CSC 555: Mining Big Data

Project, Phase 2 (due Sunday, November 19th)

In this part of the project, you will various queries using Hive, Pig and Hadoop streaming. The schema is available below, but do remember that schema should specify the delimiter:
http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/SSBM1/SSBM_schema_hive.sql

The data is available at (note that the data is |-separated, not comma separated):
http://rasinsrv07.cstcis.cti.depaul.edu/CSC553/data/ (this is Scale4)
http://rasinsrv07.cstcis.cti.depaul.edu/CSC553/data/Scale14/ (This is Scale14, larger version)

Please note what instance and what cluster you are using (you can reuse your existing cluster for most of the questions).
Please be sure to submit all code (pig and python and Hive). You should also submit the command lines you use and a screenshot of a completed run (just the last page, do not worry about capturing the whole output). You can use time command to record time of execution of anything you run.
I highly recommend creating a small sample input (e.g., by running head lineorder.tbl > lineorder.tbl.sample and testing your code with it, you can use head -n 100 to get first 100 lines only).

Part 1: Data Transformation

Using Scale4 data perform the following data processing.

A. Transform lineorder.tbl table into a csv (comma-separated file): Use Hive, MapReduce with HadoopStreaming and Pig (i.e. 3 different solutions)

Hive Steps (Solution 1):

1.
wget http://rasinsrv07.cstcis.cti.depaul.edu/CSC553/data/lineorder.tbl
2.
head –n100 lineorder.tbl > lineorder.tbl.sample (For code testing)
3.

```
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 '|';
```

4.

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

5.

```
ADD FILE /home/ec2-user/project_final_1a.py;
```

Python Code

```
#!/usr/bin/env python
import sys
#Reading from terminal
for line in sys.stdin:
        words = line.strip()
        vals = words.split('|')
        #Result output to table
        print ','.join(vals)
```

6.

```
create table lineorder2 (
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 ',';
```

7.
INSERT OVERWRITE TABLE lineorder2
SELECT TRANSFORM
(lo_orderkey,lo_linenumber,lo_custkey,lo_partkey,lo_suppkey,lo_orderdate,lo_orderpriority,lo_shippriority,lo_quantity,lo_extendedprice,lo_ordertotalprice,lo_discount,lo_revenue,lo_supplycost,lo_tax,lo_commitdate,lo_shipmode)
USING 'python project_final_1a.py'
AS
(lo_orderkey,lo_linenumber,lo_custkey,lo_partkey,lo_suppkey,lo_orderdate,lo_orderpriority,lo_shippriority,lo_quantity,lo_extendedprice,lo_ordertotalprice,lo_discount,lo_revenue,lo_supplycost,lo_tax,lo_commitdate,lo_shipmode)
from lineorder;
8.
hadoop fs -get /user/hive/warehouse/lineorder2
9 (Execute in the lineorder2 directory)
cat 000000_0 000001_0 000002_0 000003_0 000004_0 000005_0 000006_0 000007_0 000008_0 000009_0 >> lineorder2.csv

