**CSC 555 Phase 5**

**Yiyang Yang**

**Part 1**

A.

Hive:

CREATE TABLE lineorder (

lo\_orderkey int,

lo\_linenumber int,

lo\_custkey int,

lo\_partkey int,

lo\_suppkey int,

lo\_orderdate int,

lo\_orderpriority String,

lo\_shippriority String,

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 String)

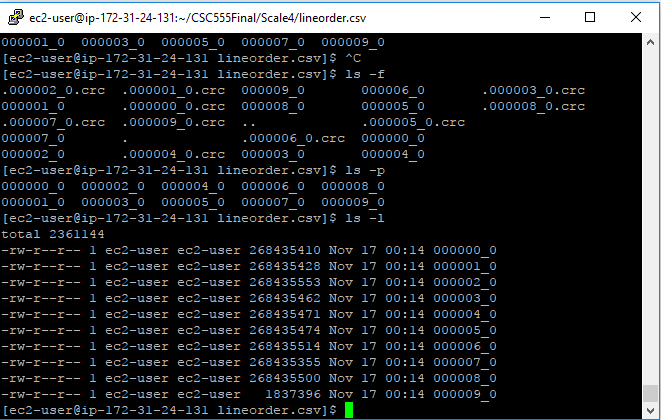
ROW FORMAT DELIMITED FIELDS TERMINATED BY '|';

LOAD DATA LOCAL INPATH '/home/ec2-user/CSC555Final/Scale4/lineorder.tbl' OVERWRITE INTO TABLE lineorder;

INSERT OVERWRITE LOCAL DIRECTORY '/home/ec2-user/CSC555Final/Scale4/lineorder.csv'

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

SELECT \* FROM lineorder;



Pig:

lineorderData = LOAD '/FinalData/lineorder.tbl' USING PigStorage('|')

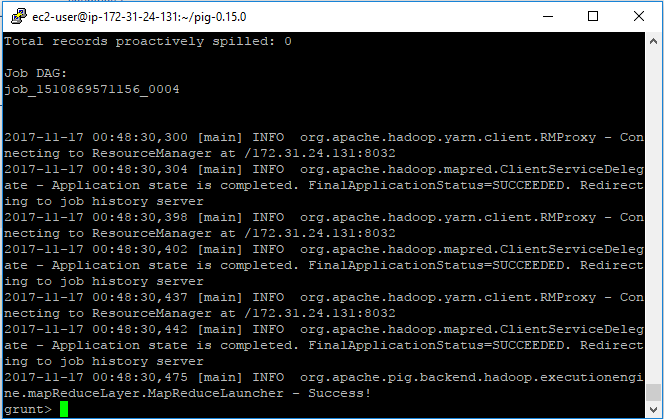
AS (lo\_orderkey:int, lo\_linenumber:int, lo\_custkey:int, lo\_partkey:int,

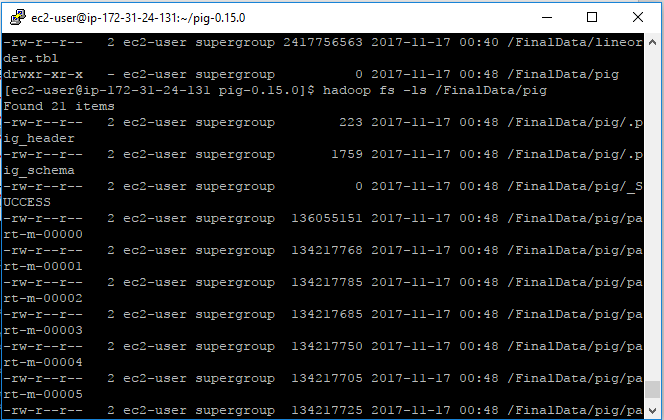
lo\_suppkey:int, lo\_orderdate:int, lo\_orderpriority:chararray, lo\_shippriority:chararray,

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 lineorderData INTO '/FinalData/pig' USING PigStorage('\t', '-schema');





