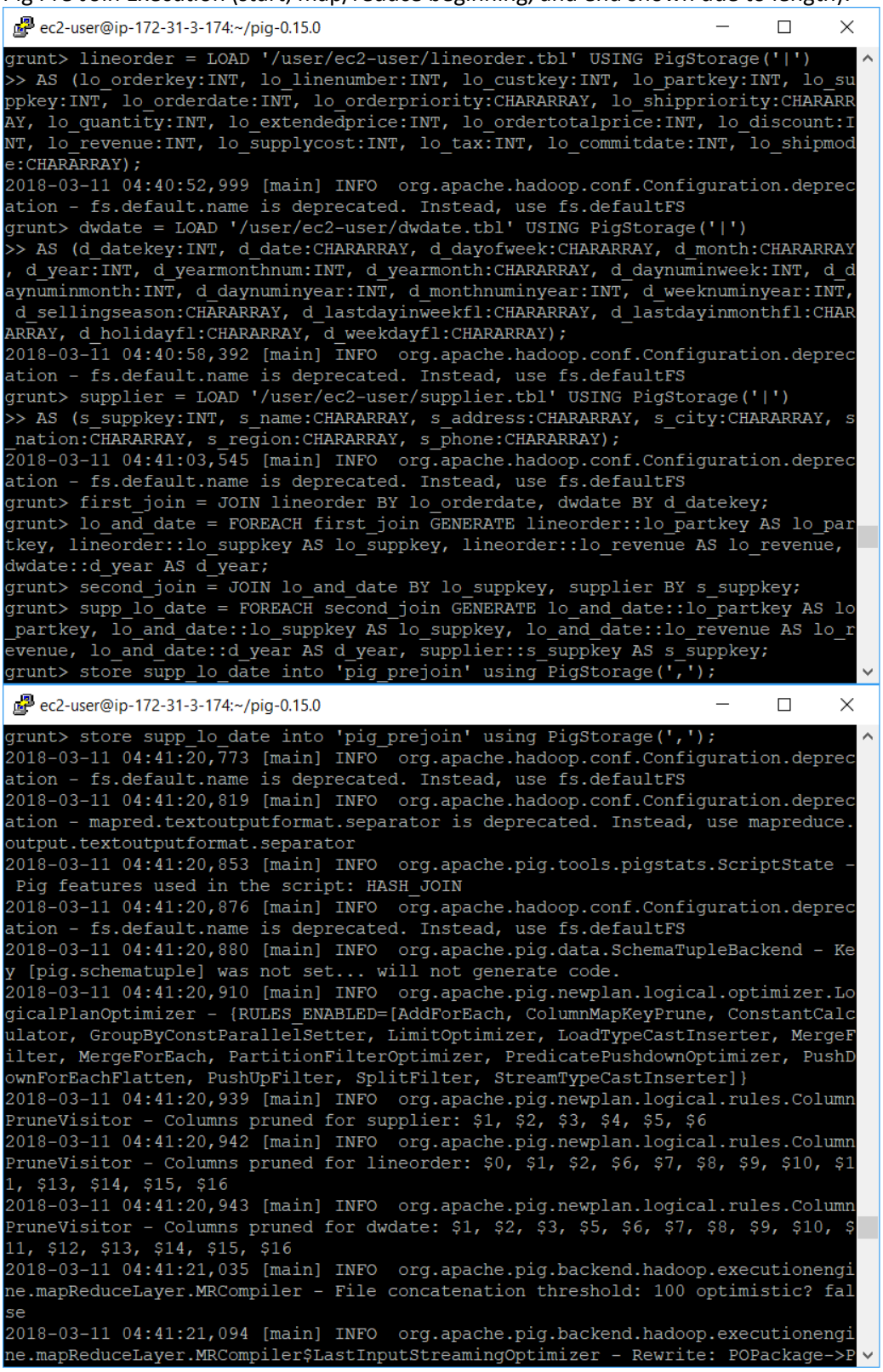
The size of the Hive pre-join output directory is 744597239 bytes or approximately 744,597 MB.

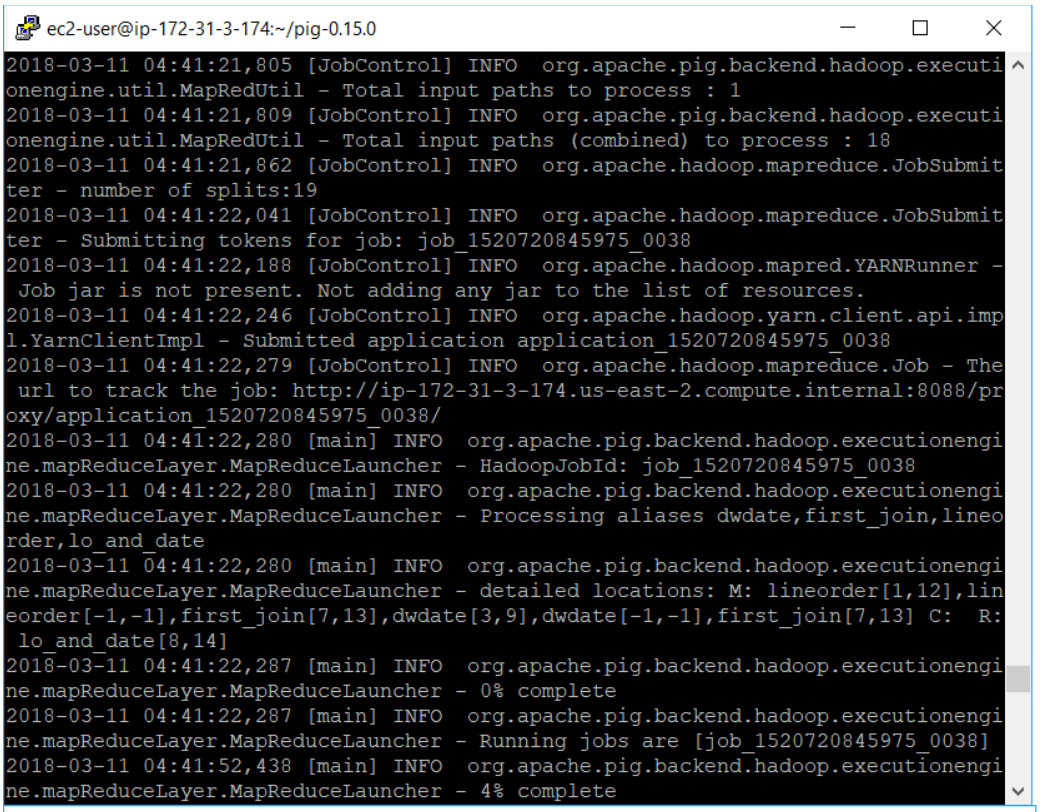
Pig Pre-Join

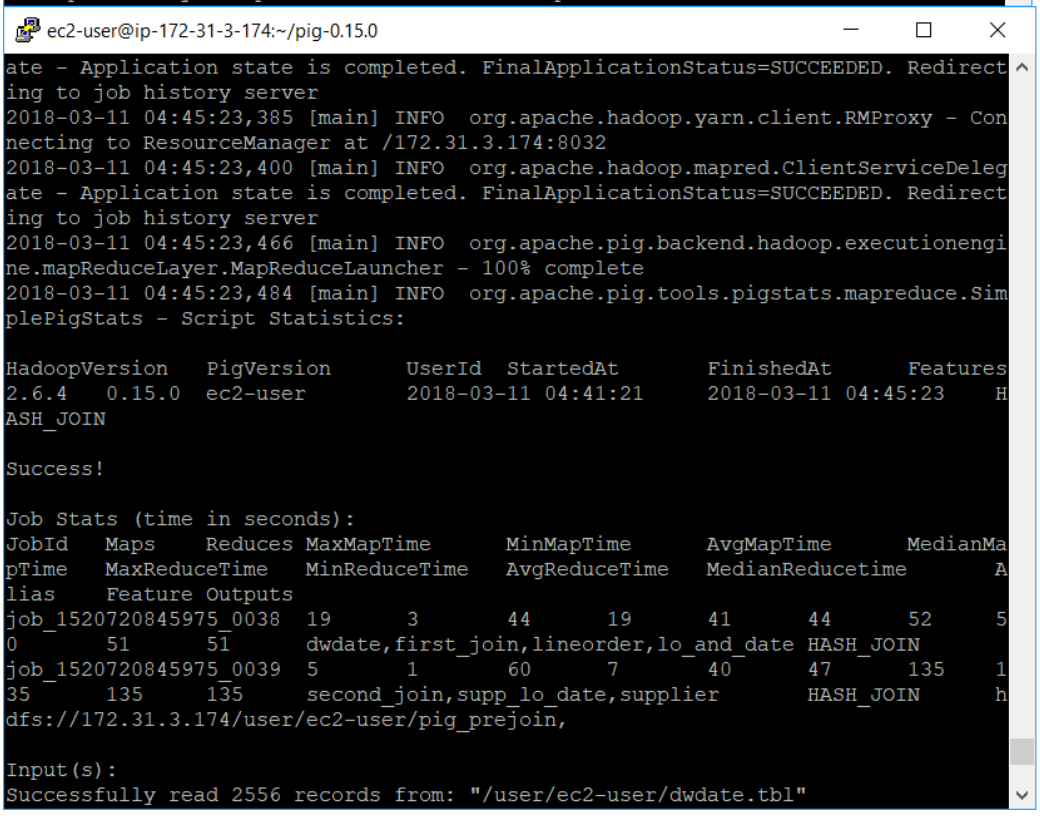
Pig Pre-Join Code:

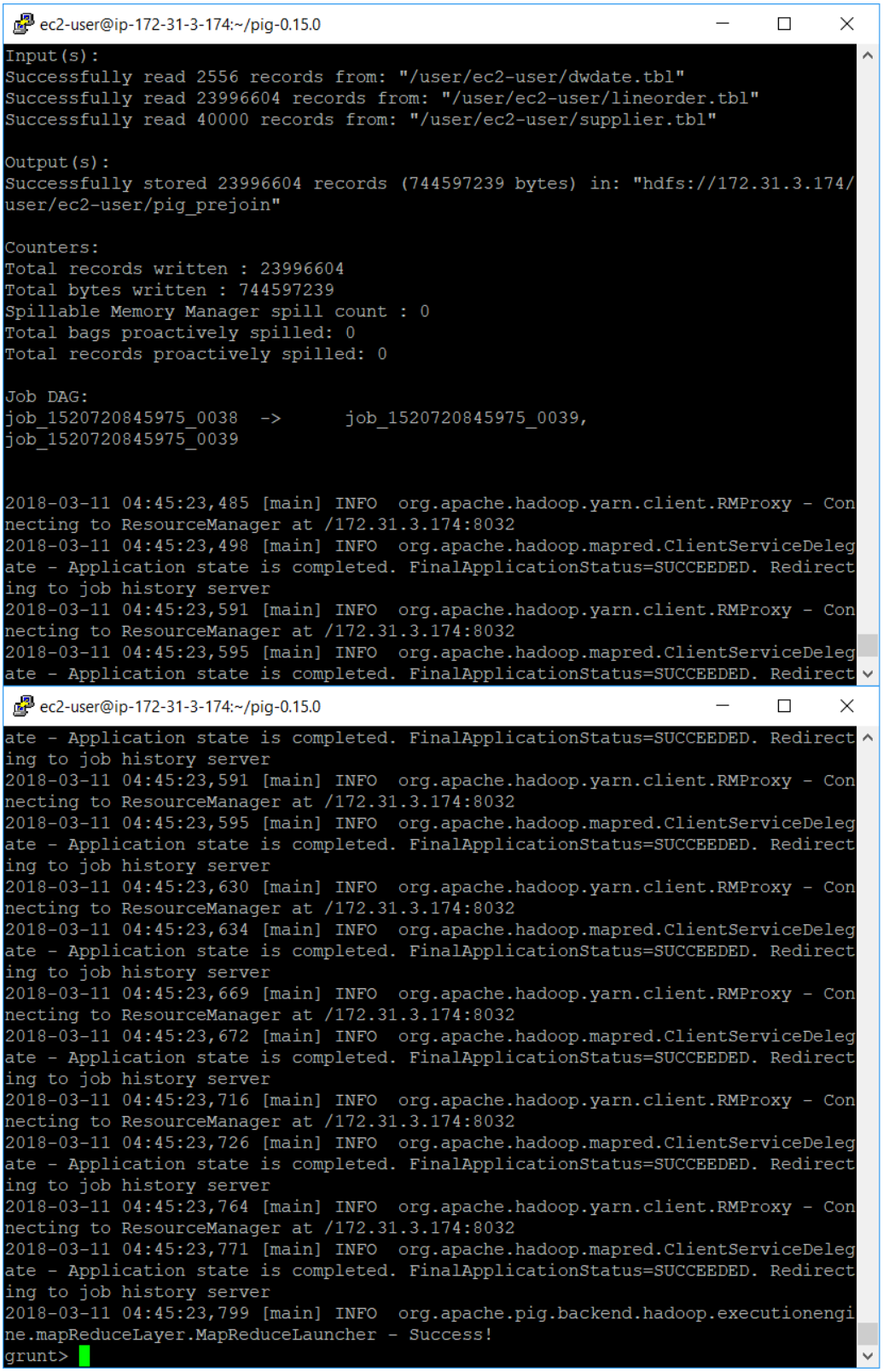
```
lineorder = LOAD '/user/ec2-user/lineorder.tbl' USING PigStorage('|')
AS (lo_orderkey:INT, lo_linenumber:INT, lo_custkey:INT, lo_partkey:INT, lo_suppkey:INT,
lo_orderdate:INT, lo_orderpriority:CHARARRAY, lo_shippriority:CHARARRAY, lo_quantity:INT,
lo_extendedprice:INT, lo_ordertotalprice:INT, lo_discount:INT,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

dwdate = LOAD '/user/ec2-user/dwdate.tbl' USING PigStorage('|')
AS (d_datekey:INT, d_date:CHARARRAY, d_dayofweek:CHARARRAY, d_month:CHARARRAY, d_year:INT,
d_yearmonthnum:INT, d_yearmonth:CHARARRAY, d_daynuminweek:INT, d_daynuminmonth:INT,
d_daynuminyear:INT, d_monthnuminyear:INT, d_weeknuminyear:INT, d_sellingseason:CHARARRAY,
d_lastdayinweekfl:CHARARRAY, d_lastdayinmonthfl:CHARARRAY, d_holidayfl:CHARARRAY,
d_weekdayfl:CHARARRAY);

supplier = LOAD '/user/ec2-user/supplier.tbl' USING PigStorage('|')
AS (s_suppkey:INT, s_name:CHARARRAY, s_address:CHARARRAY, s_city:CHARARRAY,
s_nation:CHARARRAY, s_region:CHARARRAY, s_phone:CHARARRAY);

first_join = JOIN lineorder BY lo_orderdate, dwdate BY d_datekey;
lo_and_date = FOREACH first_join GENERATE lineorder::lo_partkey AS lo_partkey, lineorder::lo_suppkey
AS lo_suppkey, lineorder::lo_revenue AS lo_revenue, dwdate::d_year AS d_year;

second_join = JOIN lo_and_date BY lo_suppkey, supplier BY s_suppkey;
supp_lo_date = FOREACH second_join GENERATE lo_and_date::lo_partkey AS lo_partkey,
lo_and_date::lo_suppkey AS lo_suppkey, lo_and_date::lo_revenue AS lo_revenue, lo_and_date::d_year
AS d_year, supplier::s_suppkey AS s_suppkey;

store supp_lo_date into 'pig_prejoin' using PigStorage(',');
```

Pig Pre-Join Execution (start, map/reduce beginning, and end shown due to length):



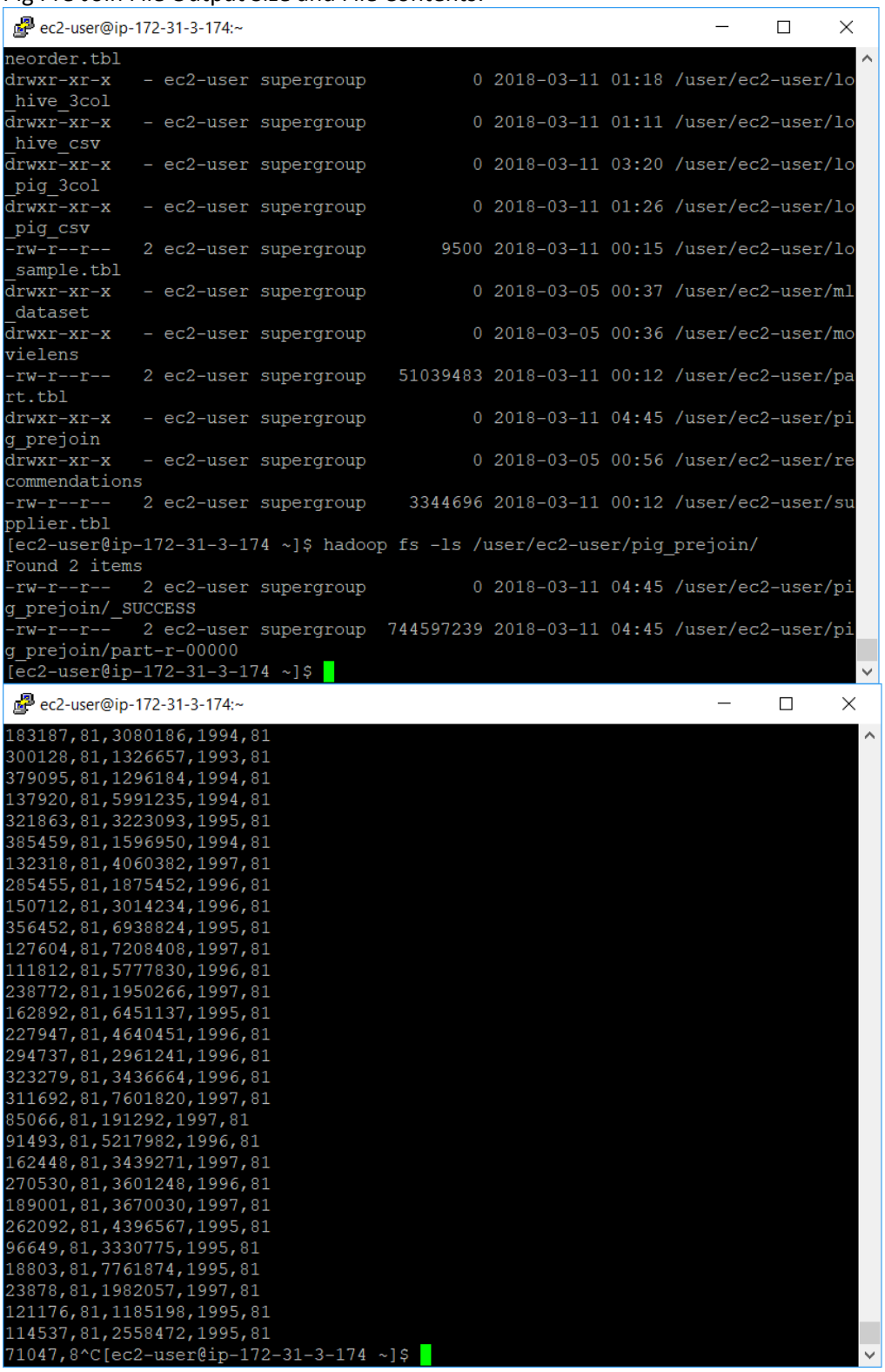

ec2-user@ip-172-31-3-174:~/pig-0.15.0
2018-03-11 04:41:21,805 [JobControl] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
2018-03-11 04:41:21,809 [JobControl] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths (combined) to process : 18
2018-03-11 04:41:21,862 [JobControl] INFO org.apache.hadoop.mapreduce.JobSubmitter - number of splits:19
2018-03-11 04:41:22,041 [JobControl] INFO org.apache.hadoop.mapreduce.JobSubmitter - Submitting tokens for job: job_1520720845975_0038
2018-03-11 04:41:22,188 [JobControl] INFO org.apache.hadoop.mapred.YARNRunner - Job jar is not present. Not adding any jar to the list of resources.
2018-03-11 04:41:22,246 [JobControl] INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl - Submitted application application_1520720845975_0038
2018-03-11 04:41:22,279 [JobControl] INFO org.apache.hadoop.mapreduce.Job - The url to track the job: http://ip-172-31-3-174.us-east-2.compute.internal:8088/proxy/application_1520720845975_0038/
2018-03-11 04:41:22,280 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - HadoopJobId: job_1520720845975_0038
2018-03-11 04:41:22,280 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Processing aliases dwdate,first_join,lineorder,lo_and_date
2018-03-11 04:41:22,280 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - detailed locations: M: lineorder[1,12],lineorder[-1,-1],first_join[7,13],dwdate[3,9],dwdate[-1,-1],first_join[7,13] C: R: lo_and_date[8,14]
2018-03-11 04:41:22,287 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 0% complete
2018-03-11 04:41:22,287 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Running jobs are [job_1520720845975_0038]
2018-03-11 04:41:52,438 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 4% complete


ec2-user@ip-172-31-3-174:~/pig-0.15.0
ate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2018-03-11 04:45:23,385 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /172.31.3.174:8032
2018-03-11 04:45:23,400 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2018-03-11 04:45:23,466 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 100% complete
2018-03-11 04:45:23,484 [main] INFO org.apache.pig.tools.pigstats.mapreduce.SimplePigStats - Script Statistics:
HadoopVersion PigVersion UserId StartedAt FinishedAt Features
2.6.4 0.15.0 ec2-user 2018-03-11 04:41:21 2018-03-11 04:45:23 HASH_JOIN
Success!
Job Stats (time in seconds):
JobId Maps Reduces MaxMapTime MinMapTime AvgMapTime MedianMapTime MaxReduceTime MinReduceTime AvgReduceTime MedianReducetime Alias Feature Outputs
job_1520720845975_0038 19 3 44 19 41 44 52 50 51 51 dwdate,first_join,lineorder,lo_and_date HASH_JOIN
job_1520720845975_0039 5 1 60 7 40 47 135 135 135 135 second_join,supp_lo_date,supplier HASH_JOIN hdfs://172.31.3.174/user/ec2-user/pig_prejoin,
Input(s):
Successfully read 2556 records from: "/user/ec2-user/dwdate.tbl"



The Pig Pre-Join code started at 4:41:21 and ended at 4:45:23, so it took 4 min 4 secs (or 244 secs) to execute on a 4-node cluster.

Pig Pre-Join File Output Size and File Contents:



The size of the Pig pre-join output file is 744597239 bytes or approximately 744,597 MB. This matches the size of the Hive pre-join output directory.