Hive Output: (4- Node)

```
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1510651862836_0003, Tracking URL = http://ip-172-31-18-209.us-wes
t-1.compute.internal:8088/proxy/application_1510651862836_0003/
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job  -kill job_1510651862836_0
003
Hadoop job information for Stage-1: number of mappers: 10; number of reducers: 0
2017-11-14 10:02:55,267 Stage-1 map = 0%,  reduce = 0%
2017-11-14 10:03:09,652 Stage-1 map = 10%,  reduce = 0%, Cumulative CPU 38.99 sec
2017-11-14 10:03:26,254 Stage-1 map = 15%,  reduce = 0%, Cumulative CPU 161.45 sec
2017-11-14 10:03:29,538 Stage-1 map = 20%,  reduce = 0%, Cumulative CPU 181.37 sec
2017-11-14 10:03:30,867 Stage-1 map = 25%,  reduce = 0%, Cumulative CPU 188.73 sec
2017-11-14 10:03:34,157 Stage-1 map = 30%,  reduce = 0%, Cumulative CPU 208.24 sec
2017-11-14 10:03:38,598 Stage-1 map = 35%,  reduce = 0%, Cumulative CPU 239.07 sec
2017-11-14 10:03:40,772 Stage-1 map = 40%,  reduce = 0%, Cumulative CPU 257.9 sec
2017-11-14 10:03:46,297 Stage-1 map = 45%,  reduce = 0%, Cumulative CPU 284.21 sec
2017-11-14 10:03:47,387 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 295.95 sec
2017-11-14 10:03:49,616 Stage-1 map = 65%,  reduce = 0%, Cumulative CPU 310.12 sec
2017-11-14 10:03:52,966 Stage-1 map = 70%,  reduce = 0%, Cumulative CPU 326.28 sec
2017-11-14 10:03:55,142 Stage-1 map = 75%,  reduce = 0%, Cumulative CPU 329.84 sec
2017-11-14 10:04:17,276 Stage-1 map = 80%,  reduce = 0%, Cumulative CPU 401.17 sec
2017-11-14 10:04:19,430 Stage-1 map = 85%,  reduce = 0%, Cumulative CPU 402.54 sec
2017-11-14 10:04:20,459 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 408.9 sec
MapReduce Total cumulative CPU time: 6 minutes 48 seconds 900 msec
Ended Job = job_1510651862836_0003
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://172.31.18.209/user/hive/warehouse/lineorder2/.hive-staging_hi
ve_2017-11-14_10-02-49_316_984824217666851051-1/-ext-10000
Loading data to table default.lineorder2
MapReduce Jobs Launched:
Stage-Stage-1: Map: 10   Cumulative CPU: 408.9 sec   HDFS Read: 2417934850 HDFS Writ
e: 2417757419 SUCCESS
Total MapReduce CPU Time Spent: 6 minutes 48 seconds 900 msec
OK
Time taken: 92.592 seconds
hive>
```

```
[ec2-user@ip-172-31-18-209 ~]$ hadoop fs -cat /user/hive/warehouse/lineorder2/000000_0 | more
1,1,29521,310379,16546,19960102,5-LOW,0,17,2361912,18150369,4,2267435,83361,2,19960212,TRUCK
1,2,29521,134619,3259,19960102,5-LOW,0,36,5952996,18150369,9,5417226,99216,6,19960228,MAIL
1,3,29521,127400,1418,19960102,5-LOW,0,8,1141920,18150369,10,1027728,85644,2,19960305,REG AIR
1,4,29521,4263,18842,19960102,5-LOW,0,28,3268328,18150369,9,2974178,70035,6,19960330,AIR
1,5,29521,48054,32491,19960102,5-LOW,0,24,2404920,18150369,10,2164428,60123,4,19960314,FOB
1,6,29521,31269,27344,19960102,5-LOW,0,32,3840832,18150369,7,3571973,72015,2,19960207,MAIL
2,1,62402,212340,21314,19961201,1-URGENT,0,38,4758854,4996796,0,4758854,75139,5,19970114,RAIL
3,1,98653,8594,39169,19931014,5-LOW,0,45,6761655,22702464,6,6355955,90155,0,19940104,AIR
3,2,98653,38071,33331,19931014,5-LOW,0,49,4944443,22702464,10,4449998,60544,0,19931220,RAIL
3,3,98653,256897,28180,19931014,5-LOW,0,27,5005476,22702464,6,4705147,111232,7,19931122,SHIP
3,4,98653,58760,14834,19931014,5-LOW,0,2,343752,22702464,1,340314,103125,6,19940107,TRUCK
3,5,98653,366189,32134,19931014,5-LOW,0,28,3514476,22702464,4,3373896,75310,0,19940110,FOB
3,6,98653,124286,35308,19931014,5-LOW,0,26,3406728,22702464,10,3066055,78616,2,19931218,RAIL
4,1,109421,176070,36239,19951011,5-LOW,0,30,3438210,3601868,3,3335063,68764,8,19951214,REG AIR
5,1,35588,217139,834,19940730,5-LOW,0,15,1584180,11228843,2,1552496,63367,4,19940831,AIR
5,2,35588,247854,5180,19940730,5-LOW,0,26,4684784,11228843,7,4356849,108110,8,19940925,FOB
5,3,35588,75061,16661,19940730,5-LOW,0,50,5180300,11228843,8,4765876,62163,3,19941013,AIR
```

Hadoop Streaming Steps (Solution 2):

1.

hadoop fs -put linorder.tbl

2.

time hadoop jar /home/ec2-user/hadoop-2.6.4/share/hadoop/tools/lib/hadoop-streaming-2.6.4.jar -D mapred.output.key.comparator.class=org.apache.hadoop.mapred.lib.KeyFieldBasedComparator -input lineorder.tbl.1 -output /final1a_hs/ -mapper project_final_1a.py -file project_final_1a.py

Python Code (project_final_1a.py) store in local drive:

```
#!/usr/bin/env python
import sys
#Reading from terminal
for line in sys.stdin:
        words = line.strip()
        vals = words.split('|')
        #Result output to table
        print ','.join(vals)
```