Hadoop Streaming:

mapper4.py

#!/usr/bin/python

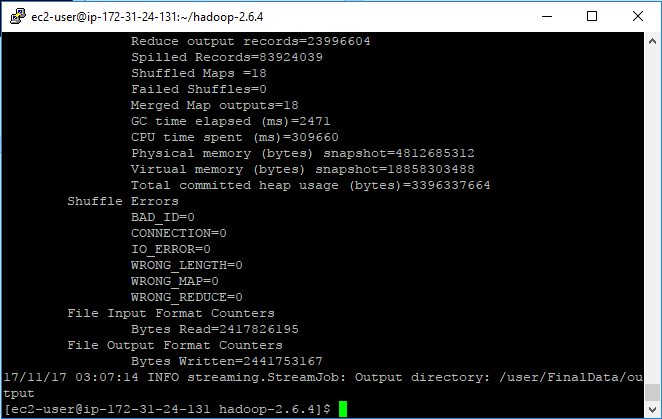
import sys

for line in sys.stdin:

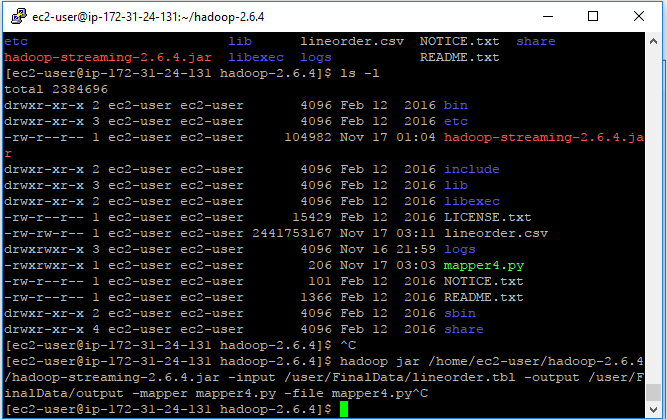
w = line.strip().split('|')

print w[0],w[1],w[2],w[3],w[4],w[5],w[6],w[7],w[8],w[9],w[10],w[11],w[12],w[13],w[14],w[15],w[16]

hadoop jar /home/ec2-user/hadoop-2.6.4/hadoop-streaming-2.6.4.jar -input /user/FinalData/lineorder.tbl -output /user/FinalData/output -mapper mapper4.py -file mapper4.py



hadoop fs -cat /user/FinalData/output/part-00000 > lineorder.csv



B:

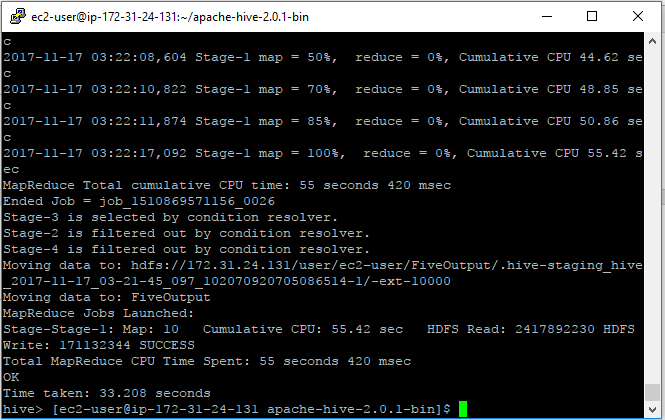
Hive:

INSERT OVERWRITE DIRECTORY 'FiveOutput'

SELECT lo\_orderkey, lo\_linenumber, lo\_partkey, lo\_custkey, lo\_quantity

FROM lineorder

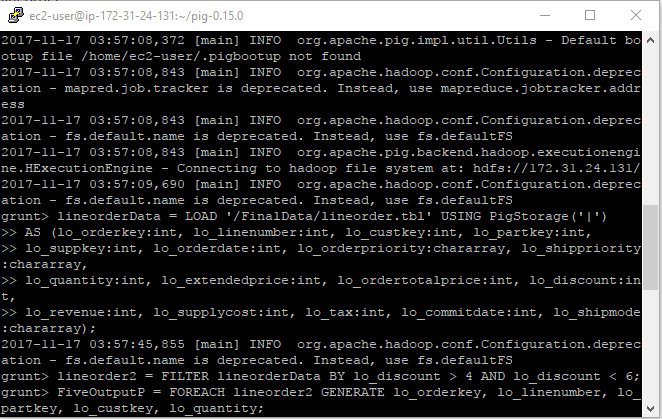
WHERE lo\_discount BETWEEN 4 AND 6;