Part 3

A)

Mahout Synthetic Clustering with Number of Clusters = 11

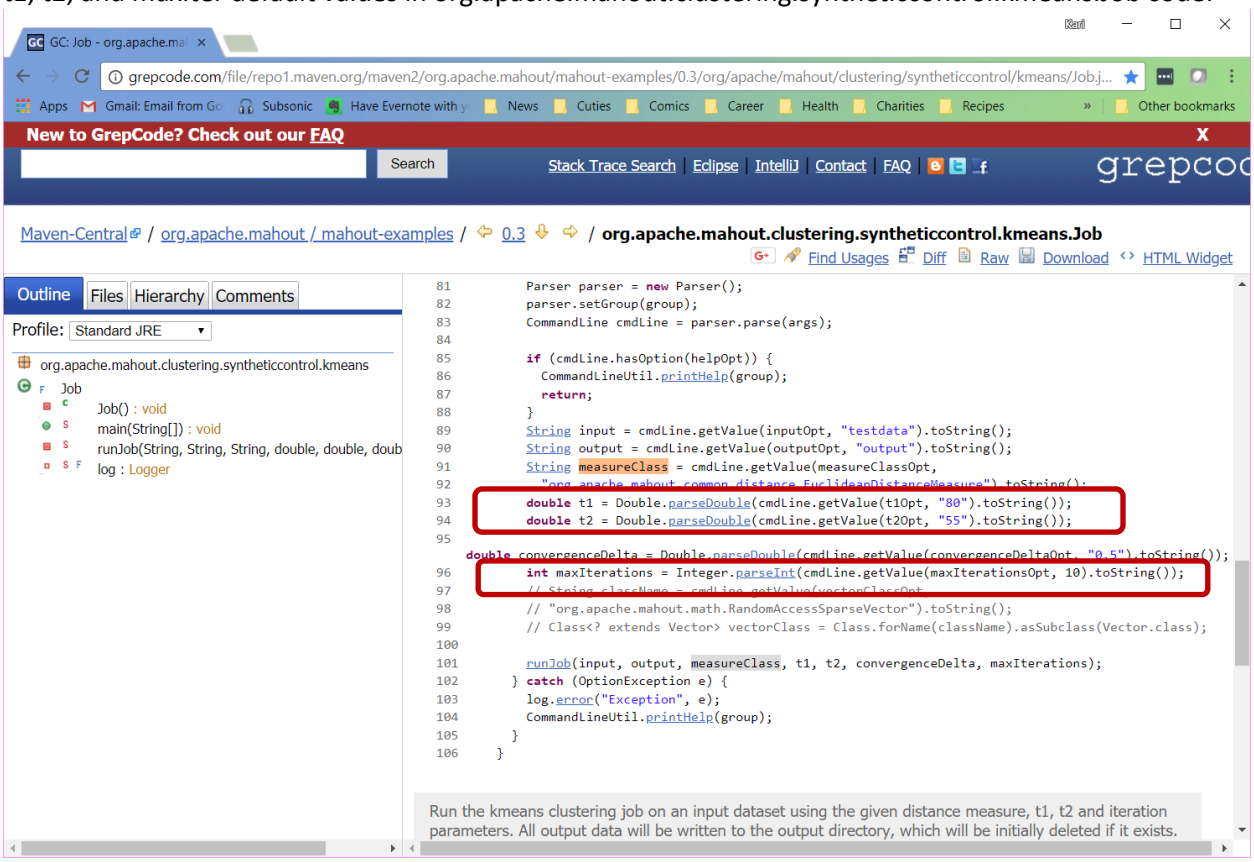
I first created a testdata directory in the HDFS /user/ec2-user/ directory. I then copied the part-00000 Hadoop Streaming output of 1B (the Hadoop Streaming 3 column extraction output) from the HDFS /data/lo_streaming_3col/ directory into the new /user/ec2-user/testdata directory. In Linux, I then went into the /home/ec2-user/apache-mahout-distribution-0.11.2/ directory and ran the Mahout synthetic clustering command below.

Mahout 11 Cluster Synthetic KMeans Command:

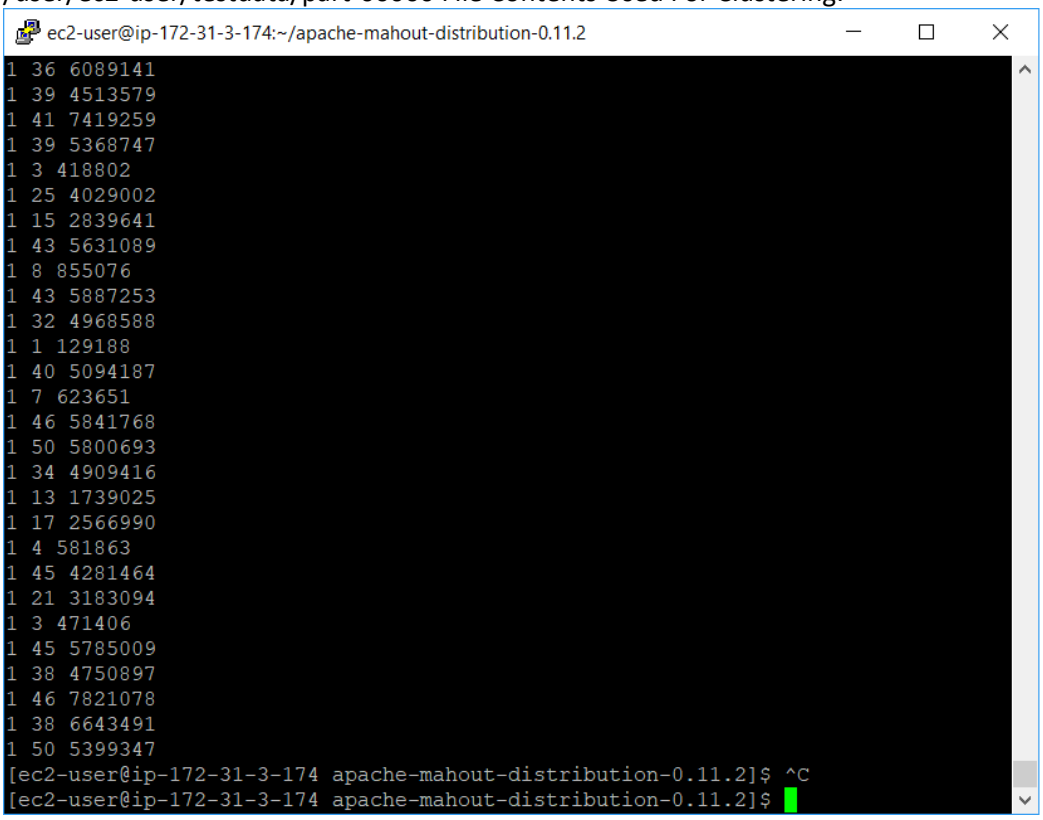
time bin/mahout org.apache.mahout.clustering.syntheticcontrol.kmeans.Job --numClusters 11 --t1 100 --t2 75 --maxIter 10 --input testdata --output output11

I set the t1 value to 100 and t2 to 75 because I was able to find the default values and set them close (the defaults were t1 = 80 and t2 = 55). I set the maxIter to 10 because this was the default.

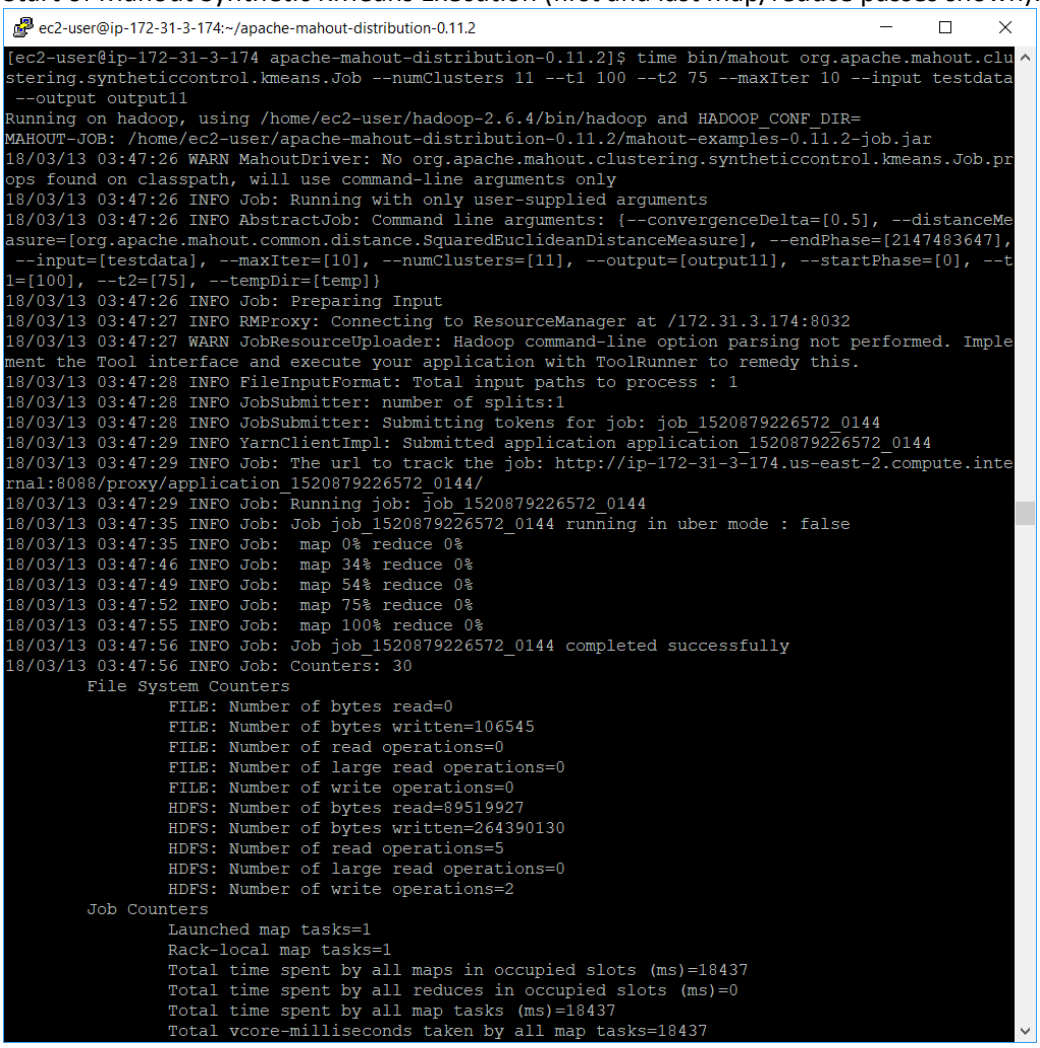
t1, t2, and maxIter default values in org.apache.mahout.clustering.syntheticcontrol.kmeans.Job code:

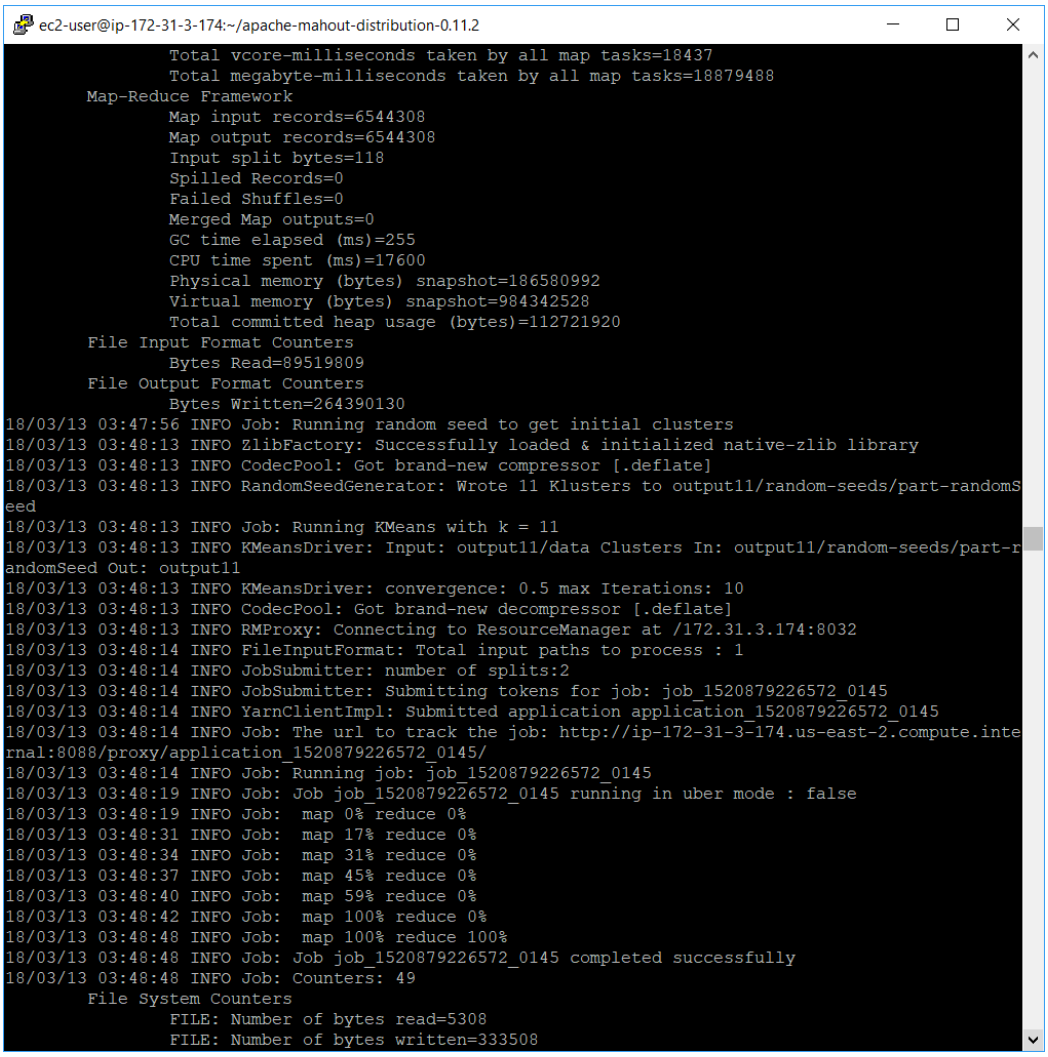


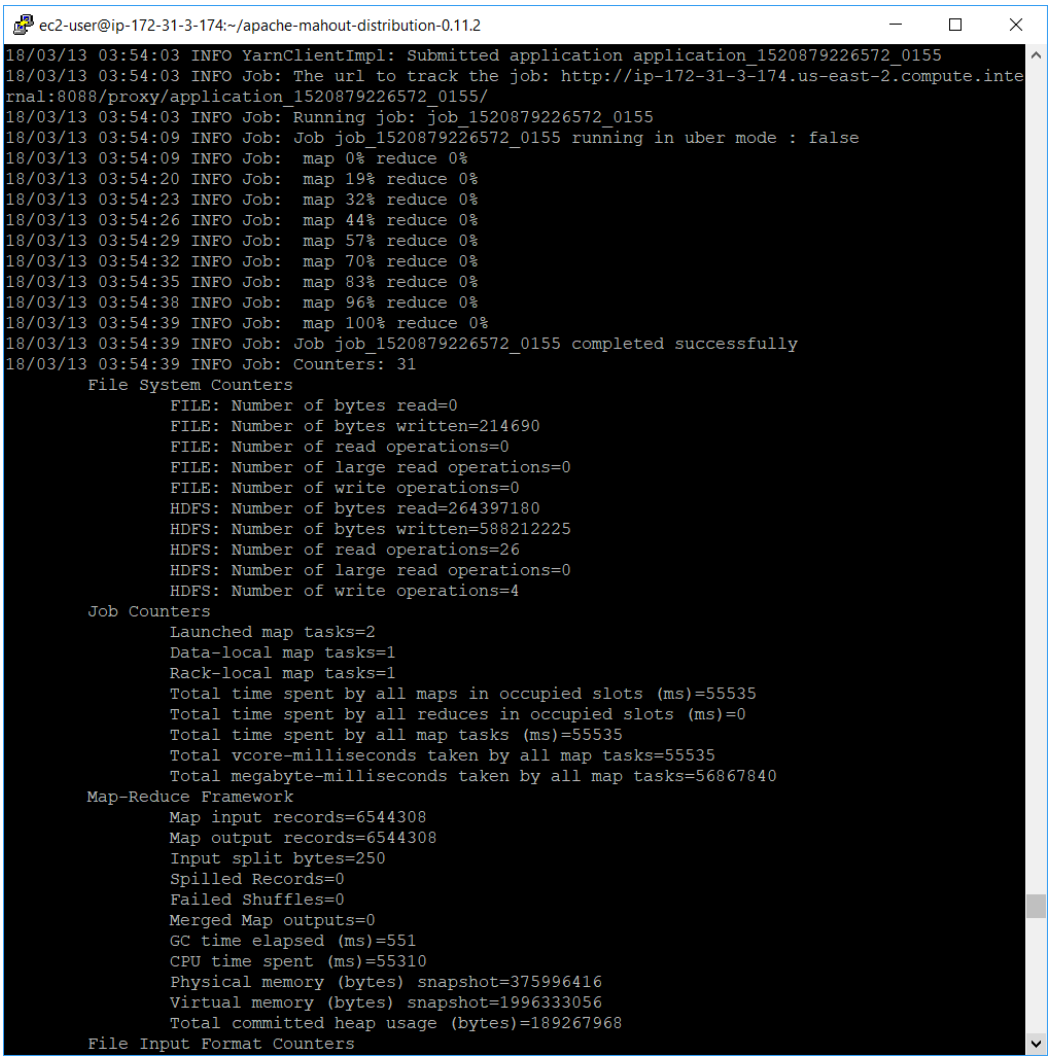
/user/ec2-user/testdata/part-00000 File Contents Used For Clustering:

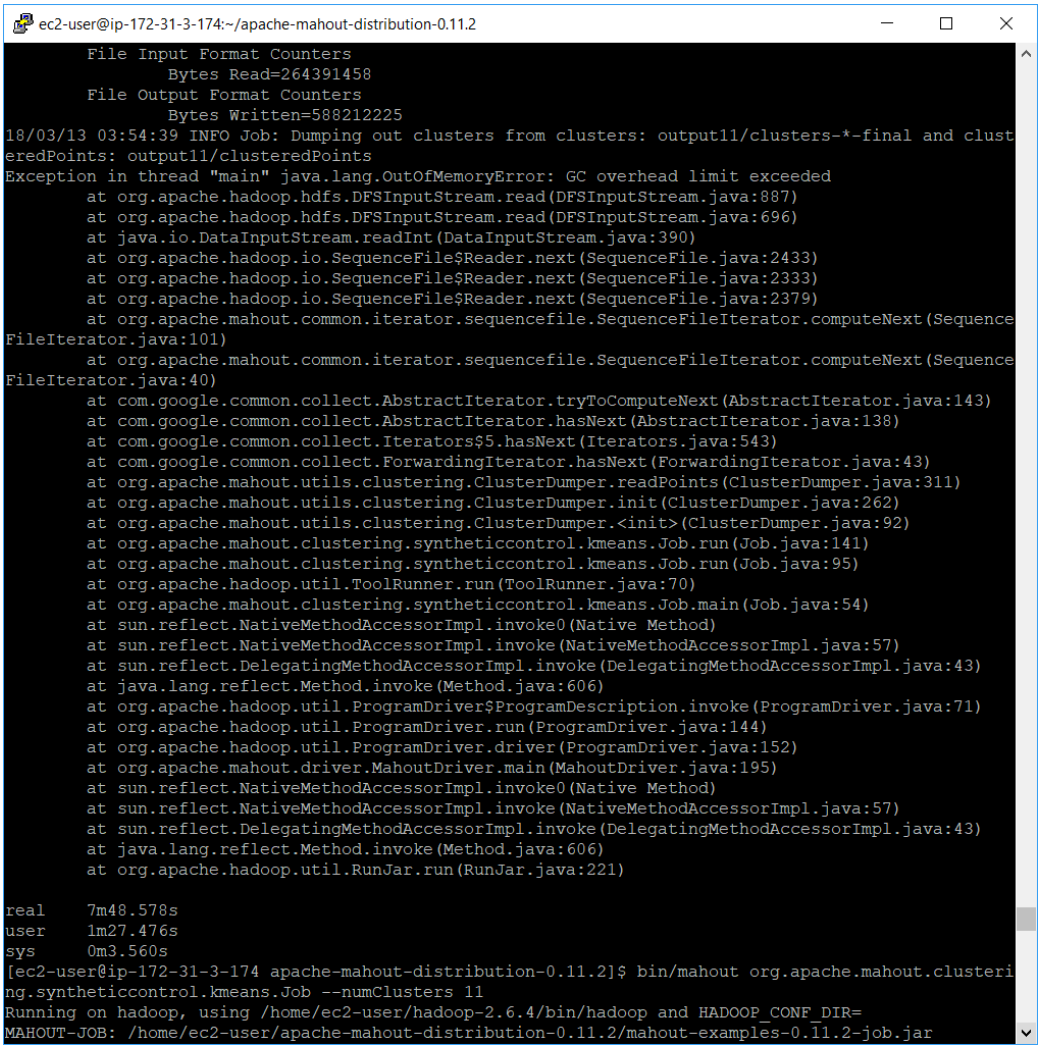


Start of Mahout Synthetic KMeans Execution (first and last map/reduce passes shown):