Hadoop Streaming Output: (4- Node)

```
                Total time spent by all reduce tasks (ms)=114296
                Total vcore-milliseconds taken by all map tasks=928346
                Total vcore-milliseconds taken by all reduce tasks=114296
                Total megabyte-milliseconds taken by all map tasks=950626304
                Total megabyte-milliseconds taken by all reduce tasks=117039104
        Map-Reduce Framework
                Map input records=23996604
                Map output records=23996604
                Map output bytes=2441753167
                Map output materialized bytes=2489746483
                Input split bytes=1800
                Combine input records=0
                Combine output records=0
                Reduce input groups=23996604
                Reduce shuffle bytes=2489746483
                Reduce input records=23996604
                Reduce output records=23996604
                Spilled Records=83924039
                Shuffled Maps =18
                Failed Shuffles=0
                Merged Map outputs=18
                GC time elapsed (ms)=2601
                CPU time spent (ms)=237220
                Physical memory (bytes) snapshot=4819709952
                Virtual memory (bytes) snapshot=18846203904
                Total committed heap usage (bytes)=3379560448
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=2417826195
        File Output Format Counters
                Bytes Written=2441753167
17/11/14 09:56:01 INFO streaming.StreamJob: Output directory: /final1a_hs/

real    2m40.626s
user    0m4.180s
sys     0m0.184s
[ec2-user@ip-172-31-18-209 ~]$
```

```
[ec2-user@ip-172-31-18-209 ~]$ hadoop fs -cat /user/hive/warehouse/lineorder2/000000_0 | more
1,1,29521,310379,16546,19960102,5-LOW,0,17,2361912,18150369,4,2267435,83361,2,19960212,TRUCK
1,2,29521,134619,3259,19960102,5-LOW,0,36,5952996,18150369,9,5417226,99216,6,19960228,MAIL
1,3,29521,127400,1418,19960102,5-LOW,0,8,1141920,18150369,10,1027728,85644,2,19960305,REG AIR
1,4,29521,4263,18842,19960102,5-LOW,0,28,3268328,18150369,9,2974178,70035,6,19960330,AIR
1,5,29521,48054,32491,19960102,5-LOW,0,24,2404920,18150369,10,2164428,60123,4,19960314,FOB
1,6,29521,31269,27344,19960102,5-LOW,0,32,3840832,18150369,7,3571973,72015,2,19960207,MAIL
2,1,62402,212340,21314,19961201,1-URGENT,0,38,4758854,4996796,0,4758854,75139,5,19970114,RAIL
3,1,98653,8594,39169,19931014,5-LOW,0,45,6761655,22702464,6,6355955,90155,0,19940104,AIR
3,2,98653,38071,33331,19931014,5-LOW,0,49,4944443,22702464,10,4449998,60544,0,19931220,RAIL
3,3,98653,256897,28180,19931014,5-LOW,0,27,5005476,22702464,6,4705147,111232,7,19931122,SHIP
```

Pig Steps (Solution 3):

Steps:

1.
```
lod = LOAD '/user/ec2-user/lineorder.tbl' USING PigStorage('|')
AS (lo_orderkey :float,
lo_linenumber :float,
lo_custkey :float,
lo_partkey :float,
lo_suppkey :float,
lo_orderdate :float,
lo_orderpriority :chararray,
lo_shippriority : chararray,
lo_quantity : chararray,
lo_extendedprice :float,
lo_ordertotalprice :float,
lo_discount :float,
lo_revenue :float,
lo_supplycost :float,
lo_tax :float,
lo_commitdate :float,
lo_shipmode :chararray);
```
2.
```
DESCRIBE lod
```
3.
```
lod2 = FOREACH lod GENERATE
lo_orderkey,lo_linenumber,lo_custkey,lo_partkey,lo_suppkey,lo_orderdate,lo_orderpriority,lo_shipprio
rity,lo_quantity,lo_extendedprice,lo_ordertotalprice,lo_discount,lo_revenue,lo_supplycost,lo_tax,lo_co
mmitdate,lo_shipmode;
```
4.
```
dump lod2;
```
5.
```
store lod2 into '/final1a_pig' USING PigStorage (',');
```