Pig:

lineorder2 = FILTER lineorderData BY lo\_discount > 4 AND lo\_discount < 6;

FiveOutputP = FOREACH lineorder2 GENERATE lo\_orderkey, lo\_linenumber, lo\_partkey, lo\_custkey, lo\_quantity;



C:

Hive:

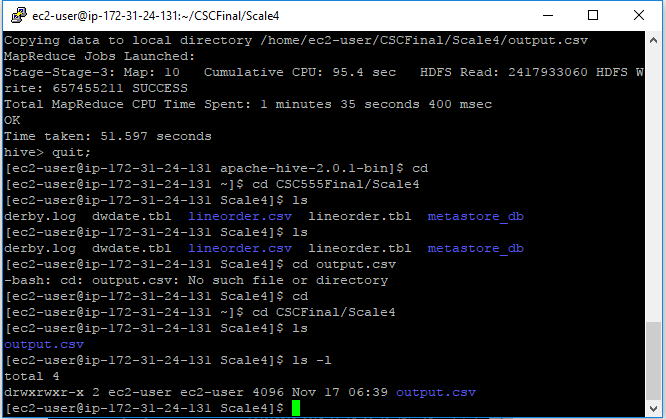
INSERT OVERWRITE LOCAL DIRECTORY '/home/ec2-user/CSCFinal/Scale4/output.csv'

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

SELECT lo\_partkey, lo\_suppkey, lo\_discount, d\_year, lo\_revenue

FROM lineorder, dwdate

WHERE lo\_orderdate = d\_datekey;



Pig:

lData = LOAD '/FinalData/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);

dData = LOAD '/FinalData/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,

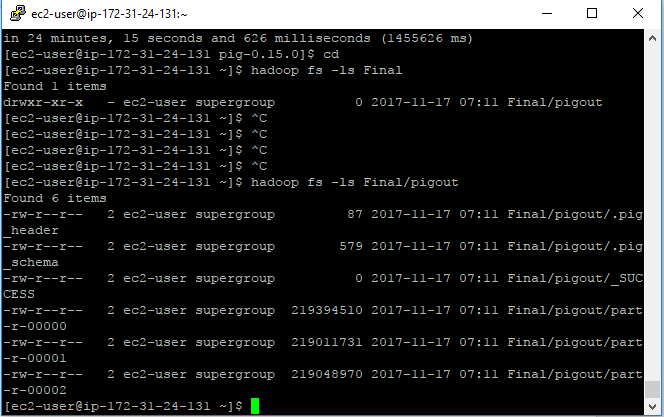
>> d\_holidayfl:chararray, d\_weekdayfl:chararray);

joinD = JOIN lData BY lo\_orderdate LEFT OUTER, dData BY d\_datekey;

out = FOREACH joinD GENERATE lo\_partkey, lo\_suppkey,

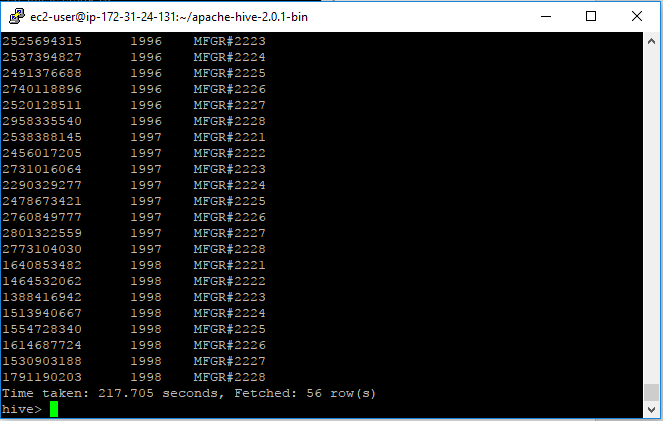
lo\_discount, d\_year, lo\_revenue;