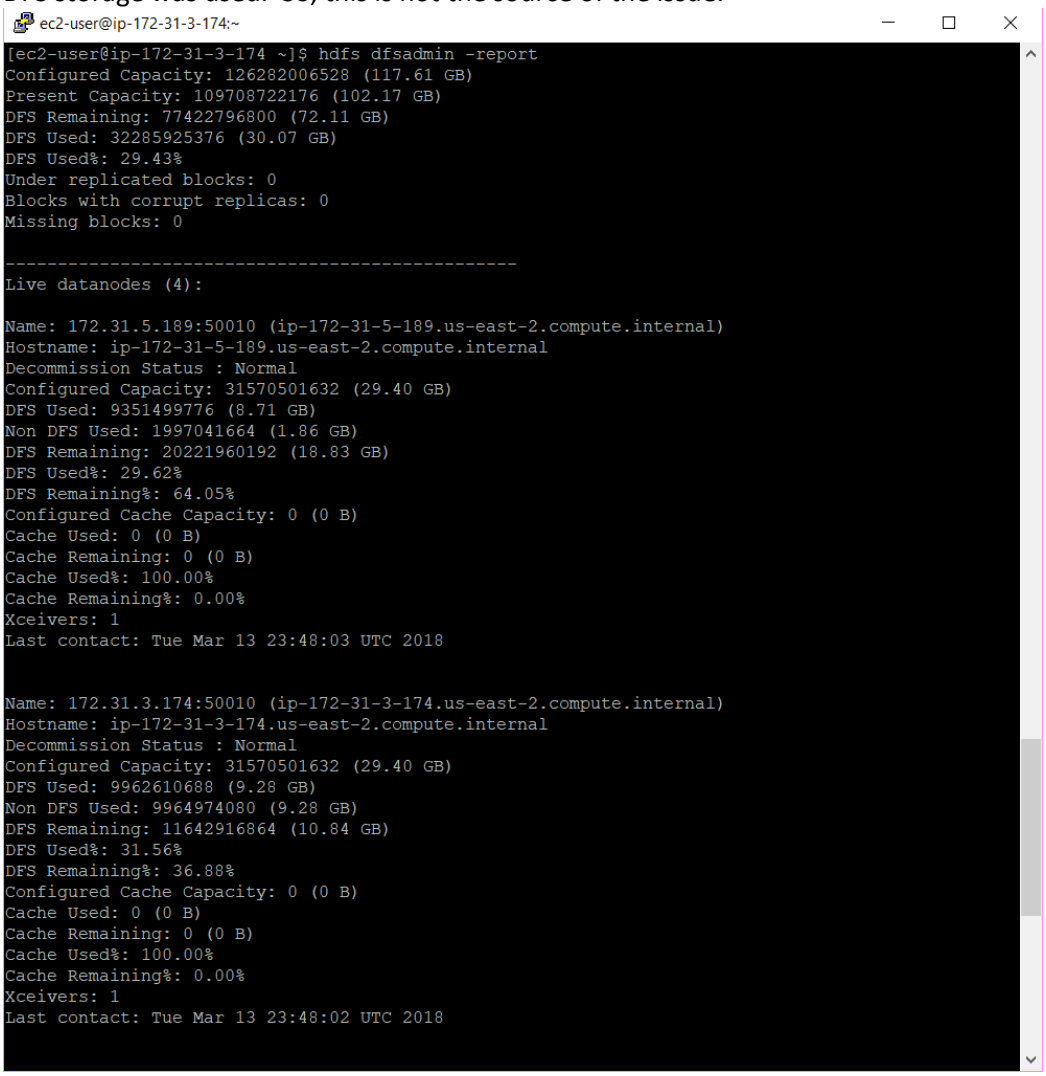

ec2-user@ip-172-31-3-174:~/apache-mahout-distribution-0.11.2
Total vcore-milliseconds taken by all map tasks=18437
Total megabyte-milliseconds taken by all map tasks=18879488
Map-Reduce Framework
Map input records=6544308
Map output records=6544308
Input split bytes=118
Spilled Records=0
Failed Shuffles=0
Merged Map outputs=0
GC time elapsed (ms)=255
CPU time spent (ms)=17600
Physical memory (bytes) snapshot=186580992
Virtual memory (bytes) snapshot=984342528
Total committed heap usage (bytes)=112721920
File Input Format Counters
Bytes Read=89519809
File Output Format Counters
Bytes Written=264390130
18/03/13 03:47:56 INFO Job: Running random seed to get initial clusters
18/03/13 03:48:13 INFO ZlibFactory: Successfully loaded & initialized native-zlib library
18/03/13 03:48:13 INFO CodecPool: Got brand-new compressor [.deflate]
18/03/13 03:48:13 INFO RandomSeedGenerator: Wrote 11 Klusters to output11/random-seeds/part-randomS
eed
18/03/13 03:48:13 INFO Job: Running KMeans with k = 11
18/03/13 03:48:13 INFO KMeansDriver: Input: output11/data Clusters In: output11/random-seeds/part-r
andomSeed Out: output11
18/03/13 03:48:13 INFO KMeansDriver: convergence: 0.5 max Iterations: 10
18/03/13 03:48:13 INFO CodecPool: Got brand-new decompressor [.deflate]
18/03/13 03:48:13 INFO RMProxy: Connecting to ResourceManager at /172.31.3.174:8032
18/03/13 03:48:14 INFO FileInputFormat: Total input paths to process : 1
18/03/13 03:48:14 INFO JobSubmitter: number of splits:2
18/03/13 03:48:14 INFO JobSubmitter: Submitting tokens for job: job_1520879226572_0145
18/03/13 03:48:14 INFO YarnClientImpl: Submitted application application_1520879226572_0145
18/03/13 03:48:14 INFO Job: The url to track the job: http://ip-172-31-3-174.us-east-2.compute.inte
rnal:8088/proxy/application_1520879226572_0145/
18/03/13 03:48:14 INFO Job: Running job: job_1520879226572_0145
18/03/13 03:48:19 INFO Job: Job job_1520879226572_0145 running in uber mode : false
18/03/13 03:48:19 INFO Job: map 0% reduce 0%
18/03/13 03:48:31 INFO Job: map 17% reduce 0%
18/03/13 03:48:34 INFO Job: map 31% reduce 0%
18/03/13 03:48:37 INFO Job: map 45% reduce 0%
18/03/13 03:48:40 INFO Job: map 59% reduce 0%
18/03/13 03:48:42 INFO Job: map 100% reduce 0%
18/03/13 03:48:48 INFO Job: map 100% reduce 100%
18/03/13 03:48:48 INFO Job: Job job_1520879226572_0145 completed successfully
18/03/13 03:48:48 INFO Job: Counters: 49
File System Counters
FILE: Number of bytes read=5308
FILE: Number of bytes written=333508

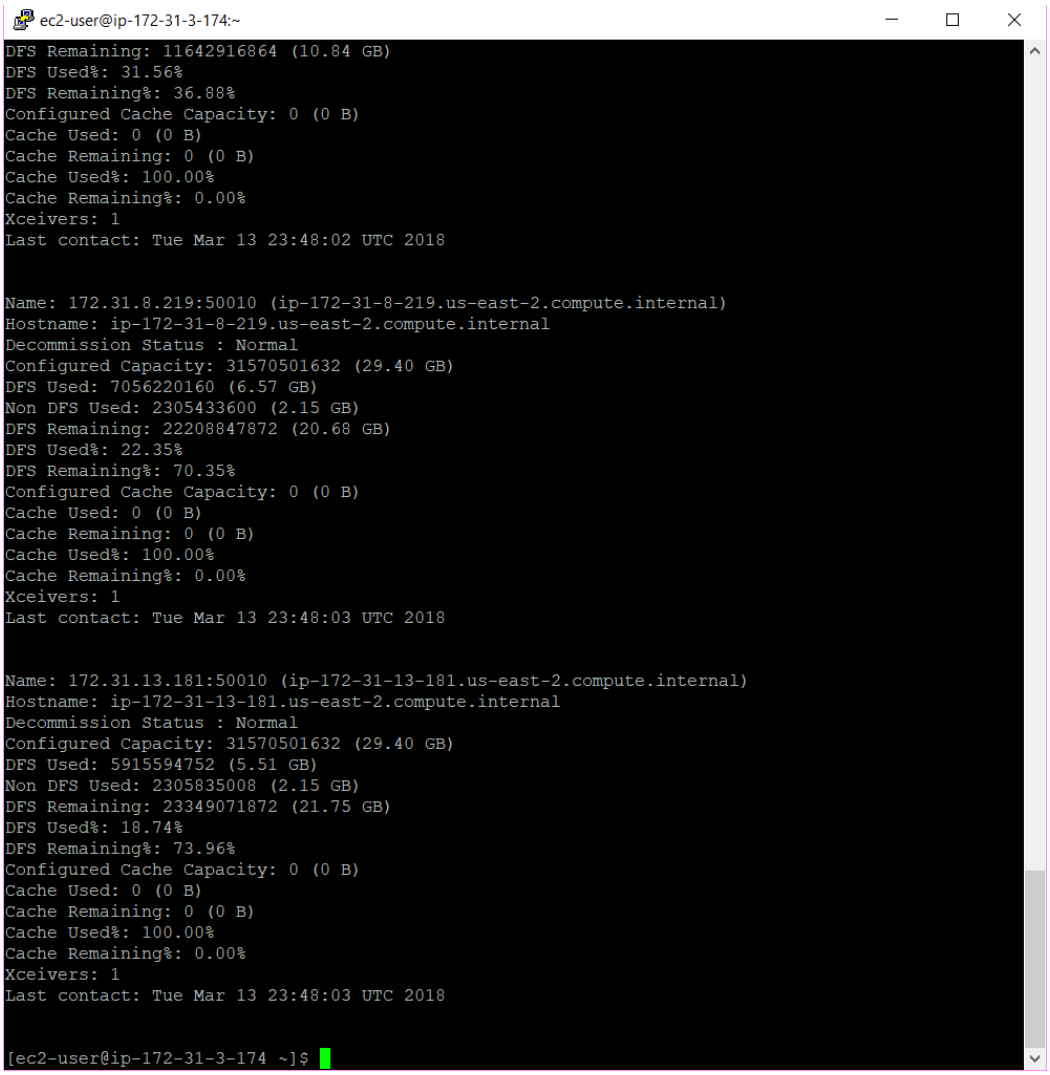

ec2-user@ip-172-31-3-174:~/apache-mahout-distribution-0.11.2
18/03/13 03:54:03 INFO YarnClientImpl: Submitted application application_1520879226572_0155
18/03/13 03:54:03 INFO Job: The url to track the job: http://ip-172-31-3-174.us-east-2.compute.inte
rnal:8088/proxy/application_1520879226572_0155/
18/03/13 03:54:03 INFO Job: Running job: job_1520879226572_0155
18/03/13 03:54:09 INFO Job: Job job_1520879226572_0155 running in uber mode : false
18/03/13 03:54:09 INFO Job: map 0% reduce 0%
18/03/13 03:54:20 INFO Job: map 19% reduce 0%
18/03/13 03:54:23 INFO Job: map 32% reduce 0%
18/03/13 03:54:26 INFO Job: map 44% reduce 0%
18/03/13 03:54:29 INFO Job: map 57% reduce 0%
18/03/13 03:54:32 INFO Job: map 70% reduce 0%
18/03/13 03:54:35 INFO Job: map 83% reduce 0%
18/03/13 03:54:38 INFO Job: map 96% reduce 0%
18/03/13 03:54:39 INFO Job: map 100% reduce 0%
18/03/13 03:54:39 INFO Job: Job job_1520879226572_0155 completed successfully
18/03/13 03:54:39 INFO Job: Counters: 31
File System Counters
FILE: Number of bytes read=0
FILE: Number of bytes written=214690
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=264397180
HDFS: Number of bytes written=588212225
HDFS: Number of read operations=26
HDFS: Number of large read operations=0
HDFS: Number of write operations=4
Job Counters
Launched map tasks=2
Data-local map tasks=1
Rack-local map tasks=1
Total time spent by all maps in occupied slots (ms)=55535
Total time spent by all reduces in occupied slots (ms)=0
Total time spent by all map tasks (ms)=55535
Total vcore-milliseconds taken by all map tasks=55535
Total megabyte-milliseconds taken by all map tasks=56867840
Map-Reduce Framework
Map input records=6544308
Map output records=6544308
Input split bytes=250
Spilled Records=0
Failed Shuffles=0
Merged Map outputs=0
GC time elapsed (ms)=551
CPU time spent (ms)=55310
Physical memory (bytes) snapshot=375996416
Virtual memory (bytes) snapshot=1996333056
Total committed heap usage (bytes)=189267968
File Input Format Counters



The clustering output had 12 map/reduce passes present (not sure why 12 when maxIter was 10). The last map/reduce pass was missing the reduce stage. It also had 0 for the number of merged map outputs, where all other passes had 2. The number of job counters of the last map/reduce pass was 30 versus the 49 to 50 for most map/reduce passes Just after the last pass ended, an “Exception in thread “main” java.lang.OutOfMemoryError: GC overhead limit exceeded” error was received.

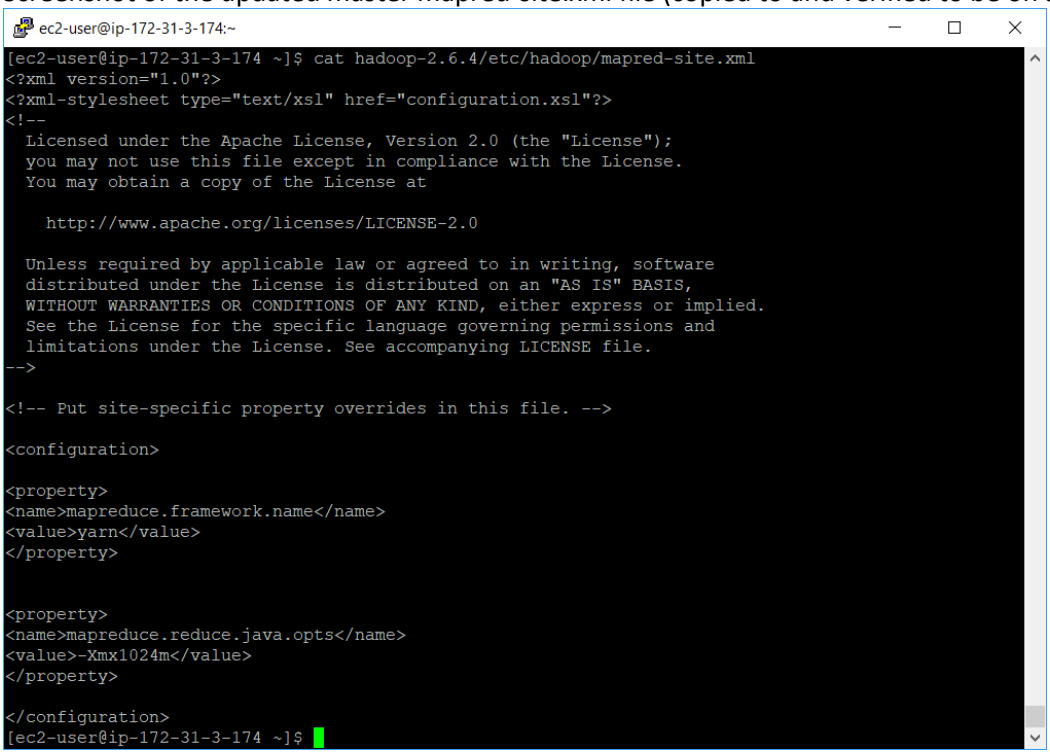
I checked the HDFS report to see how much of the storage was used. I found that only 29.43% of the DFS storage was used. So, this is not the source of the issue.





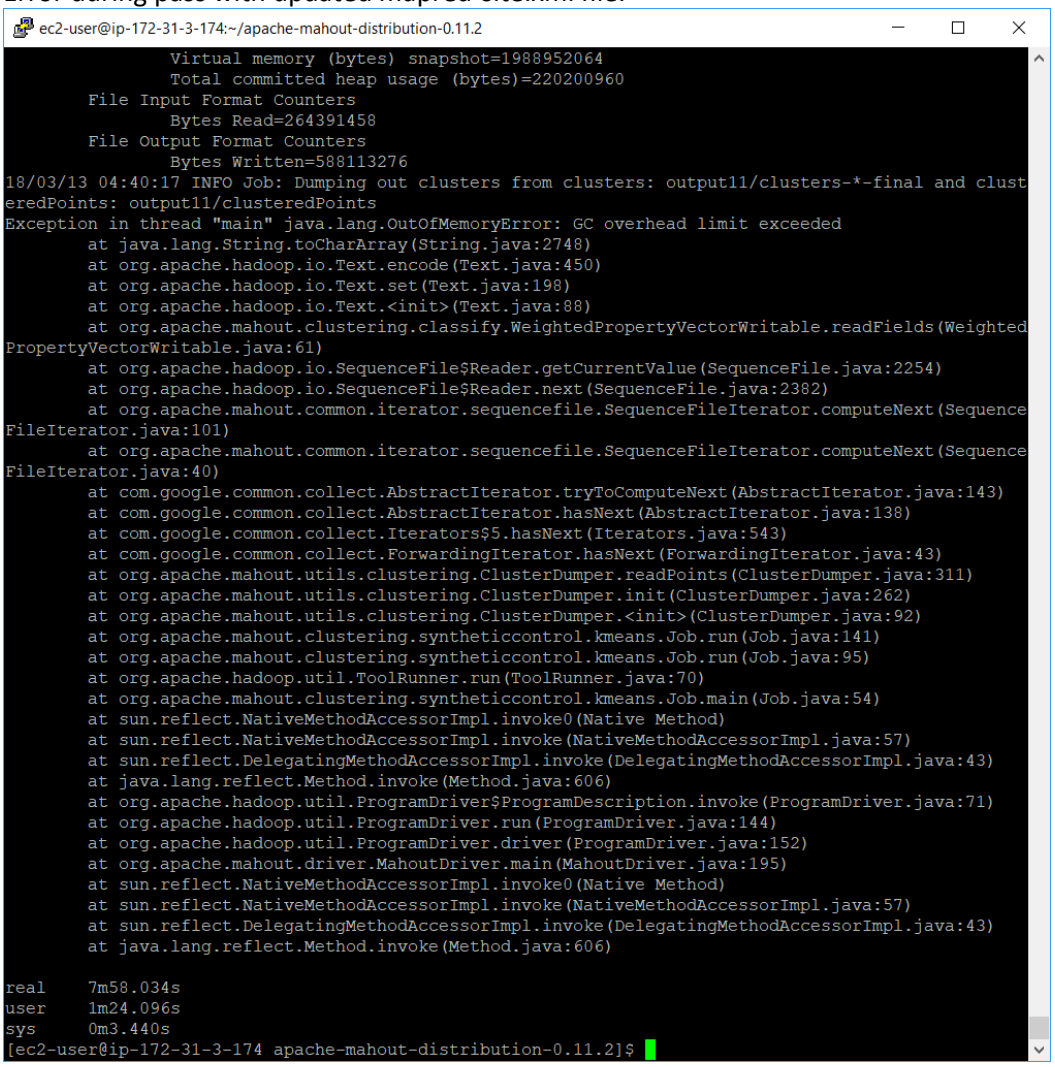
I took the advice of the project assignment and added the mapreduce.reduce.java.opts with a value of -Xmx1024 in the mapred-site.xml file of all nodes of the cluster. The screenshot below is from the master node. I copied this file to all of the workers using scp, then copied the updated file into each /hadoop-2.6.4/etc/hadoop/ folder (removing the original mapred-site.xml file first). I then shut down Hadoop (stopped the history server, yarn, and dfs). I then restarted Hadoop (started the history server, yarn, and dfs).

Screenshot of the updated master mapred-site.xml file (copied to and verified to be on all clusters):



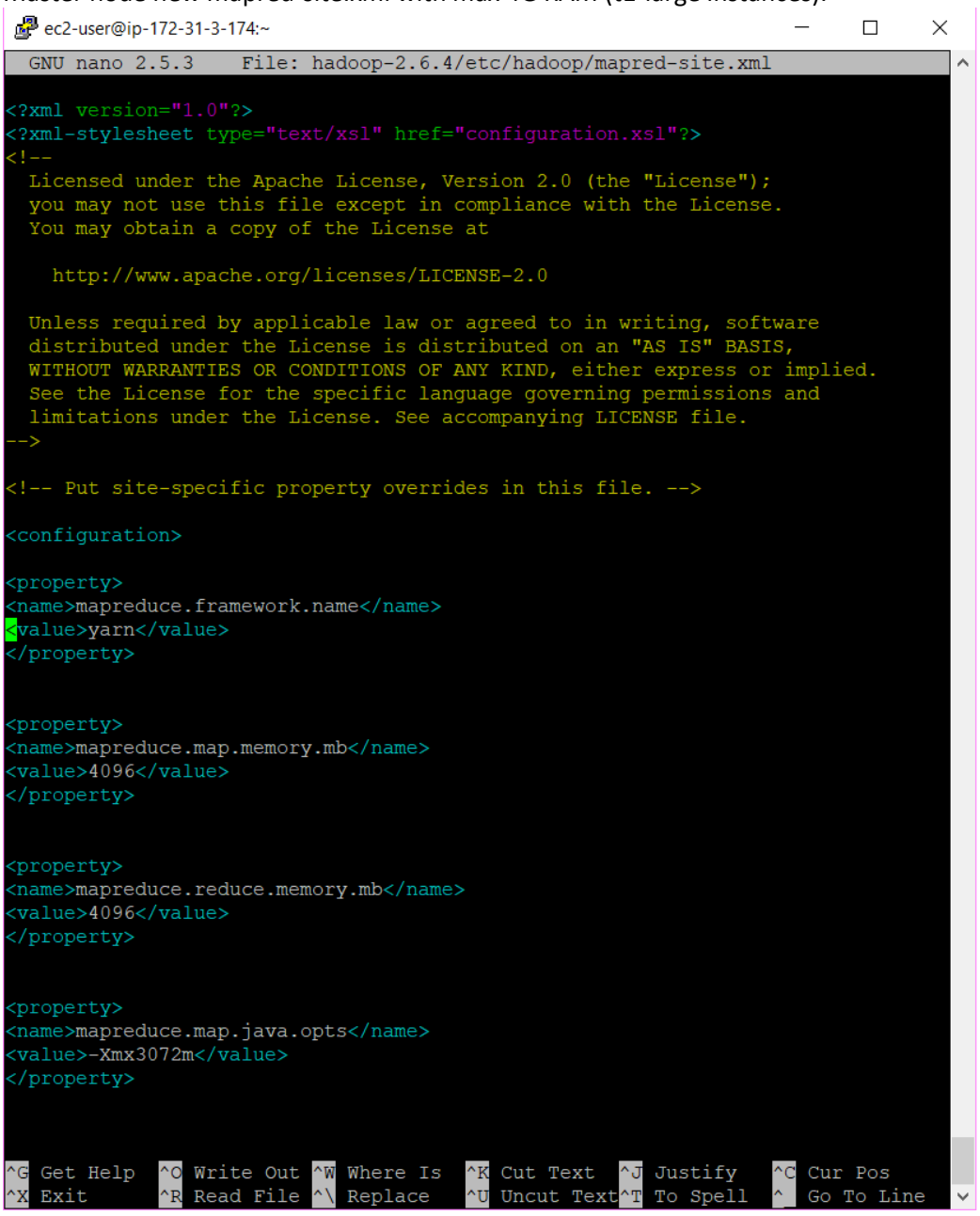
I then went back into the apache mahout directory and re-ran the clustering command for 11 clusters. I received the same error at the last map/reduce pass that I did before updating the mapred-site.xml file. It occurred during the 12th map/reduce pass again and had the same characteristics (no reduce pass, 0 merged map output setting, and number of job counters 30 versus the 49 to 50 for most map/reduce passes).