Pig Output:

```
Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime  M
edianReducetime Alias   Feature Outputs
job_1510651862836_0004  18      0       89      41      70      71      0       0       0       0       lod,lod2        MAP_ONLY       /
final1a_pig,

Input(s):
Successfully read 23996604 records (2417832945 bytes) from: "/user/ec2-user/lineorder.tbl"

Output(s):
Successfully stored 23996604 records (3106341603 bytes) in: "/final1a_pig"

Counters:
Total records written : 23996604
Total bytes written : 3106341603
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1510651862836_0004


2017-11-14 10:11:34,325 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /172.31.18.209:8032
2017-11-14 10:11:34,329 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationSt
atus=SUCCEEDED. Redirecting to job history server
2017-11-14 10:11:34,436 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /172.31.18.209:8032
2017-11-14 10:11:34,441 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationSt
atus=SUCCEEDED. Redirecting to job history server
2017-11-14 10:11:34,466 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /172.31.18.209:8032
2017-11-14 10:11:34,471 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationSt
atus=SUCCEEDED. Redirecting to job history server
2017-11-14 10:11:34,502 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
grunt>
```

```
[ec2-user@ip-172-31-18-209 ~]$ hadoop fs -cat /final1a_pig/part-m-00000 | more
2.2657156E7,1.0,115205.0,187139.0,1097.0,1.9930308E7,3-MEDIUM,0,8,980904.0,901548.0,9.0,892622.0,73567.0,1.0,1.
9930512E7,TRUCK
2.2657156E7,1.0,35278.0,229235.0,37662.0,1.9930724E7,5-LOW,0,42,4889724.0,4923951.0,5.0,4645237.0,69853.0,6.0,1
.9930928E7,RAIL
2.2657158E7,1.0,104888.0,340795.0,24361.0,1.9930712E7,1-URGENT,0,28,5140184.0,2.5095892E7,4.0,4934576.0,110146.
0,2.0,1.9930824E7,SHIP
2.2657158E7,2.0,104888.0,131333.0,23895.0,1.9930712E7,1-URGENT,0,47,6412351.0,2.5095892E7,10.0,5771115.0,81859.
0,2.0,1.9930928E7,RAIL
2.2657158E7,3.0,104888.0,100012.0,29888.0,1.9930712E7,1-URGENT,0,33,3339633.0,2.5095892E7,8.0,3072462.0,60720.0
,3.0,1.9930828E7,FOB
2.2657158E7,4.0,104888.0,89669.0,36648.0,1.9930712E7,1-URGENT,0,28,4644248.0,2.5095892E7,6.0,4365593.0,99519.0,
6.0,1.99309E7,SHIP
2.2657158E7,5.0,104888.0,252527.0,14331.0,1.9930712E7,1-URGENT,0,25,3698775.0,2.5095892E7,8.0,3402873.0,88770.0
,6.0,1.993082E7,TRUCK
2.2657158E7,6.0,104888.0,217181.0,14917.0,1.9930712E7,1-URGENT,0,20,2196340.0,2.5095892E7,5.0,2086523.0,65890.0
,8.0,1.9930832E7,FOB
```

B. Extract five of the numeric columns that for rows where lo_discount is between 4 and 6 into a space-separated text file (for K-Means clustering later). Use Hive and Pig (2 different solutions) (NOTE: you do not need to use your code to identify what is a numeric column, just go by what the data types say. You should manually pick any 5 columns that contain only numbers)

Hive(Solution 1): (4-nodes)

1.
```
INSERT OVERWRITE DIRECTORY '/fiveCol1b_hive'
row format delimited FIELDS TERMINATED BY ' '
select ((lo_discount - min_dis) / lo_dis_range) as dis,
((lo_quantity - min_qua) / lo_qua_range) as qua,
((lo_extendedprice - min_exprice) / lo_exprice_range) as exp,
((lo_ordertotalprice - min_otp) / lo_otp_range) as otp,
((lo_revenue - min_rev) / lo_rev_range) as rev
from (select lo_discount ,
MIN(lo_discount) over () as min_dis,
(MAX(lo_discount) over ()  -  MIN(lo_discount) over () ) as lo_dis_range,
lo_quantity,
MIN(lo_quantity) over () as min_qua,
(MAX(lo_quantity) over ()  -  MIN(lo_quantity) over () ) as lo_qua_range,
lo_extendedprice,
MIN(lo_extendedprice) over () as min_exprice,
(MAX(lo_extendedprice) over ()  -  MIN(lo_extendedprice) over () ) as lo_exprice_range,
lo_ordertotalprice,
MIN(lo_ordertotalprice) over () as min_otp,
(MAX(lo_ordertotalprice) over ()  -  MIN(lo_ordertotalprice) over () ) as lo_otp_range,
lo_revenue,
MIN(lo_revenue) over () as min_rev,
(MAX(lo_revenue) over ()  -  MIN(lo_revenue) over () ) as lo_rev_range
from lineorder
where lo_discount between 4.0 and 6.0 ) x;
```