STORE out INTO 'Final/pigout' USING PigStorage('\t', '-schema');

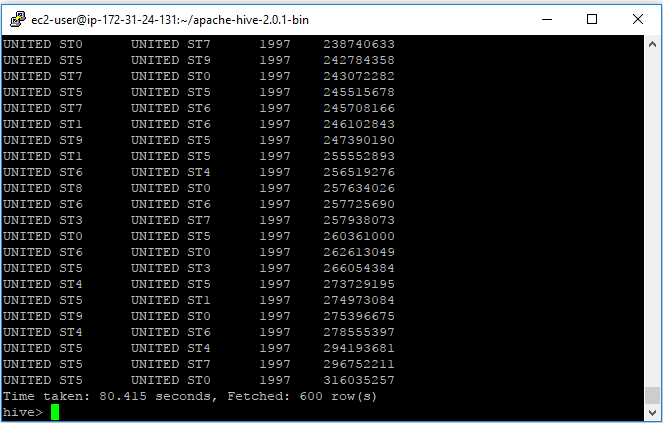


**Part 2**

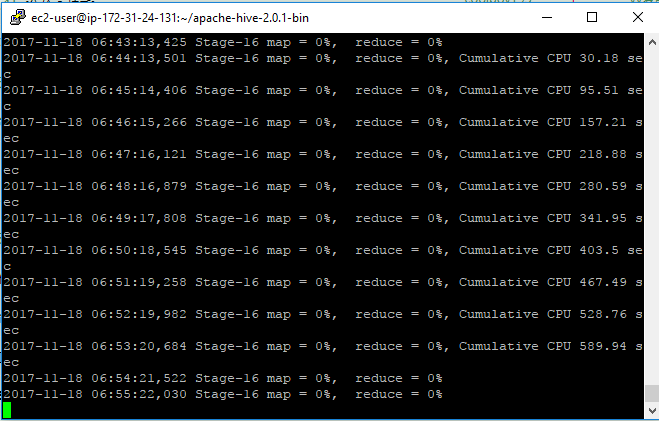
A: 2.2



3.2

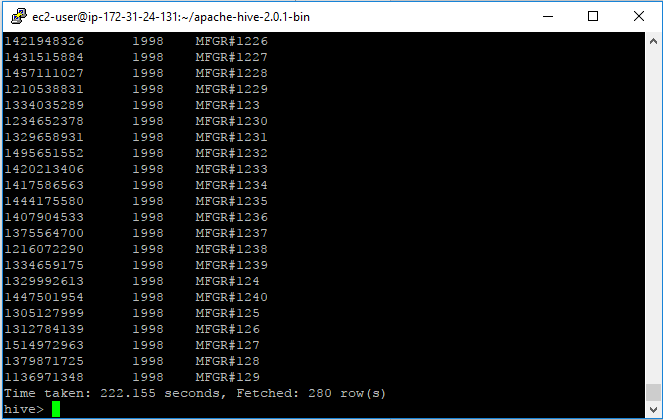


4.2

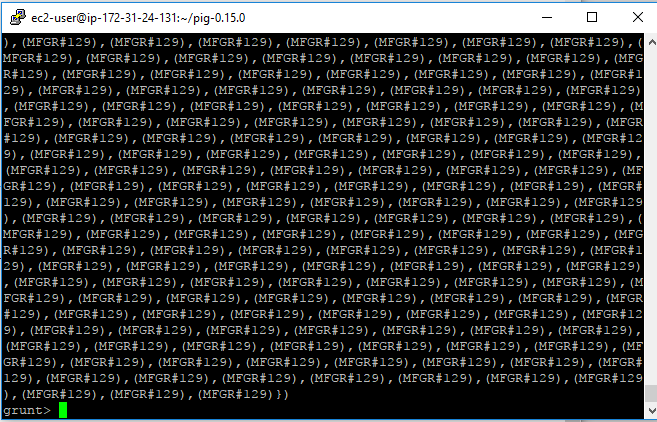


B.