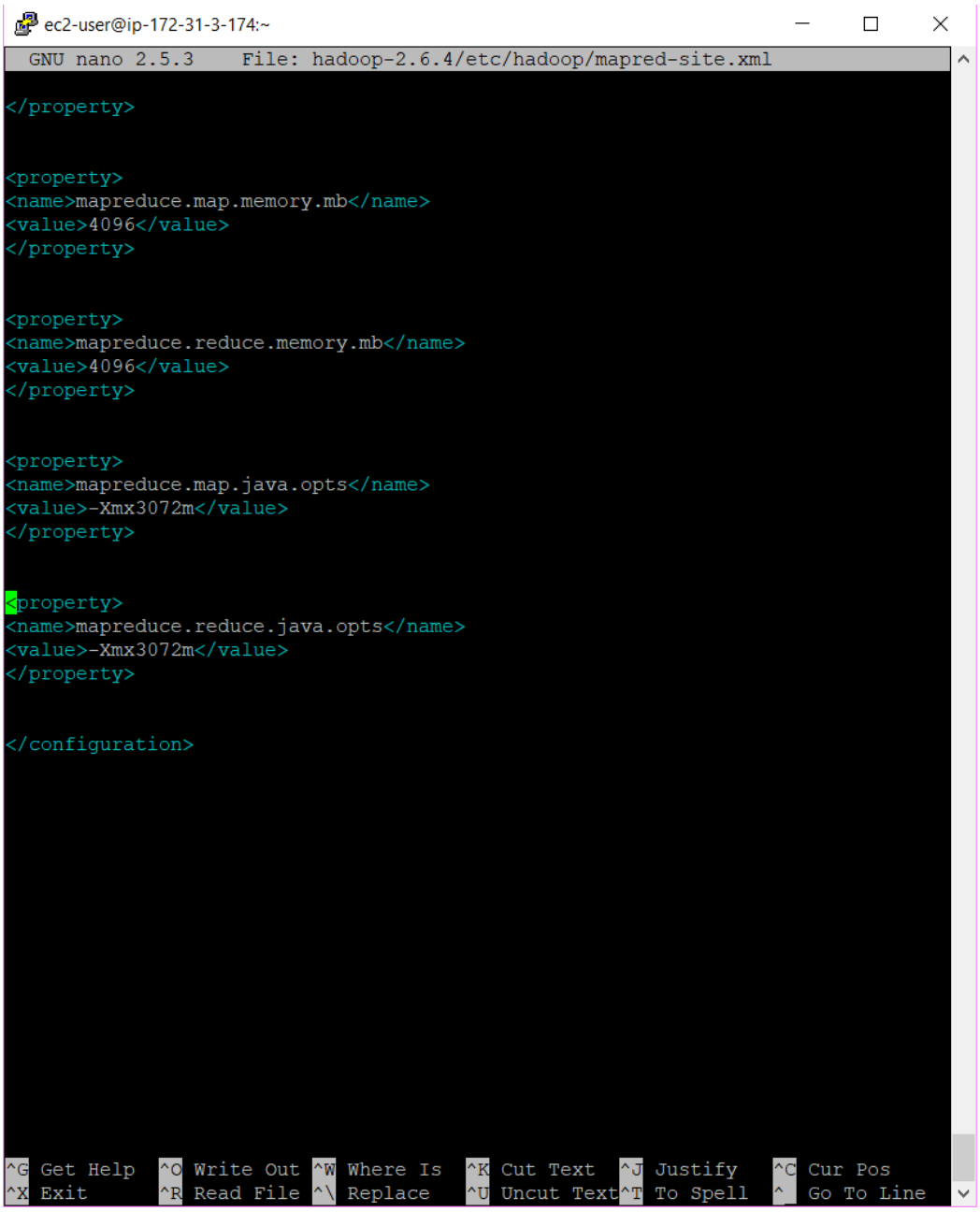
Error during pass with updated mapred-site.xml file:



Next, I stopped Hadoop, exited the cluster, and stopped all of the instances in AWS. I then increased all of the nodes in AWS to be t1-large instances with 8G of memory. After this, I started the instances again. I then updated the mapred-site.xml file to set the max memory available for mapping and reducing to 4G each and the map and reduce java.opts to 3G each (found this to be a way online to work around the out of memory error). I again made the same updates to all of the mapred-site.xml files on all of the nodes. The screenshot below is of the file on the master node.

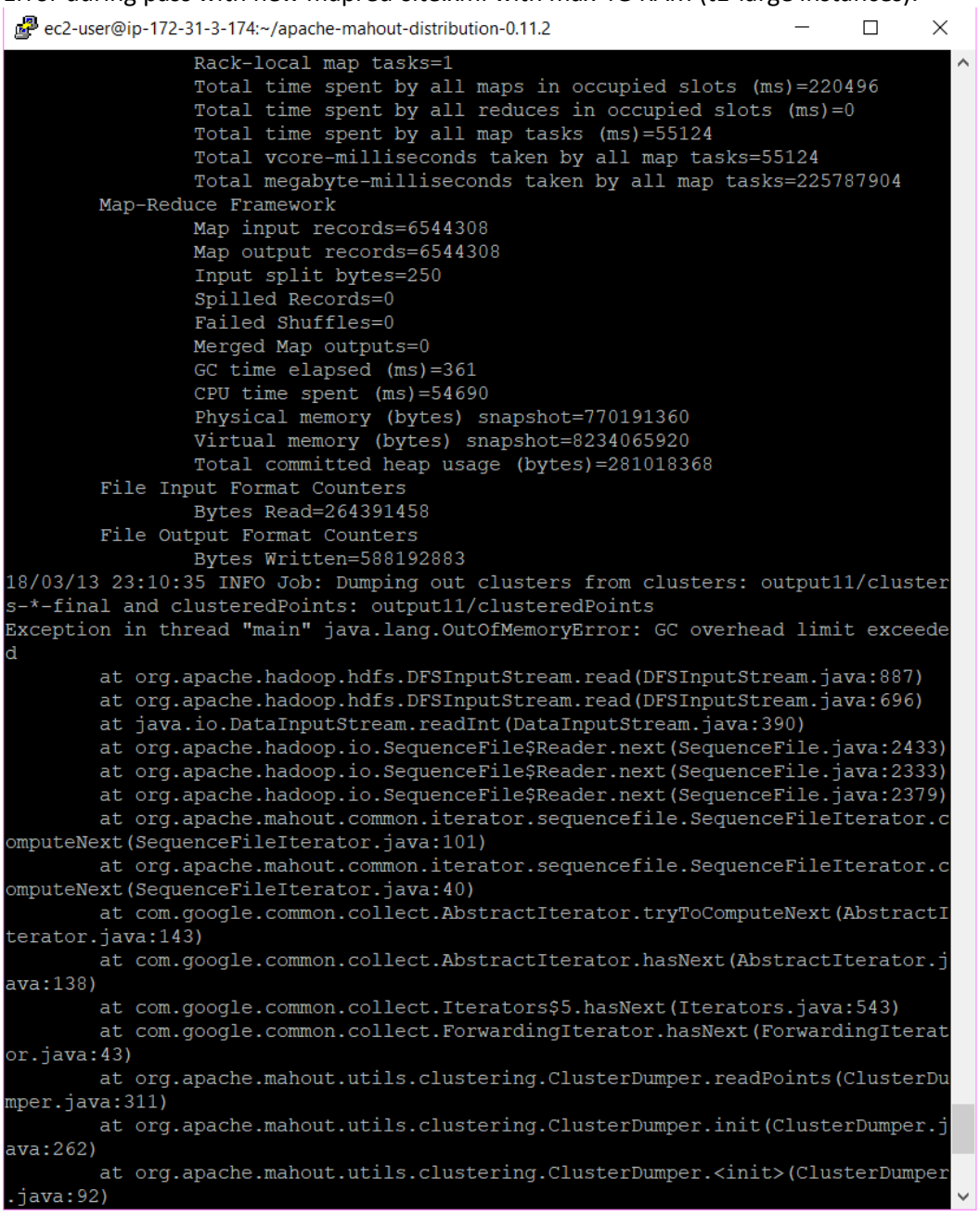
Master node new mapred-site.xml with max 4G RAM (t1-large instances):





I then restarted Hadoop (started the history server, yarn, and dfs). I then went back into the apache mahout directory and re-ran the clustering command for 11 clusters. I again received the same error at the last map/reduce pass that I did on the previous 2 tries). I shutdown Hadoop, stopped the instances, and changed them back to t1 medium. I also reverted all of the mapred-site.xml files on all nodes to the original (without any mapred memory or java.opts entries). I then started Hadoop again for use in the rest of the project (the cluster was returned to the same state it was in before part 3A).

Error during pass with new mapred-site.xml with max 4G RAM (t1-large instances):

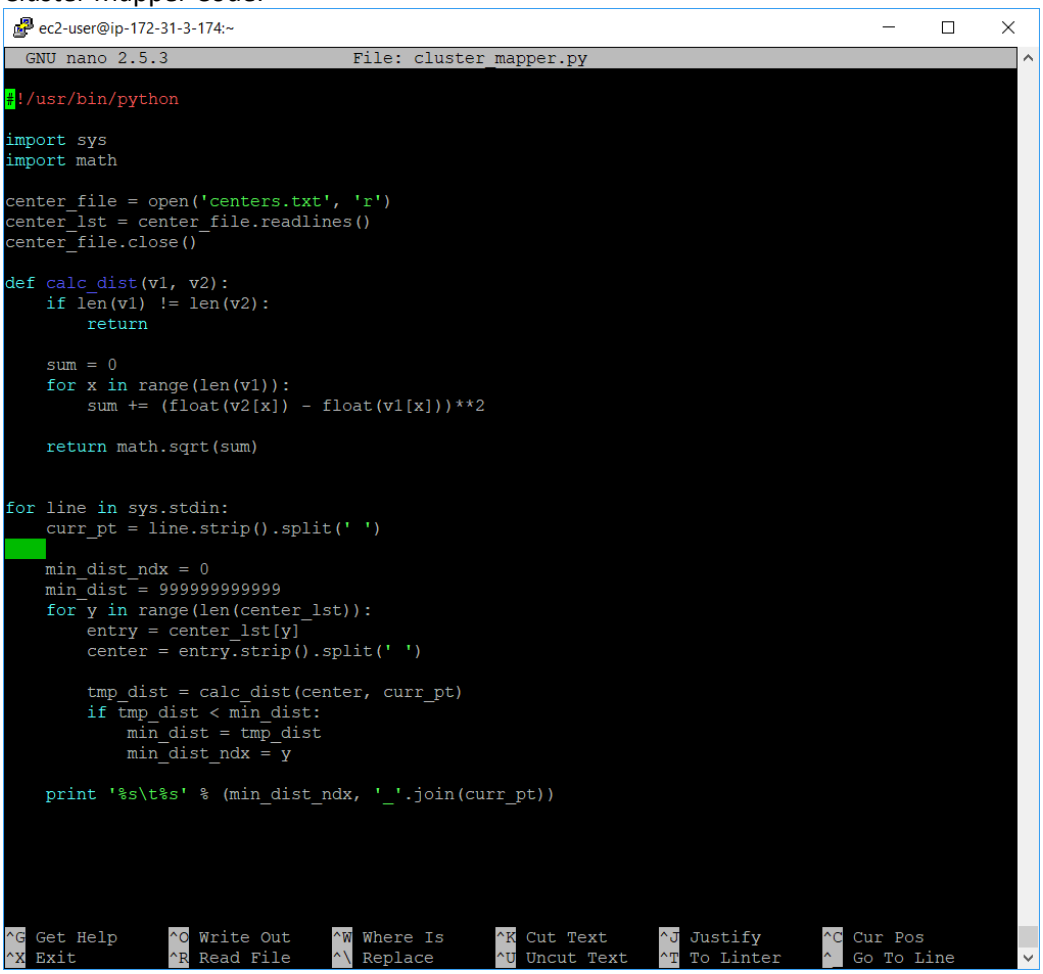


B)

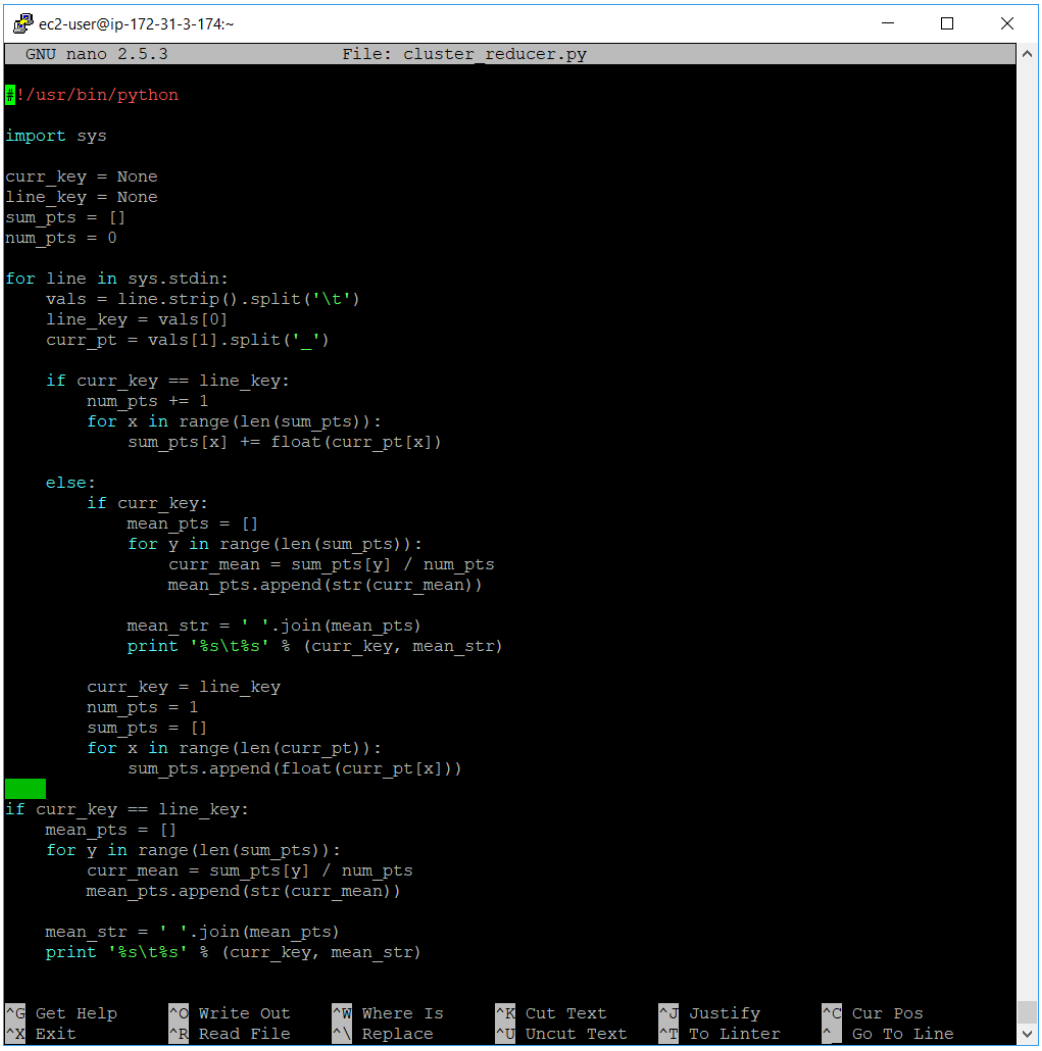
3 Passes of Manual Hadoop Streaming Clustering With Number of Clusters = 11

I created a centers.txt file in the /home/ec2-user/ Linux directory. I populated it with 11 points I chose from the 3 column data in the HDFS /user/ec2-user/testdata/part-00000 file. This centers file was read in by the mapper code. The mapper code determined which center each line of the current block was closest to and printed the cluster number as the key and the current point (the three values of the current line) as the value, separated by underscores. The reducer then received all of the points assigned with the same cluster number in order. The reducer looped over the points and added each of the column entries together (generated a sum of the first column, second column, and third column), as well as updated a counter to track the number of points in the current cluster number. When a new cluster number was reached, the sum of each of the columns was divided by the number of points to create a mean point (mean of the first column, mean of the second column, and mean of the third column are the mean point). The keys of each reducer output were set to the cluster number corresponding to each center and the value was the three entries of the new cluster mean, separated by spaces. The output of all of the reducers was the new cluster centers to be used in the next round of clustering. After the new cluster centers were created, I copied the file from HDFS to the Linux ec2-user home directory, renamed the old centers.txt file to reflect the pass it was used for, then renamed the clustering output to centers.txt. I then updated the centers.txt file to delete the cluster number keys, so only the center points remained. I repeated this entire process until a total of 3 passes were reached.

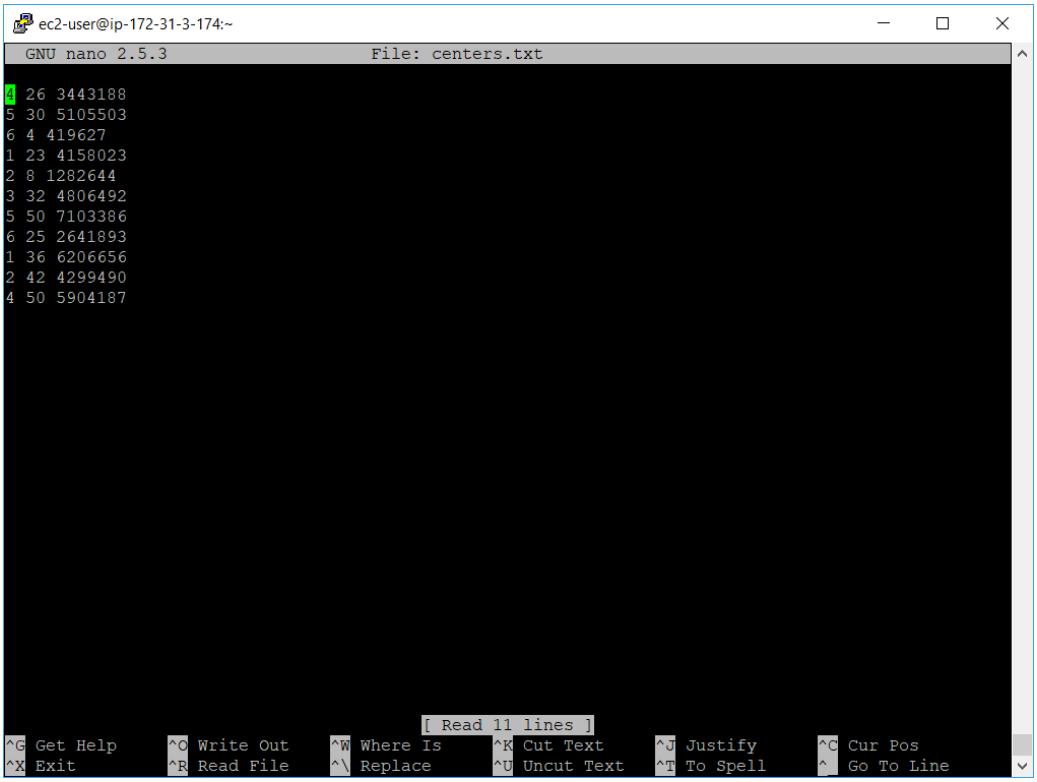
Cluster Mapper Code:



Cluster Reducer Code:



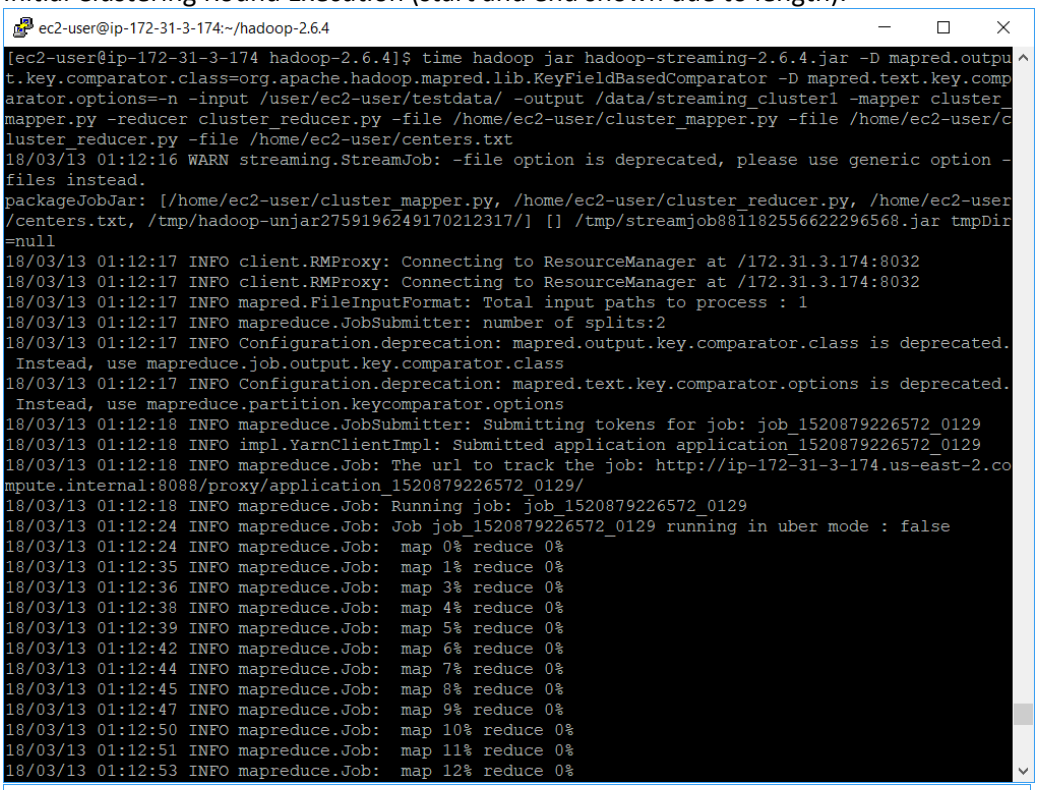
Initial Cluster centers.txt File:

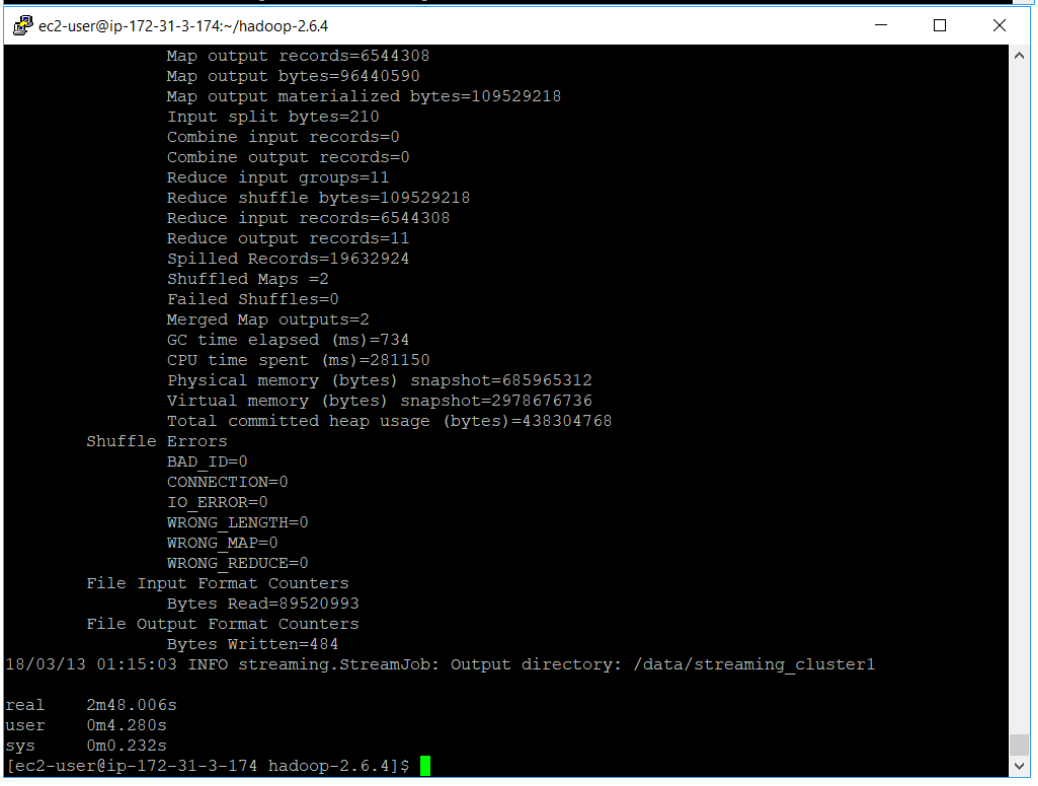


Initial Hadoop Streaming Command Run:

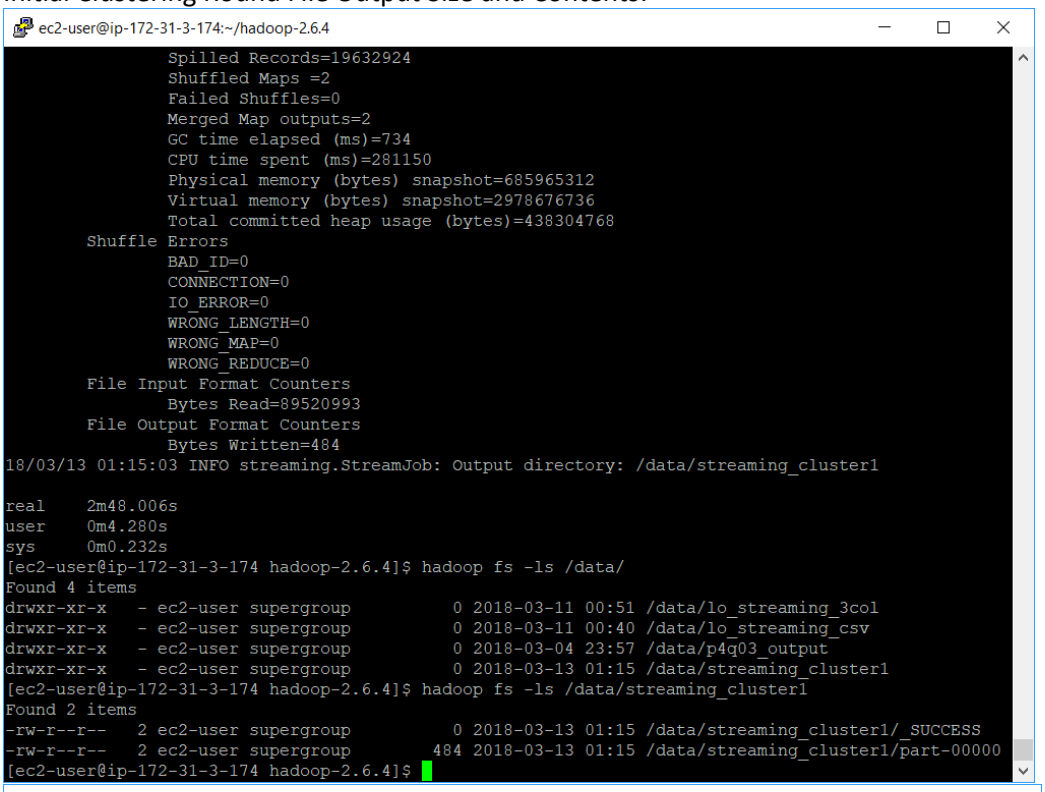
time hadoop jar hadoop-streaming-2.6.4.jar -D mapred.output.key.comparator.class=org.apache.hadoop.mapred.lib.KeyFieldBasedComparator -D mapred.text.key.comparator.options=-n -input /user/ec2-user/testdata/ -output /data/streaming_cluster1 -mapper cluster_mapper.py -reducer cluster_reducer.py -file /home/ec2-user/cluster_mapper.py -file /home/ec2-user/cluster_reducer.py -file /home/ec2-user/centers.txt

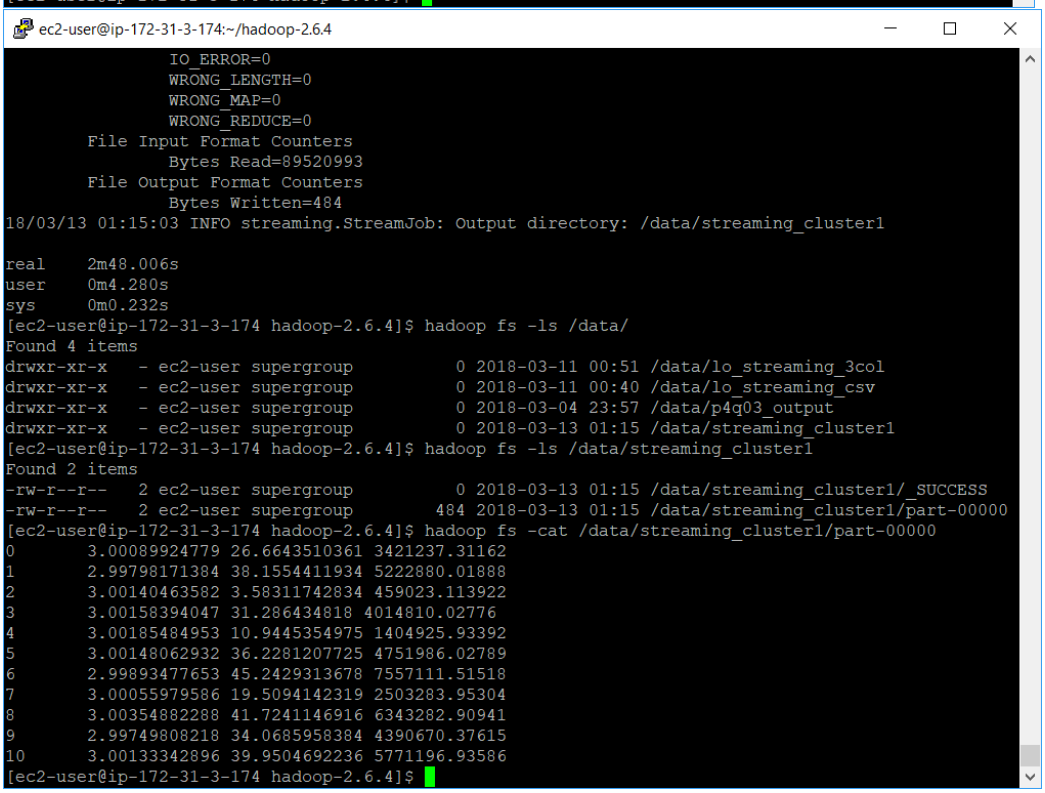
Initial Clustering Round Execution (start and end shown due to length):



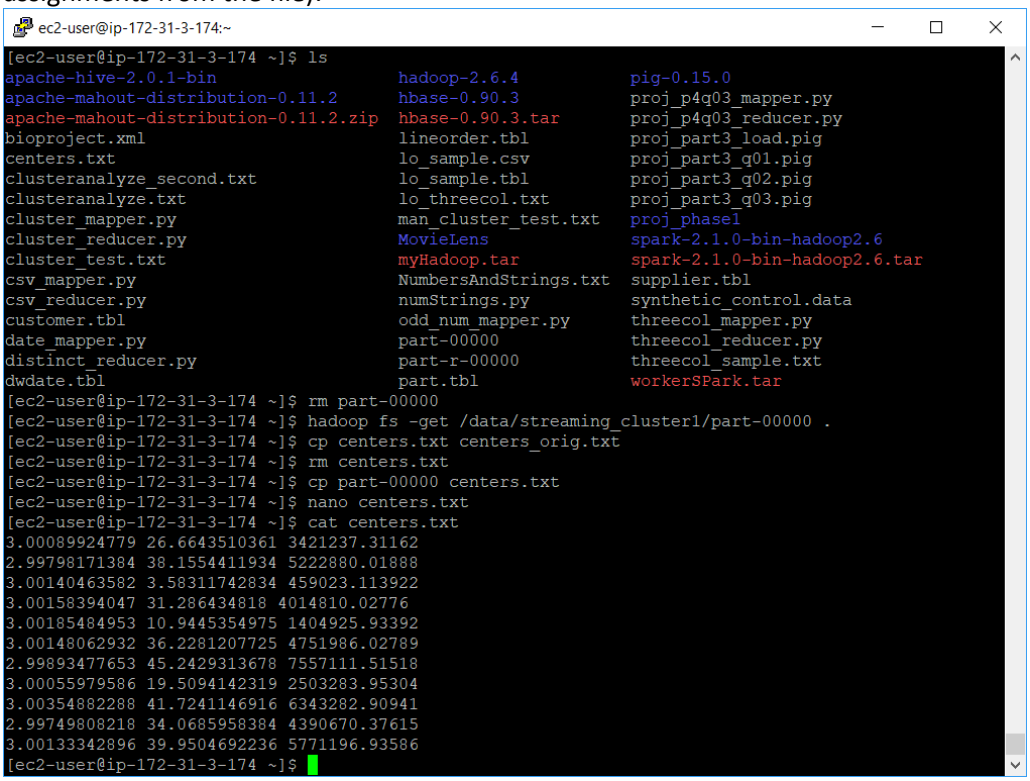


Initial Clustering Round File Output Size and Contents:





Creation of new centers.txt file from the initial round clustering output (including nano to remove center assignments from the file):

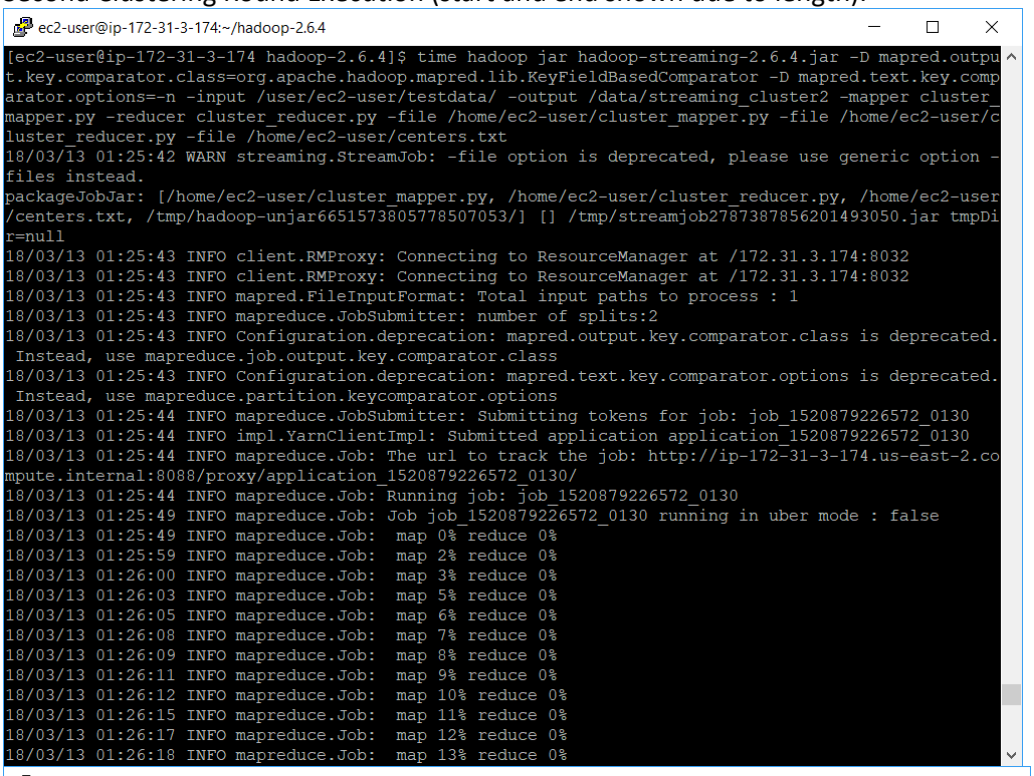


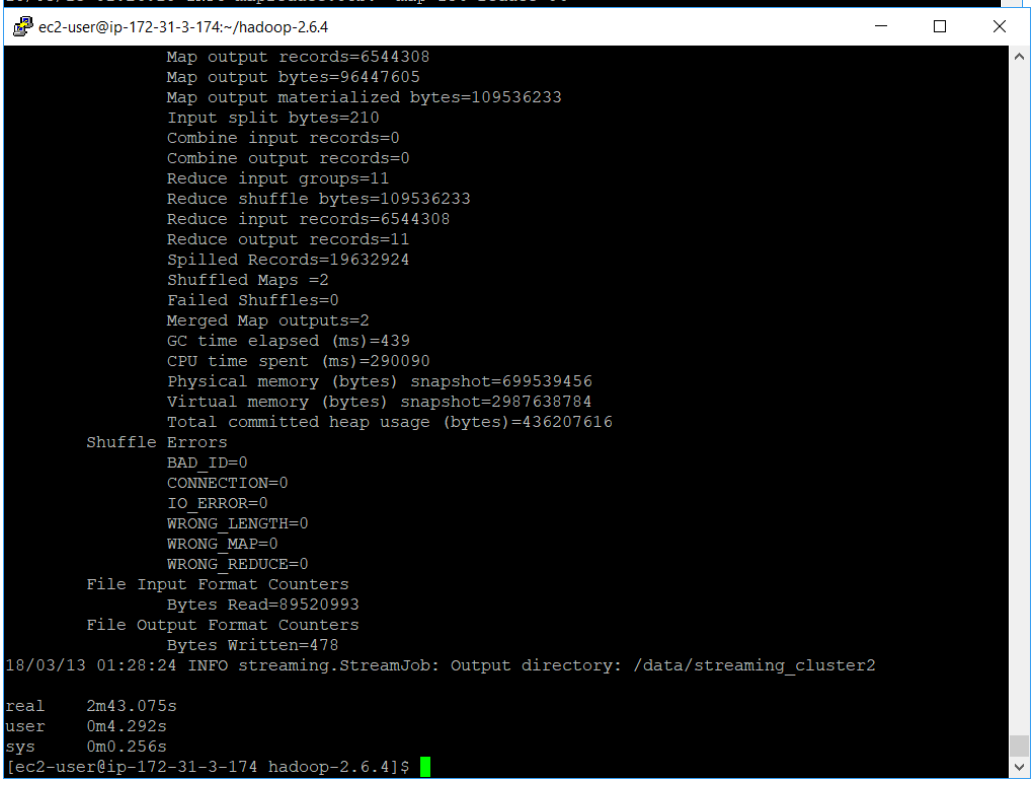
Second Hadoop Streaming Command Run:

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/testdata/ -output /data/streaming_cluster2 -mapper cluster_mapper.py -reducer cluster_reducer.py -file /home/ec2-user/cluster_mapper.py -file /home/ec2-user/cluster_reducer.py -file /home/ec2-user/centers.txt

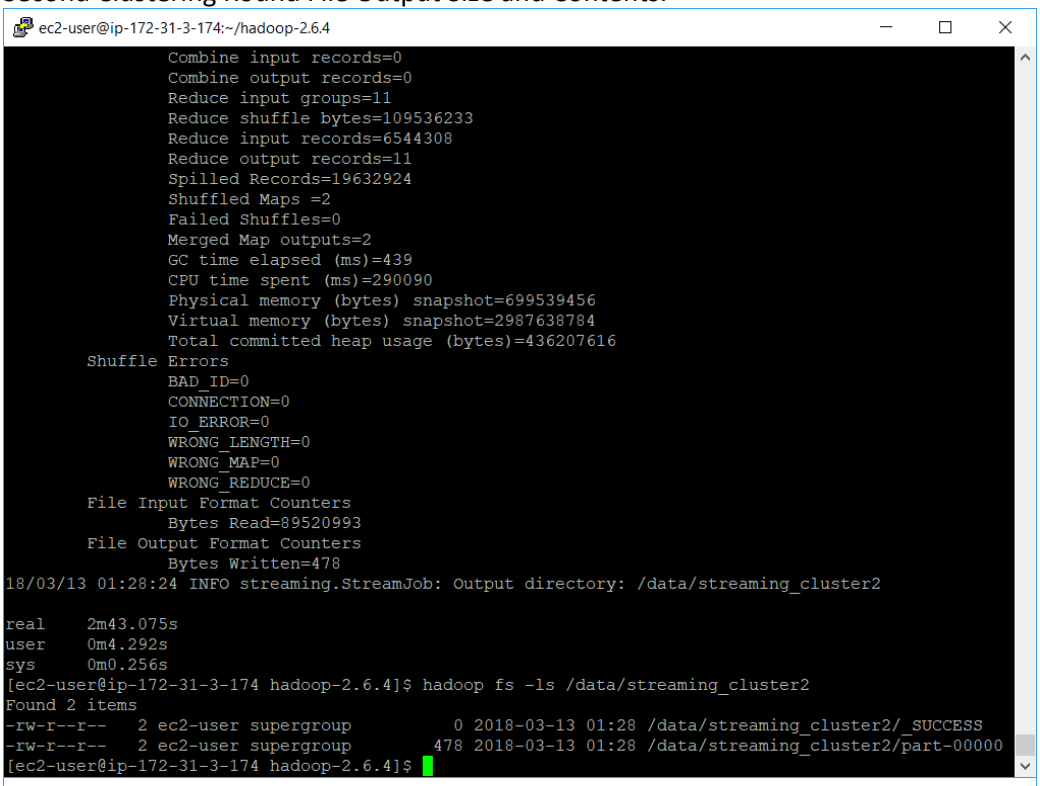
The only change to this command is the output folder being streaming_cluster2 instead of streaming_cluster1.