Hive:

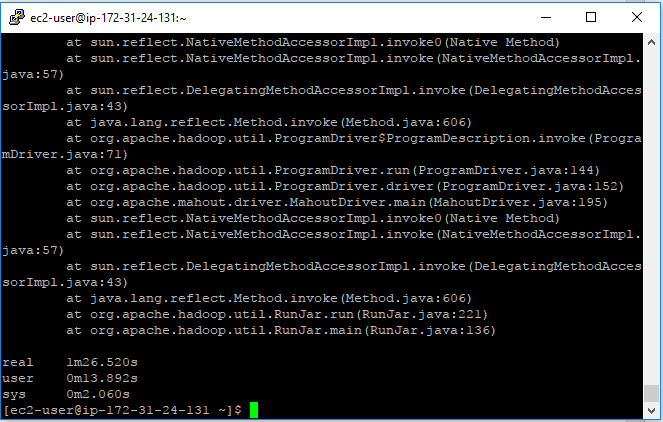


Pig:



**Part 3**

A.



Mapper:

#!/usr/bin/python

import sys

clusters = []

i=0

centers = []

center\_index=[]

def get\_center(v0, v1, v2, v3, v4):

n\_index = None

n\_distance = 9999999999

for c in centers:

dist = ((v0 - c[0])\*\*2 + (v1 - c[1])\*\*2 + (v2 - c[2])\*\*2 + (v3 - c[3])\*\*2 + (v4 - c[4])\*\*2 )\*\*0.5

if dist < n\_distance;

n\_index = i

n\_distance = dist

i += 1

return nearest\_c\_index

for line in sys.stdin:

line = line.strip()

vals = line.split(" ")

if len(vals) == 4:

clusters.append(vals)

else:

center\_index=vals

for n in center\_index:

centers.append(clusters[n])

n\_index = get\_center(vals[0], vals[1], vals[2], vals[3], vals[4])

**Part 4**

A.

a.

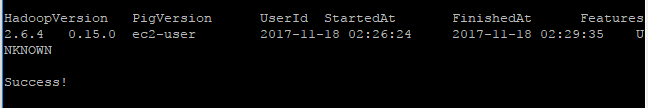
Hive 4 nodes:



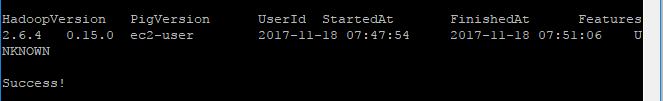
Hive single node:



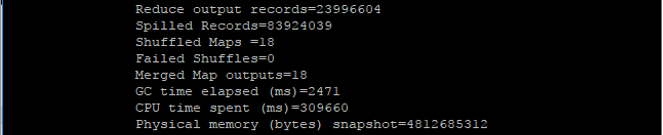
Pig 4 nodes:



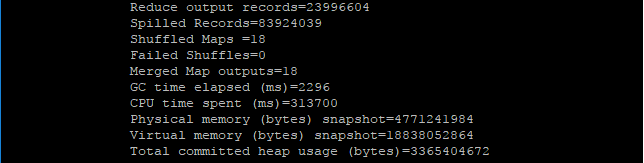
Pig single node:



Hadoop Streaming 4 nodes:



Hadoop Streaming single node:

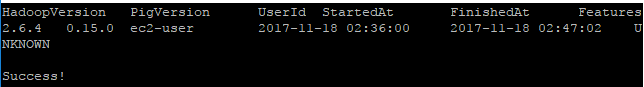


b.

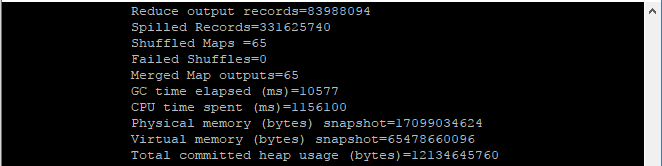
Hive:



Pig:



Hadoop Streaming:



B.

a.

Hive 4 Nodes:



Hive Single node:



Pig 4 Nodes:



Pig single node:



It looks like more nodes can reduce the run time. Larger dataset with multiple nodes will be much faster, but small dataset with single node will be much faster. I think assign the jobs and blocks will need more time.