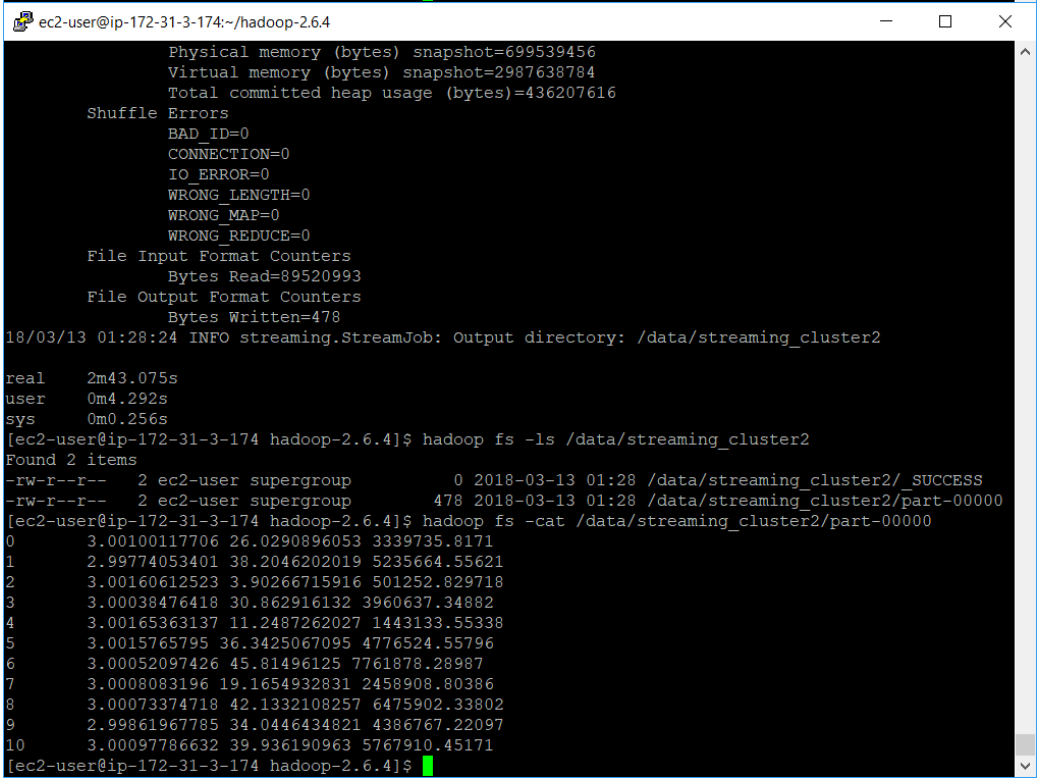
Second Clustering Round Execution (start and end shown due to length):



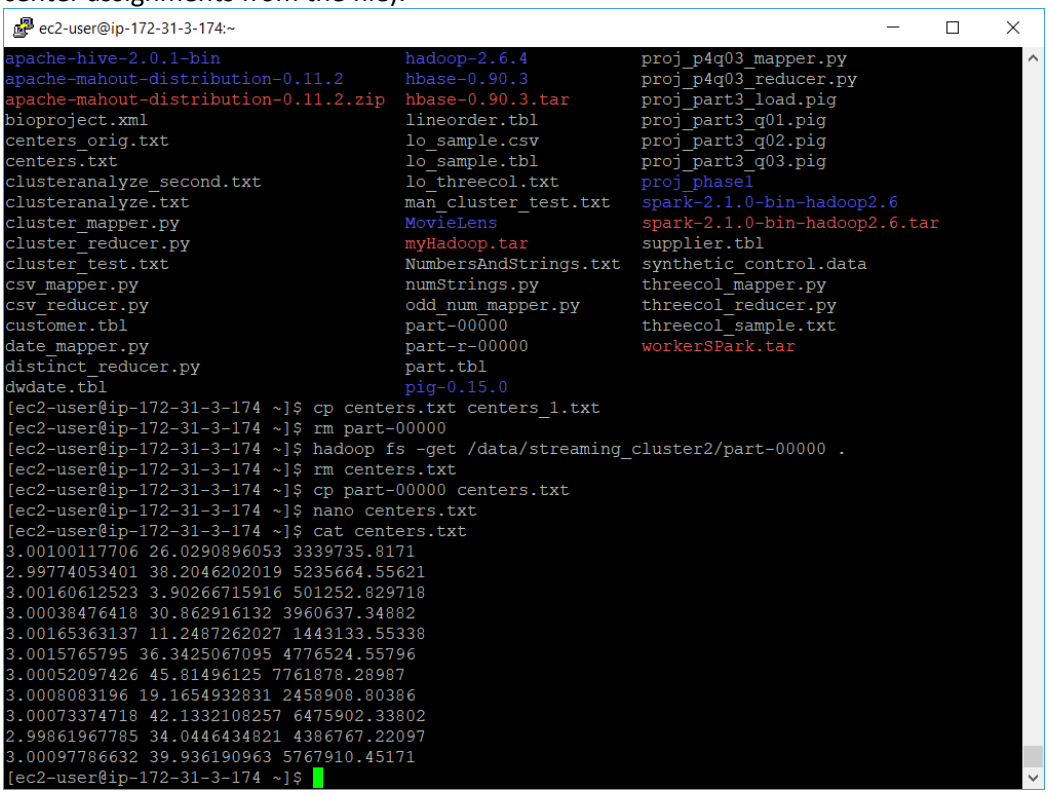


Second Clustering Round File Output Size and Contents:





Creation of new centers.txt file from the second round clustering output (including nano to remove center assignments from the file):

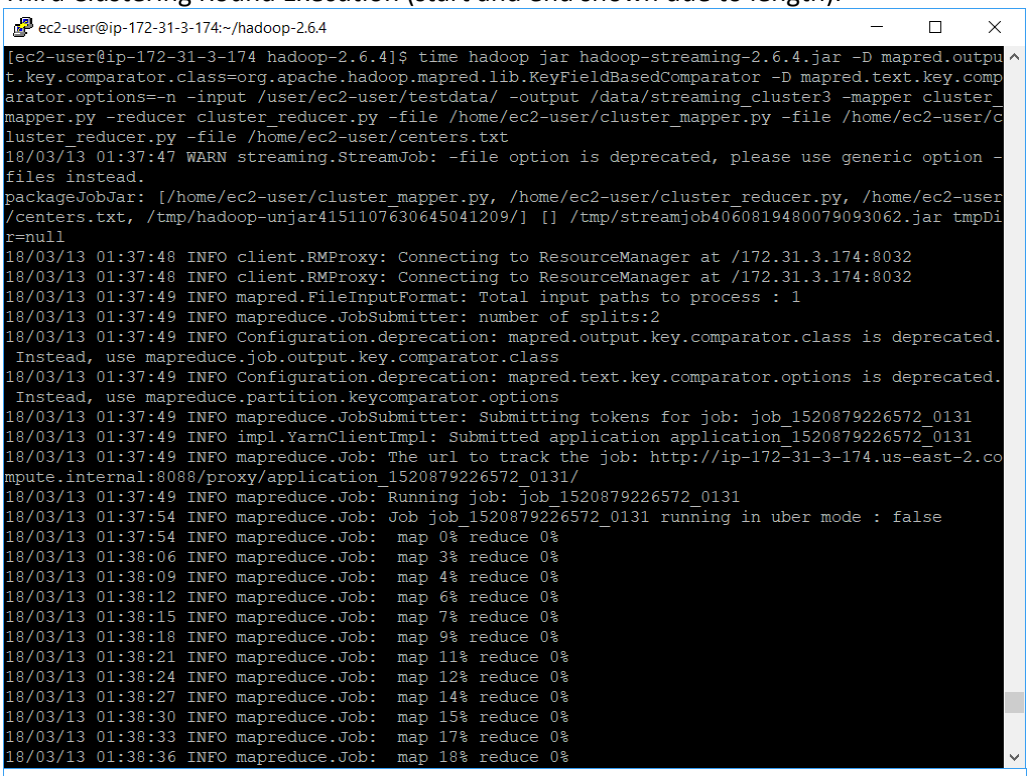


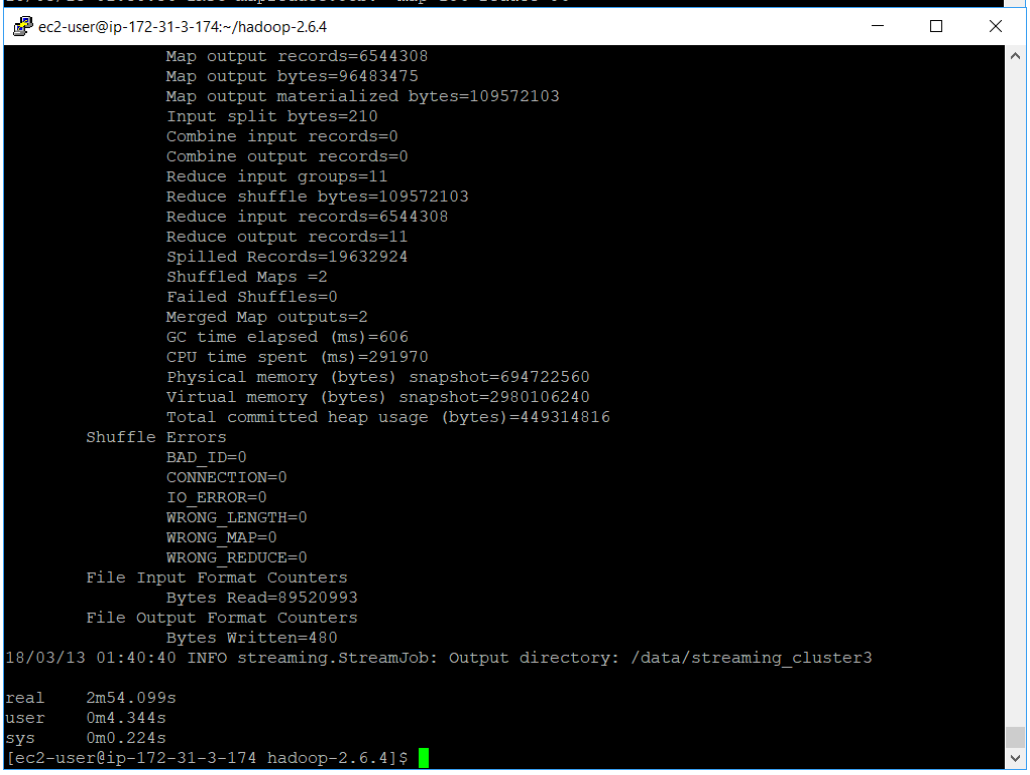
Third Hadoop Streaming Command Run:

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/testdata/ -output /data/streaming_cluster3 -mapper cluster_mapper.py -reducer cluster_reducer.py -file /home/ec2-user/cluster_mapper.py -file /home/ec2-user/cluster_reducer.py -file /home/ec2-user/centers.txt

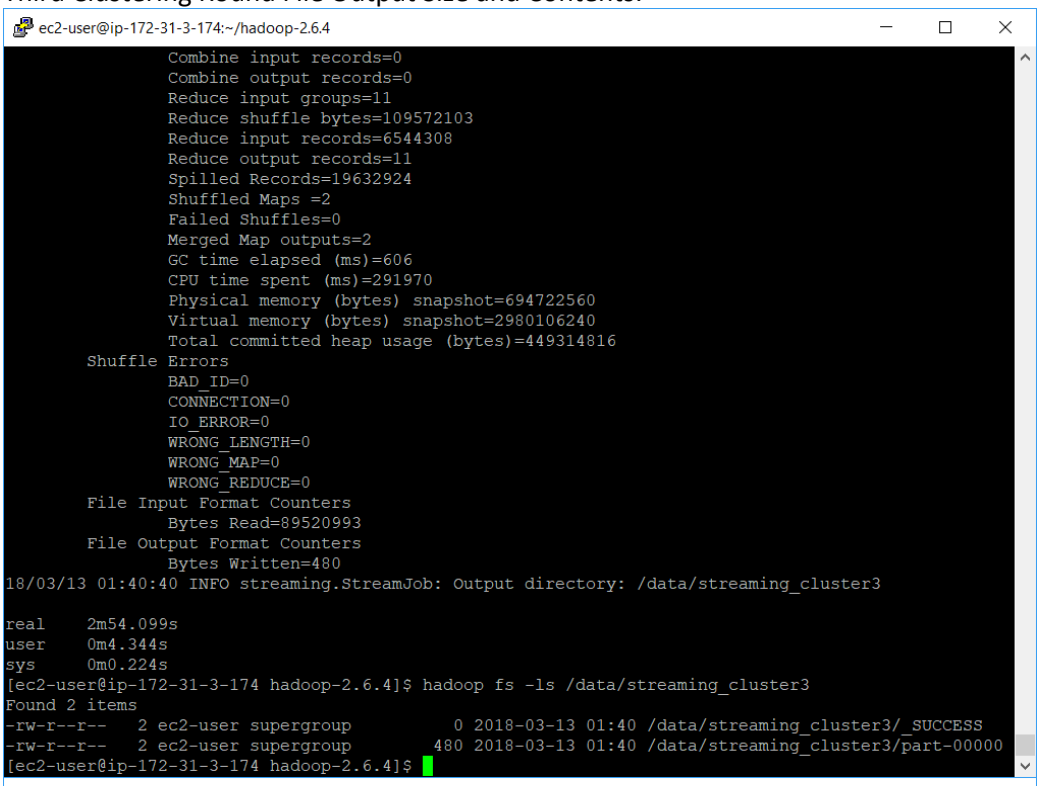
The only change to this command is the output folder being streaming_cluster3 instead of streaming_cluster2.

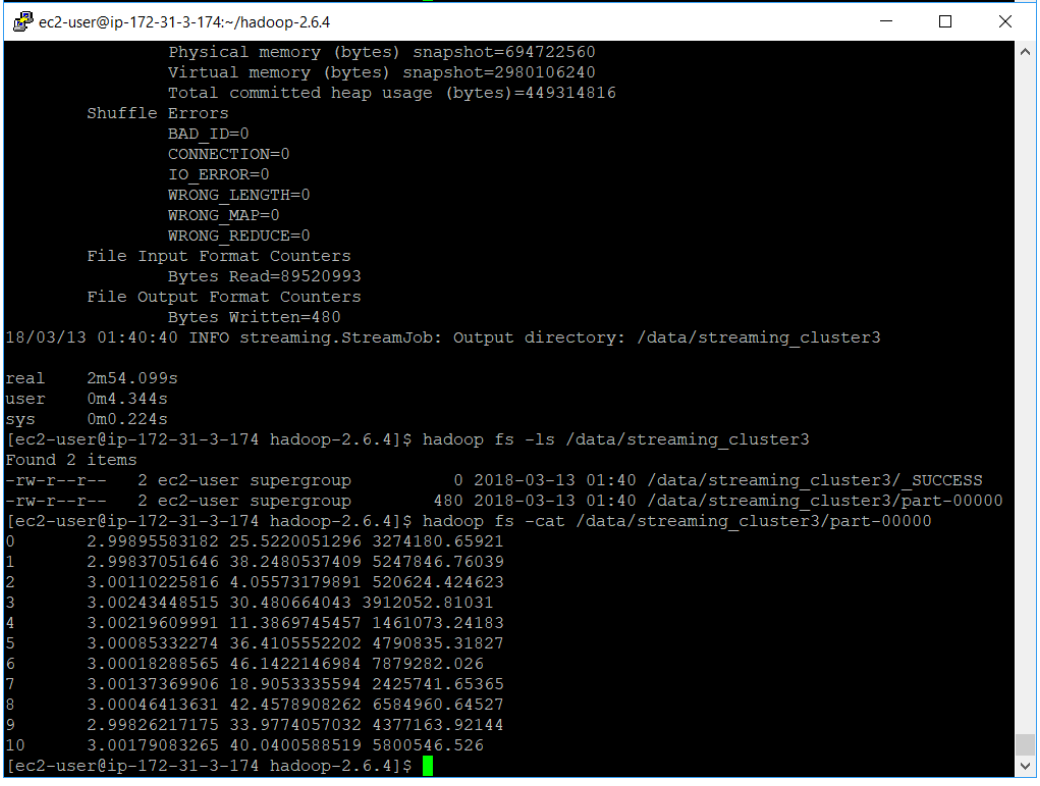
Third Clustering Round Execution (start and end shown due to length):



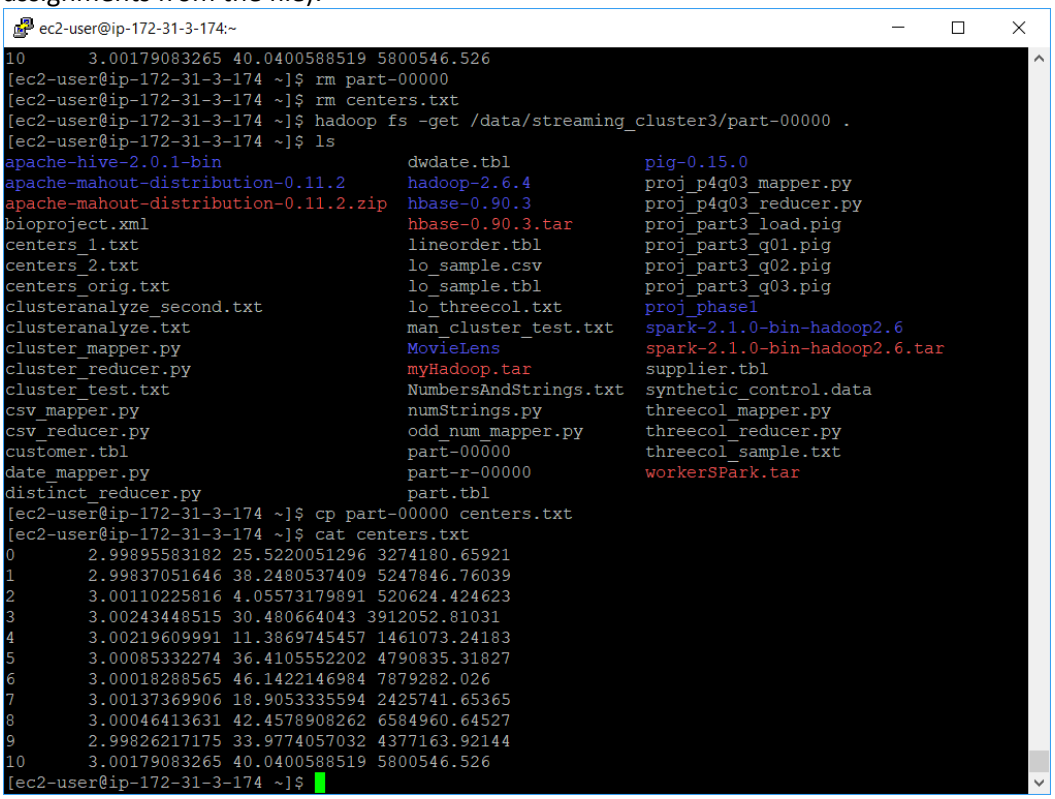


Third Clustering Round File Output Size and Contents:





Creation of new centers.txt file from the third round clustering output (including nano to remove center assignments from the file):



Part 4

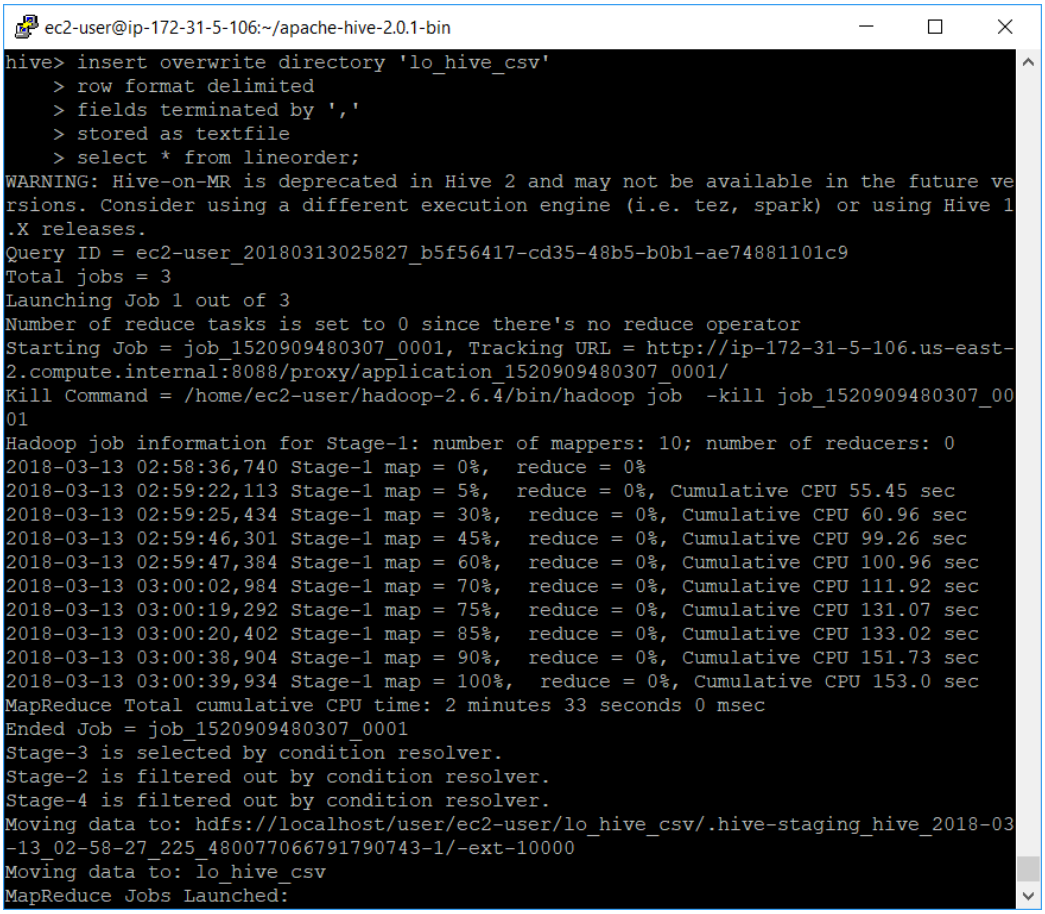
I ran all of the Pig, Hive, and Hadoop Streaming commands on a 1 node cluster to compare to the 4-node results. I did not include the code I used in this section because it is the same that was used in part 1 (thought I would make this section more compact and hopefully easier to read). I also did not include the screenshots of the output directory sizes or contents, since these also are the same as in part 1. I downloaded all of the scale 4 tables to the 1 node cluster Linux ec2-user home directory using wget. I then copied them into the HDFS /user/ec2-user/ directory. I used the same 1-node cluster that was used for assignments 1 through 3. It was a t2 medium instance with 12 GB hard drive space. The HDFS replication factor was set to 1 (since there was only 1 node).

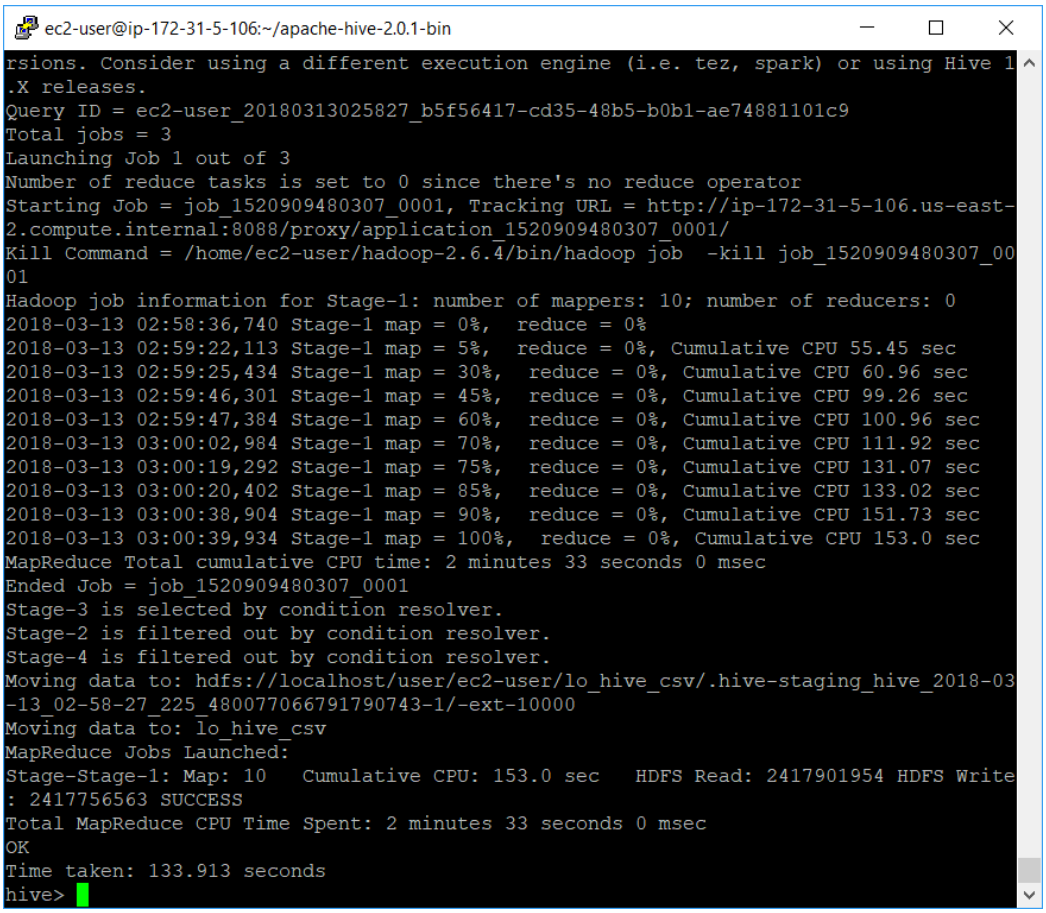
Note that I had issues with Pig running after I had run the Hadoop Streaming and Hive code and had to reformat the HDFS in order to get Pig to work (ran a small vehicles data query to check, which failed). I then found that I couldn't run Hive code after the Pig code had worked and needed to reformat the HDFS again to get Hive to work again. I'm not sure what the issue is with the 1 node cluster because it worked fine for assignments 1 through 3. It is possible that the hard drive space was not adequate (was only set to 12G). Since I had already reformatted the HDFS to fix Pig and then Hive, I was not able to check the size present during the failing cases. Since the lineorder.tbl was a couple gigabytes in size (present in both the Linux file system and HDFS), and the outputs of the Hadoop Streaming and Hive csv file transformations were each a couple gigabytes, it is possible the issue was that the HDFS was running out of space.

A)

1 Node Hive lineorder.tbl Transformation From | Separated to CSV

1 Node Hive Transformation Execution:

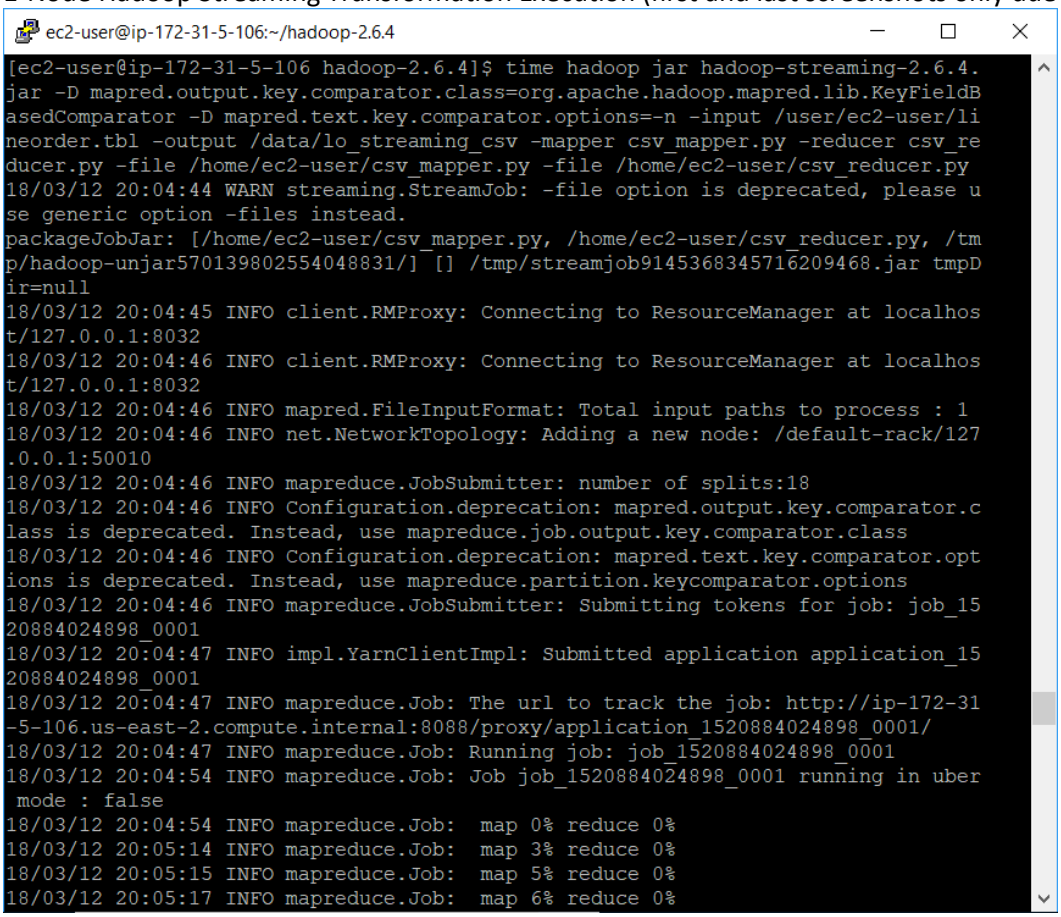


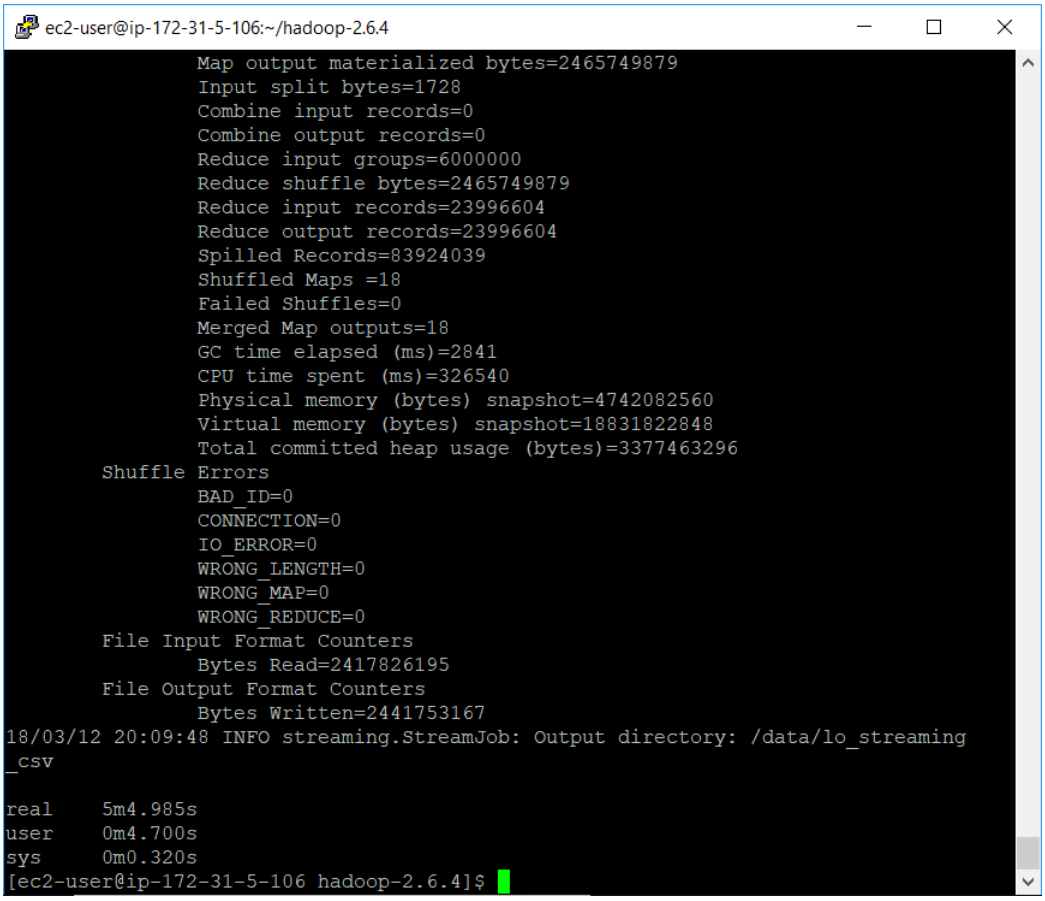


The Hive transformation code took 133.913 secs to execute on the 1-node cluster. This is compared to the 50.157 secs it took on a 4-node cluster. It took 2.67 times longer to run on the 1-node cluster than the 4-node one.

1-Node Hadoop Streaming lineorder.tbl Transformation From | Separated to CSV

1-Node Hadoop Streaming Transformation Execution (first and last screenshots only due to length):

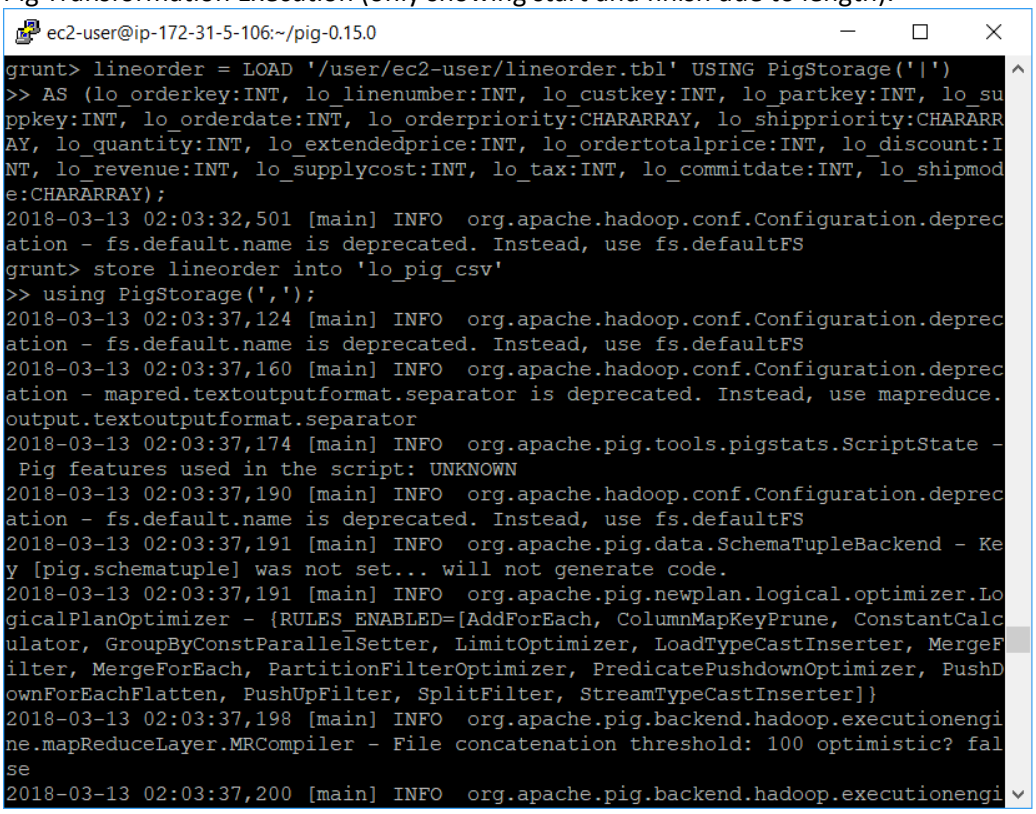


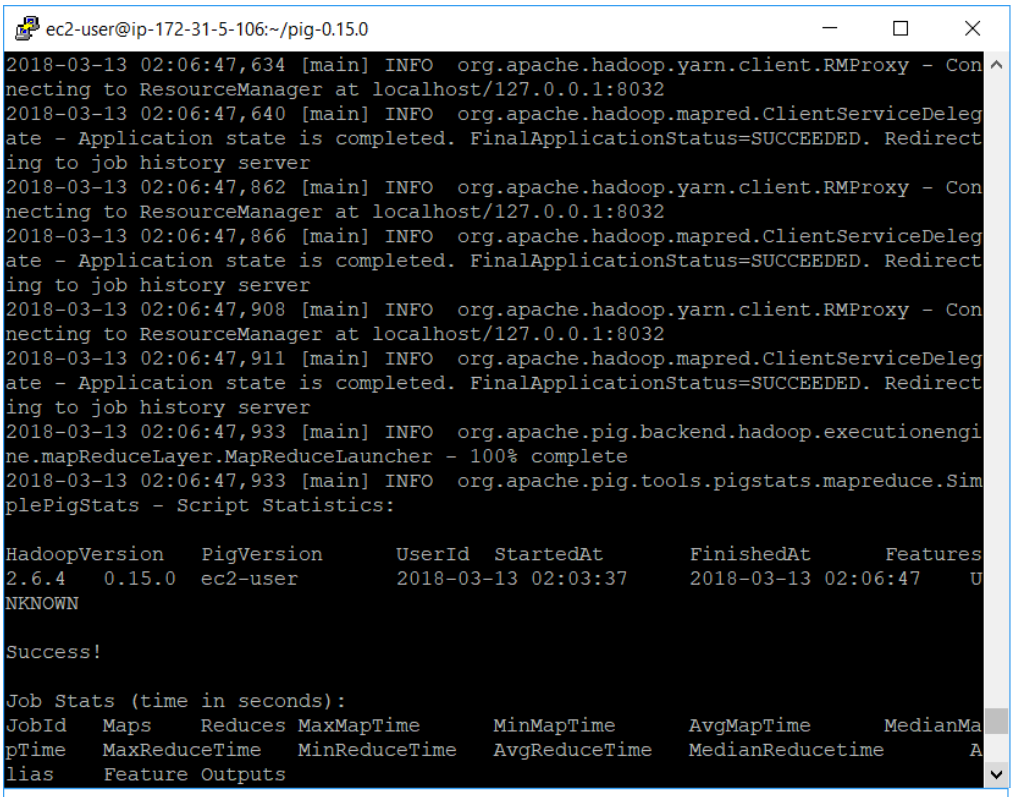


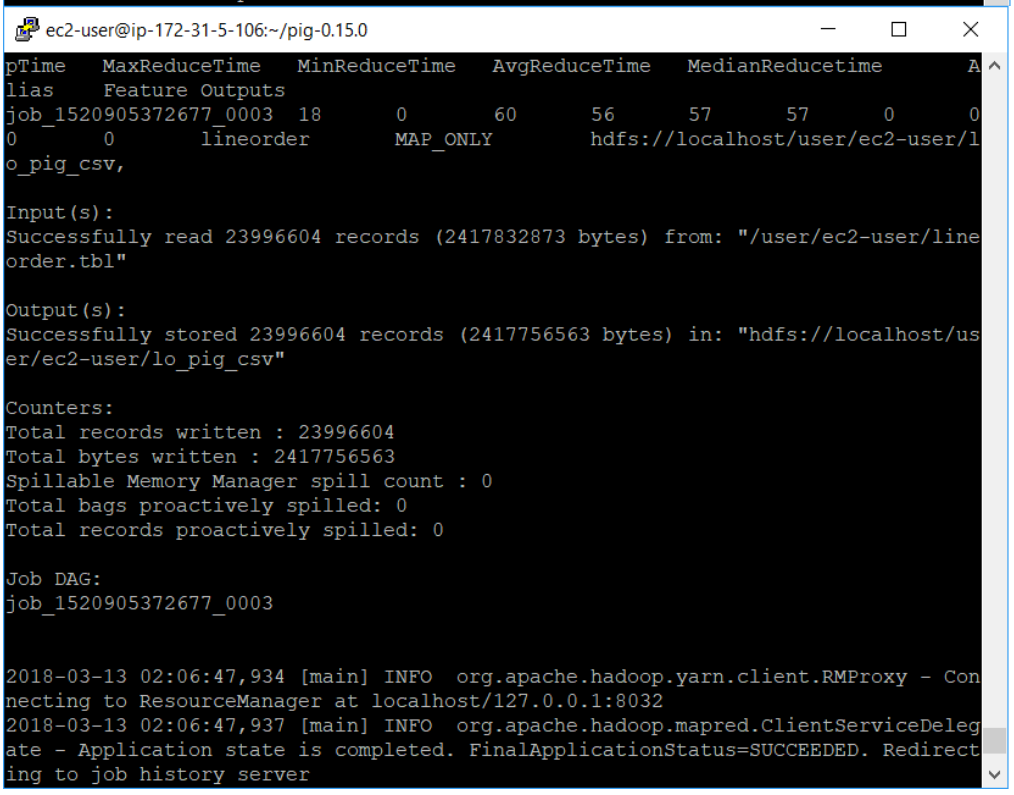
The Hadoop Streaming transformation code took 5 min 4.985 secs (or 304.985 secs) to execute on the 1-node cluster. This is compared to the 3 min and 17.27 secs (or 197.27 secs) on the 4-node cluster. It took 1.54 times longer to run on the 1-node cluster than the 4-node one.

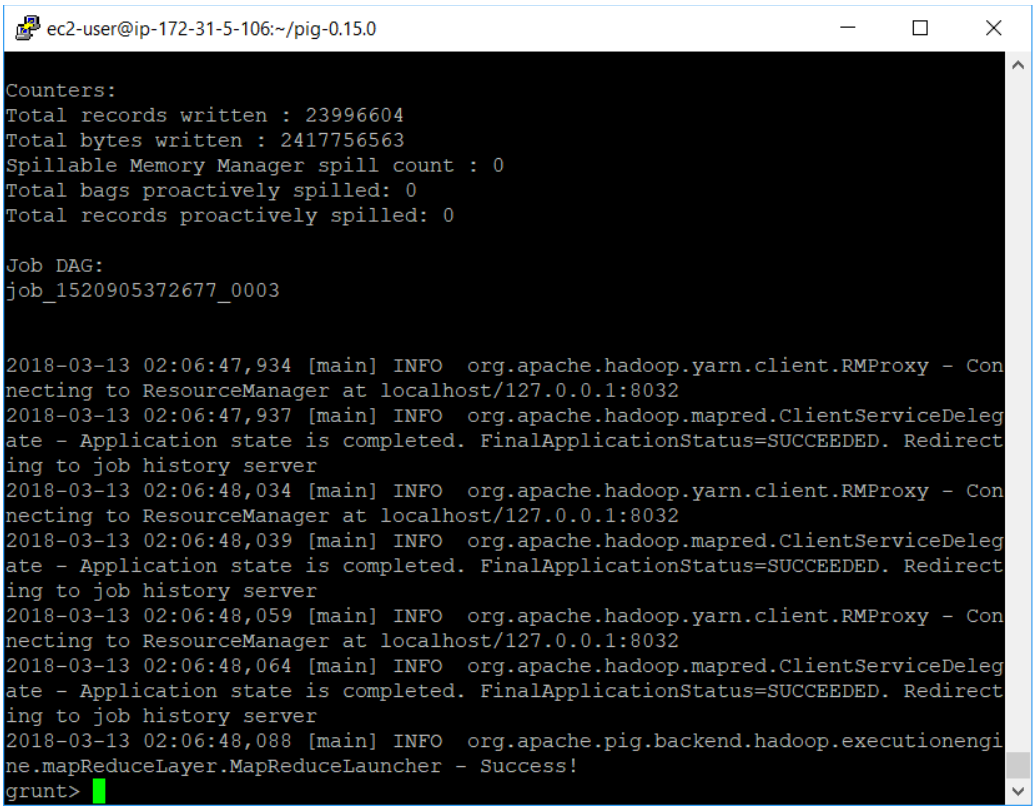
1-Node Pig lineorder.tbl Transformation From | Separated to CSV

Pig Transformation Execution (only showing start and finish due to length):




ec2-user@ip-172-31-5-106:~/pig-0.15.0
2018-03-13 02:06:47,634 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at localhost/127.0.0.1:8032
2018-03-13 02:06:47,640 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2018-03-13 02:06:47,862 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at localhost/127.0.0.1:8032
2018-03-13 02:06:47,866 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2018-03-13 02:06:47,908 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at localhost/127.0.0.1:8032
2018-03-13 02:06:47,911 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2018-03-13 02:06:47,933 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 100% complete
2018-03-13 02:06:47,933 [main] INFO org.apache.pig.tools.pigstats.mapreduce.SimplePigStats - Script Statistics:
HadoopVersion PigVersion UserId StartedAt FinishedAt Features
2.6.4 0.15.0 ec2-user 2018-03-13 02:03:37 2018-03-13 02:06:47 UNKNOWN
Success!
Job Stats (time in seconds):
JobId Maps Reduces MaxMapTime MinMapTime AvgMapTime MedianMapTime MaxReduceTime MinReduceTime AvgReduceTime MedianReducetime Alias Feature Outputs


ec2-user@ip-172-31-5-106:~/pig-0.15.0
pTime MaxReduceTime MinReduceTime AvgReduceTime MedianReducetime Alias Feature Outputs
job_1520905372677_0003 18 0 60 56 57 57 0 0 0 0 lineorder MAP_ONLY hdfs://localhost/user/ec2-user/lo_pig_csv,
Input(s):
Successfully read 23996604 records (2417832873 bytes) from: "/user/ec2-user/lineorder.tbl"
Output(s):
Successfully stored 23996604 records (2417756563 bytes) in: "hdfs://localhost/user/ec2-user/lo_pig_csv"
Counters:
Total records written : 23996604
Total bytes written : 2417756563
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
Job DAG:
job_1520905372677_0003
2018-03-13 02:06:47,934 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at localhost/127.0.0.1:8032
2018-03-13 02:06:47,937 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server

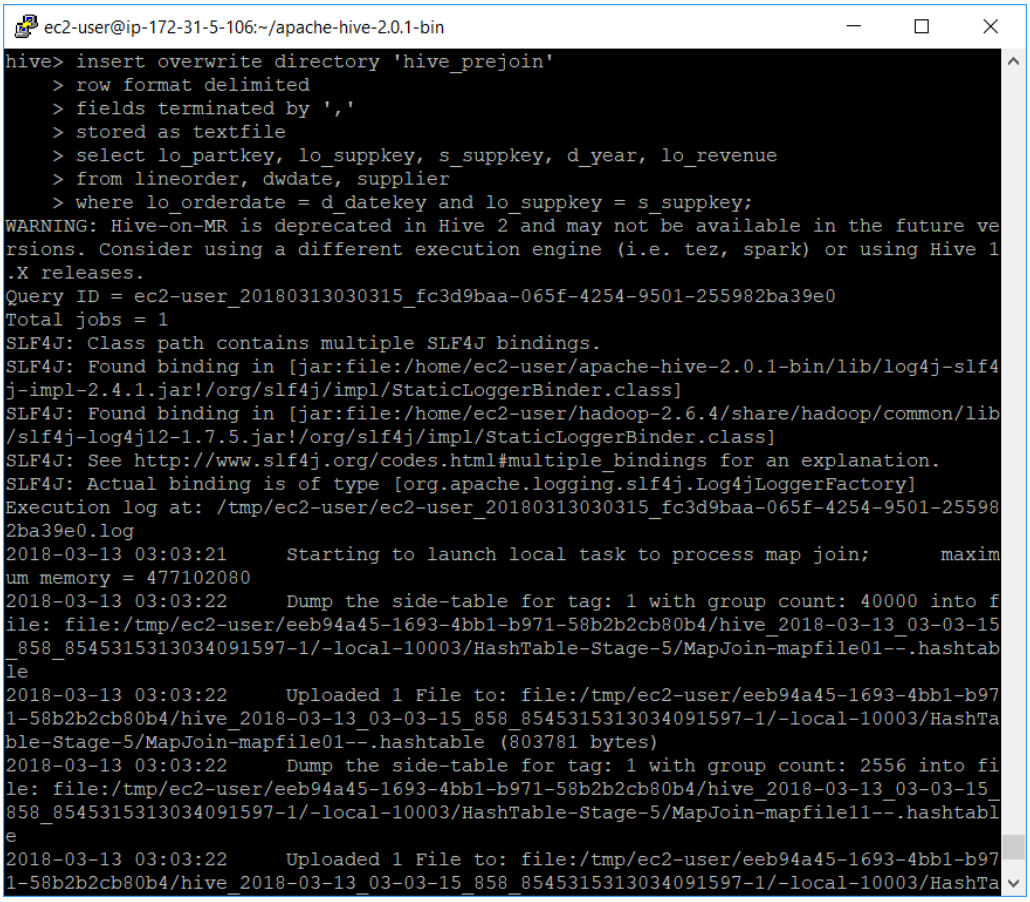


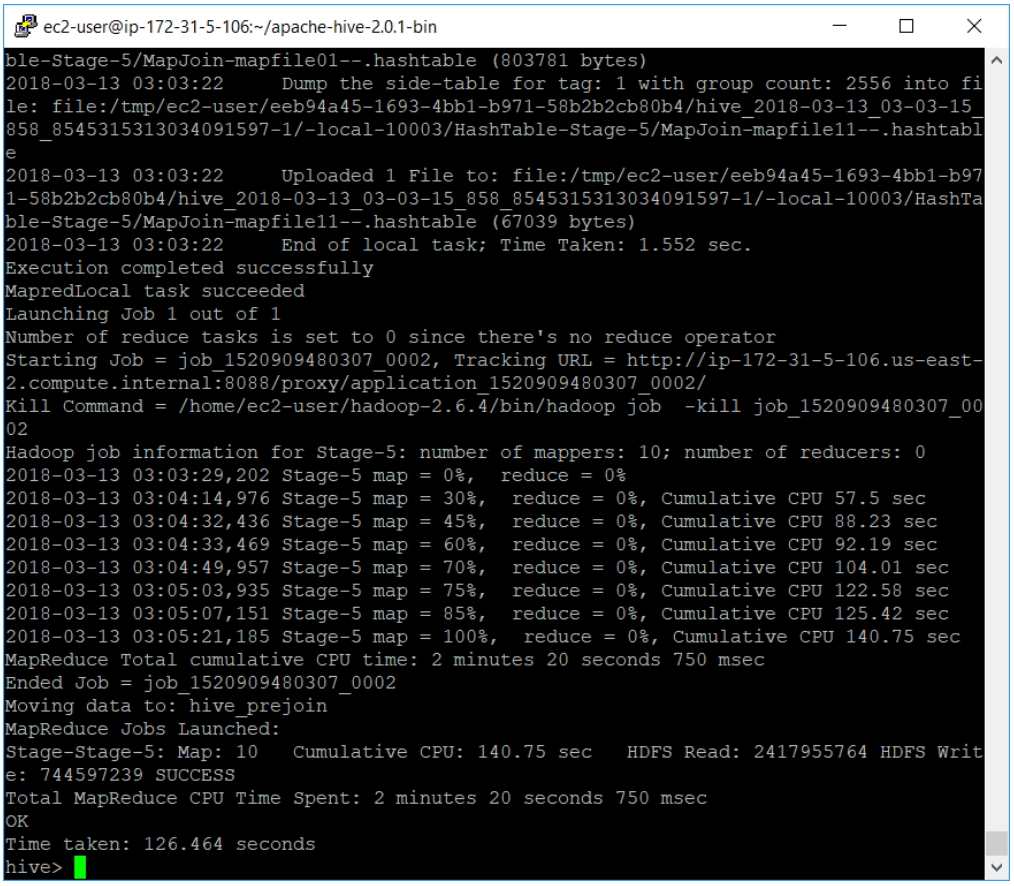
The Pig transformation code started at 2:03:37 and ended at 2:06:47, so it took 3 min 10 secs (or 190 secs) to execute on the 1-node cluster. This is in comparison to the 1 min 13 secs (or 73 secs) it took on a 4-node cluster. It took 2.6 times longer to run on the 1-node cluster than the 4-node one.

B)

1-Node Hive Pre-Join

1-Node Hive Pre-Join Execution:

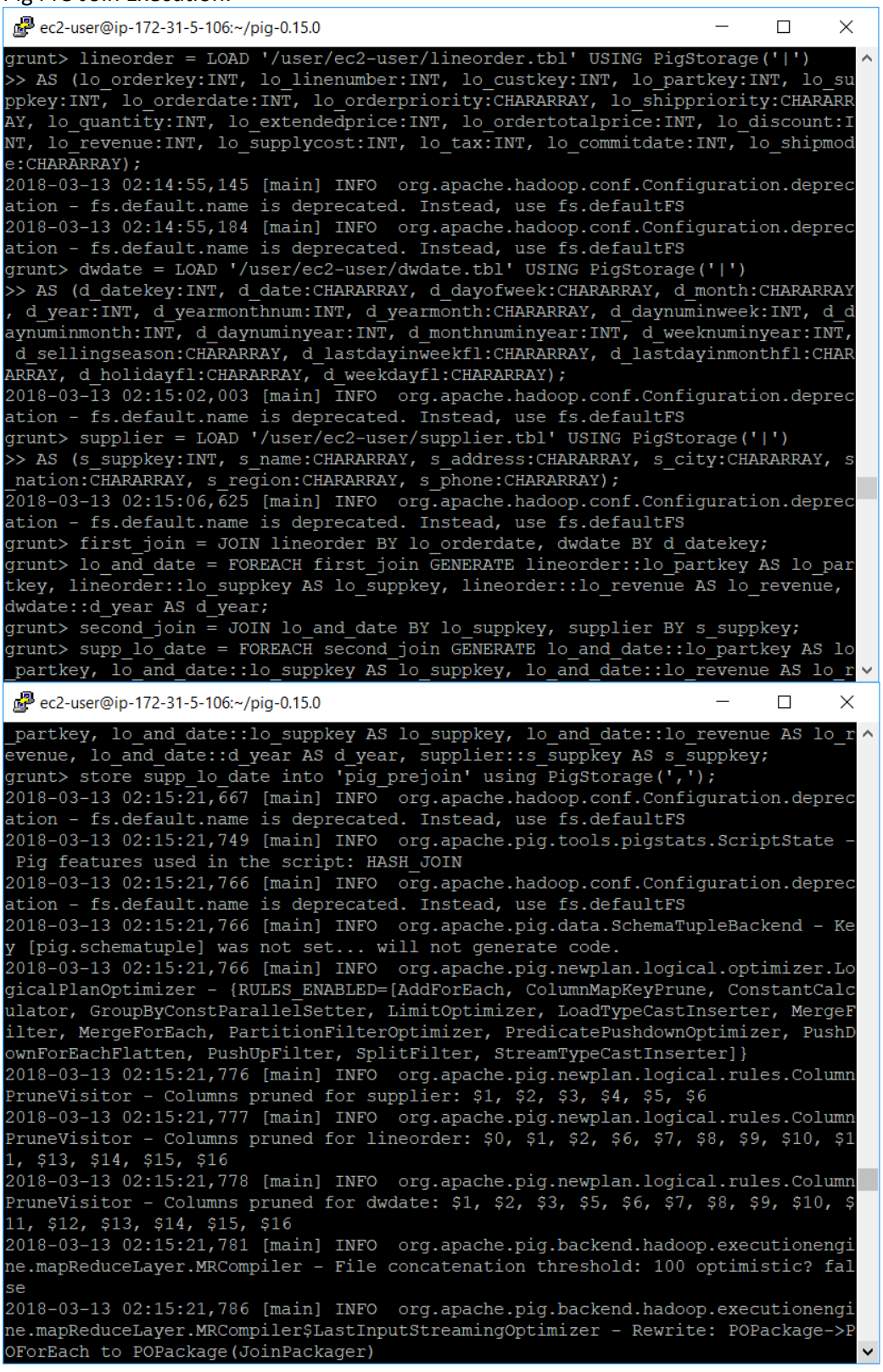


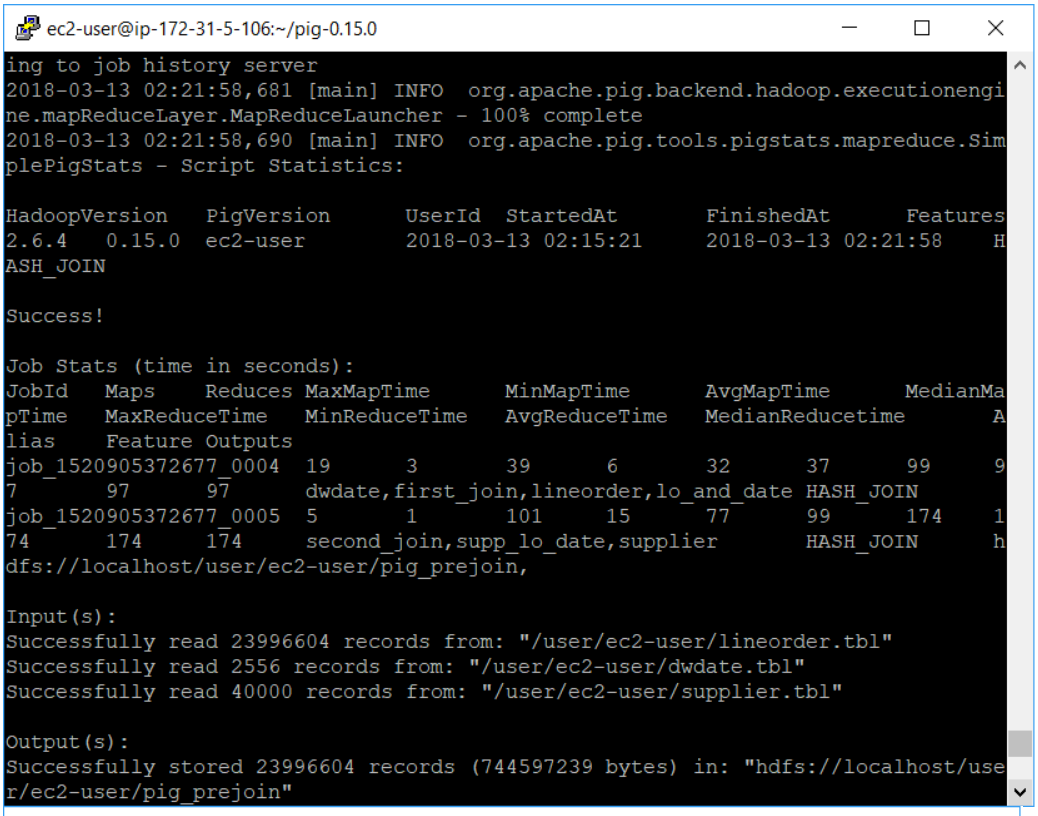


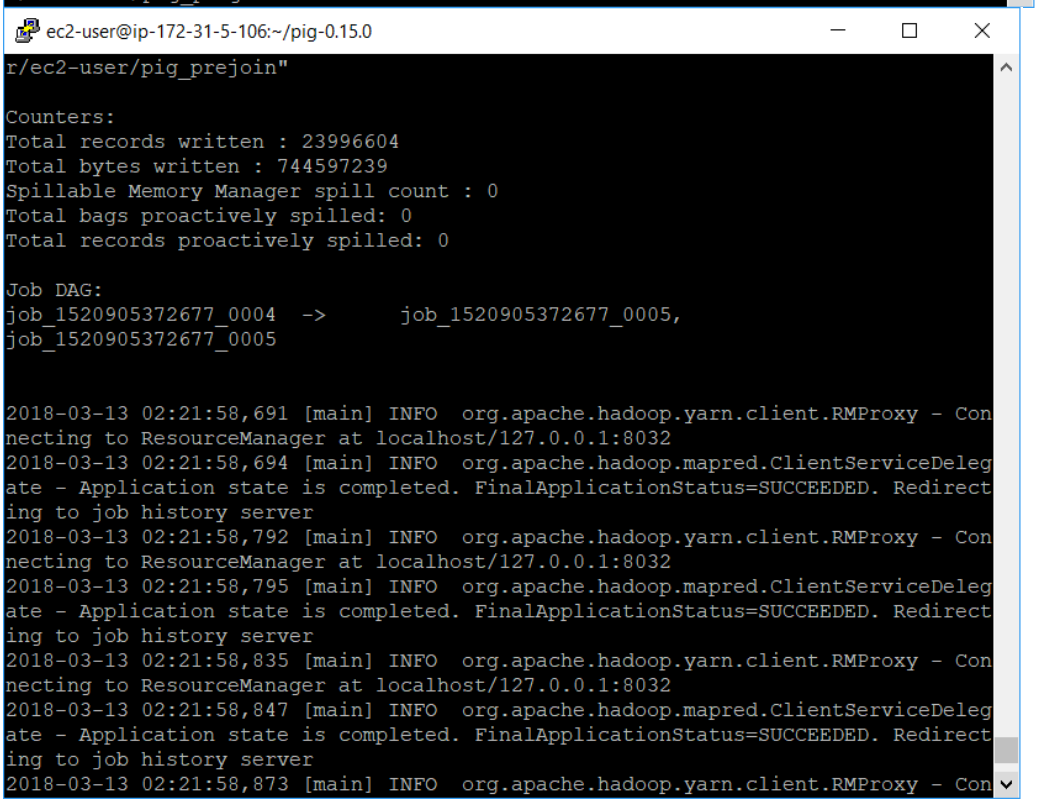
The Hive Pre-Join code took 126.464 secs to execute on a 1-node cluster. This is compared to the 48.47 secs it took on a 4-node cluster. It took 2.61 times longer to run on the 1-node cluster than the 4-node one.

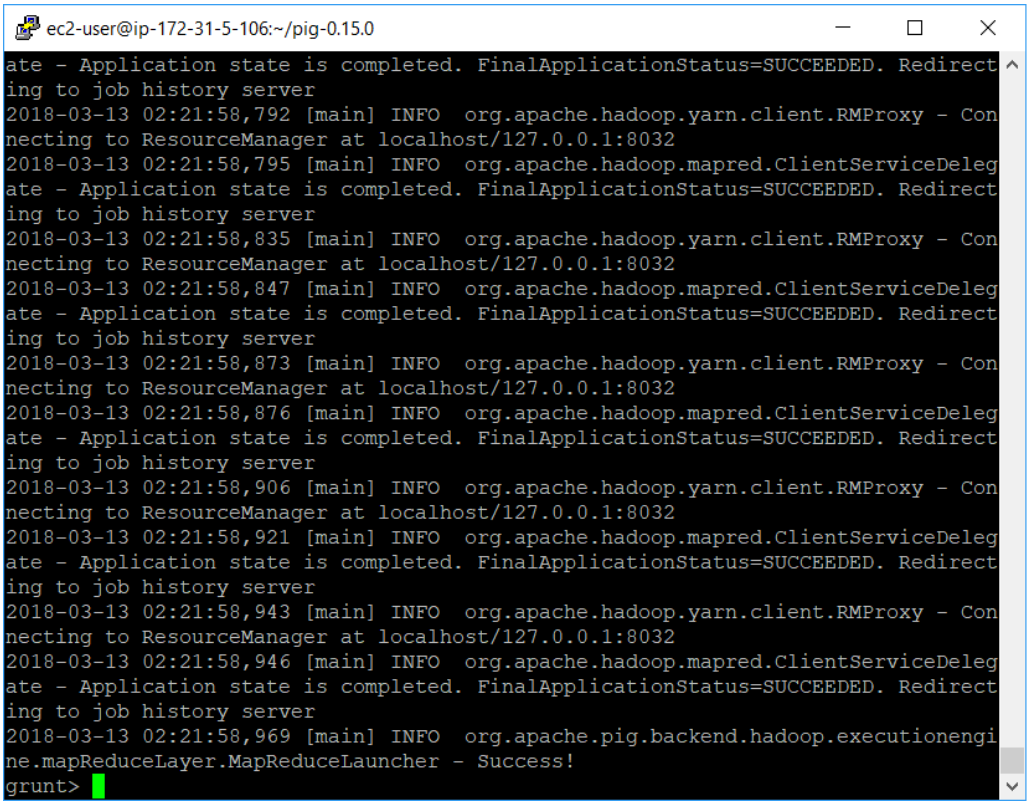
1-Node Pig Pre-Join

Pig Pre-Join Execution:




ec2-user@ip-172-31-5-106:~/pig-0.15.0
ing to job history server
2018-03-13 02:21:58,681 [main] INFO org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - 100% complete
2018-03-13 02:21:58,690 [main] INFO org.apache.pig.tools.pigstats.mapreduce.Sim
plePigStats - Script Statistics:
HadoopVersion PigVersion UserId StartedAt FinishedAt Features
2.6.4 0.15.0 ec2-user 2018-03-13 02:15:21 2018-03-13 02:21:58 H
ASH_JOIN
Success!
Job Stats (time in seconds):
JobId Maps Reduces MaxMapTime MinMapTime AvgMapTime MedianMa
pTime MaxReduceTime MinReduceTime AvgReduceTime MedianReducetime A
lias Feature Outputs
job_1520905372677_0004 19 3 39 6 32 37 99 9
7 97 97 dwdate,first_join,lineorder,lo_and_date HASH_JOIN
job_1520905372677_0005 5 1 101 15 77 99 174 1
74 174 174 second_join,supp_lo_date,supplier HASH_JOIN h
dfs://localhost/user/ec2-user/pig_prejoin,
Input(s):
Successfully read 23996604 records from: "/user/ec2-user/lineorder.tbl"
Successfully read 2556 records from: "/user/ec2-user/dwdate.tbl"
Successfully read 40000 records from: "/user/ec2-user/supplier.tbl"
Output(s):
Successfully stored 23996604 records (744597239 bytes) in: "hdfs://localhost/use
r/ec2-user/pig_prejoin"


ec2-user@ip-172-31-5-106:~/pig-0.15.0
r/ec2-user/pig_prejoin"
Counters:
Total records written : 23996604
Total bytes written : 744597239
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
Job DAG:
job_1520905372677_0004 -> job_1520905372677_0005,
job_1520905372677_0005
2018-03-13 02:21:58,691 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Con
necting to ResourceManager at localhost/127.0.0.1:8032
2018-03-13 02:21:58,694 [main] INFO org.apache.hadoop.mapred.ClientServiceDeleg
ate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirect
ing to job history server
2018-03-13 02:21:58,792 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Con
necting to ResourceManager at localhost/127.0.0.1:8032
2018-03-13 02:21:58,795 [main] INFO org.apache.hadoop.mapred.ClientServiceDeleg
ate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirect
ing to job history server
2018-03-13 02:21:58,835 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Con
necting to ResourceManager at localhost/127.0.0.1:8032
2018-03-13 02:21:58,847 [main] INFO org.apache.hadoop.mapred.ClientServiceDeleg
ate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirect
ing to job history server
2018-03-13 02:21:58,873 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Con



The Pig Pre-Join code started at 2:15:21 and ended at 2:21:59, so it took 6 min 38 secs (398 secs) or to execute on a 1-node cluster. This is compared to the 4 min 4 secs (or 244 secs) it took on a 4-node cluster. It took 1.63 times longer to run on the 1-node cluster than the 4-node one.

C)

The Hive portion of part 1A ran 2.67 times slower on the 1-node cluster than the 4-node cluster (133.913 secs versus 50.157 secs). The Pig portion of part 1A ran 2.6 times slower on the 1-node cluster than the 4-node cluster (190 secs versus 73 secs). The Hadoop Streaming portion of part 1A ran 1.54 times slower on the 1-node cluster versus the 4-node cluster (304.985 secs versus 197.27 secs). The Hive portion of part 2B ran 2.61 times slower on the 1-node cluster than the 4-node cluster (126.464 secs versus 48.47 secs). The Pig portion of part 2B ran 1.63 times slower on the 1-node cluster than the 4-node cluster (398 secs versus 244 secs). The run times of Hive on the 1-node cluster to 4-node cluster had very similar relations to each other between running part 1A and part 2B (both ran approximately 2.6 times slower on 1-node than 4-node). The Pig run times on the 1-node cluster and 4-node cluster had different relations to each other for part 1A and part 2B (part 1A ran 2.6 times slower on the 1-node versus 4-node where part 2B ran 1.63 times slower).

Running the tasks on 4-node clusters showed significant improvement in run time in all cases. The improvement, however, does not match the theoretical improvement predictions. Theoretically, without any network traffic delays, a 4-node cluster should run 4 times faster (a 1-node task taking 12 secs would take 12/4 = 3 secs and 12/3 equals 4 times faster). Since none of the 4-node tasks were close the theoretical speeds, this means that network traffic delay makes up a significant portion of all of the times. The network traffic delay is made up of the time for mappers to read from HDFS, the time for reducers to write to HDFS (HDFS write with replication factor of 2 for the 4-node cluster used), and network bus delay between the cluster nodes (for example the delay from having to send the mapper output from one node's local disk to another node that has the reducer required). Another factor is that the 1-node cluster was created with 12 GB of hard drive space, where each instance in the 4-node cluster had 30 GB. There was an issue running the Pig tasks after running the Hive tasks (Pig froze at 0% of the map/reduce stage on all queries, even simple ones on a smaller dataset). HDFS had to be reformatted between the Hive and Pig runs in order to get Pig to work. When the Hive tasks were run again after Pig was run, the same behavior was seen in Hive (was frozen at start of map/reduce). HDFS had to be reformatted again to get Hive to run again. This behavior was most likely due to the fact that the 1-node cluster only had 12 GB of hard drive space. Each output of part 1A was approximately 2.4 GB (one for the Hadoop Streaming, one for Hive, and one for Pig). On top of that, the lineorder table was 2.4 GB in size as well. This table was present in the Linux file system and the HDFS. This means that after Hadoop Streaming and Hive were run, there was at least 2.4*4 = 9.6 GB of hard drive space on the instance taken (streaming output, Hive output, Linux table, and HDFS table = 4 items 2.4 GB in size). This doesn't include the approximately 80 MB each for the part 2B outputs. It is possible that as the 1-node instance hard drive started to get filled, writing and reading to its HDFS could have been affected (could have been slowed down). The 4-node cluster HDFS was only 29% full, so this wouldn't have been an issue for the 4-node cluster. The 4-node cluster however did have a replication factor of 2, so the time to write to the HDFS would be double the time of the 1-node (without actual network bus delays for writing between nodes).

This experiment did show that running on a 4-node cluster significantly improves run time, but unfortunately since all of the variables were not held constant (4-node instances had more hard drive space than 1-node, 4-node cluster had replication of 2 where 1-node had replication of 1), it is hard to determine how much of the improvement was due to the presence of more nodes in the cluster alone. In other words, it is hard to determine exactly how increase in cluster size affected run time due to the other mitigating factors.