2.
```
hadoop fs -copyToLocal /fiveCol1b_hive
```
3.
```
cd fiveCol1b_hive
```
4.
```
cat 000000_0 000001_0 000002_0 000003_0 000004_0 000005_0 000006_0 000007_0 000008_0
000009_0 >> fiveCol1bhive_one
```
5.
```
hadoop fs -put fiveCol1bhive_one
```

Hive Output:

```
Number of reduce tasks not specified. Estimated from input data size: 10
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_1510781966365_0009, Tracking URL = http://ip-172-31-18-209.us-west-1.com
pute.internal:8088/proxy/application_1510781966365_0009/
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job  -kill job_1510781966365_0009
Hadoop job information for Stage-1: number of mappers: 10; number of reducers: 10
2017-11-15 22:51:01,561 Stage-1 map = 0%,  reduce = 0%
2017-11-15 22:51:07,962 Stage-1 map = 10%,  reduce = 0%, Cumulative CPU 2.37 sec
2017-11-15 22:51:12,355 Stage-1 map = 37%,  reduce = 0%, Cumulative CPU 29.34 sec
2017-11-15 22:51:13,437 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 30.76 sec
2017-11-15 22:51:16,859 Stage-1 map = 53%,  reduce = 0%, Cumulative CPU 47.44 sec
2017-11-15 22:51:18,028 Stage-1 map = 57%,  reduce = 0%, Cumulative CPU 48.46 sec
2017-11-15 22:51:19,115 Stage-1 map = 57%,  reduce = 2%, Cumulative CPU 49.52 sec
2017-11-15 22:51:20,173 Stage-1 map = 63%,  reduce = 10%, Cumulative CPU 55.67 sec
2017-11-15 22:51:21,234 Stage-1 map = 70%,  reduce = 10%, Cumulative CPU 56.51 sec
2017-11-15 22:51:22,268 Stage-1 map = 73%,  reduce = 11%, Cumulative CPU 57.81 sec
2017-11-15 22:51:23,312 Stage-1 map = 80%,  reduce = 16%, Cumulative CPU 63.05 sec
2017-11-15 22:51:24,346 Stage-1 map = 80%,  reduce = 19%, Cumulative CPU 63.27 sec
2017-11-15 22:51:26,464 Stage-1 map = 80%,  reduce = 21%, Cumulative CPU 66.54 sec
2017-11-15 22:51:27,497 Stage-1 map = 80%,  reduce = 23%, Cumulative CPU 66.78 sec
2017-11-15 22:51:28,771 Stage-1 map = 97%,  reduce = 23%, Cumulative CPU 71.03 sec
2017-11-15 22:51:29,827 Stage-1 map = 100%,  reduce = 24%, Cumulative CPU 71.74 sec
2017-11-15 22:51:30,881 Stage-1 map = 100%,  reduce = 74%, Cumulative CPU 80.86 sec
2017-11-15 22:51:31,917 Stage-1 map = 100%,  reduce = 93%, Cumulative CPU 84.32 sec
2017-11-15 22:51:32,947 Stage-1 map = 100%,  reduce = 97%, Cumulative CPU 86.98 sec
2017-11-15 22:51:47,415 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 102.11 sec
2017-11-15 22:52:48,217 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 165.94 sec
MapReduce Total cumulative CPU time: 2 minutes 45 seconds 940 msec
Ended Job = job_1510781966365_0009
Moving data to: /fiveCol1b_hive
MapReduce Jobs Launched:
Stage-Stage-1: Map: 10  Reduce: 10   Cumulative CPU: 188.11 sec   HDFS Read: 2418048520 HDF
S Write: 530309576 SUCCESS
Total MapReduce CPU Time Spent: 3 minutes 8 seconds 110 msec
OK
Time taken: 133.327 seconds
hive>
```

```
[ec2-user@ip-172-31-18-209 ~]$ hadoop fs -cat /fiveCol1b_hive/000000_0 | more
0.5 0.6122448979591837 0.5260732546516992 0.1536894128435018 0.5208397884555875
0.0 0.10204081632653061 0.06626556973704444 0.08021437596795314 0.06646584435001968
1.0 0.20408163265306123 0.11590458250038443 0.09329214415519044 0.11352396097985
0.5 0.9795918367346939 0.48443929724742424 0.20781534253644748 0.47962718351708106
1.0 0.3877551020408163 0.3222974011994464 0.6504270276678085 0.3156775600088527
```

Pig(Solution 2): Note: in step 6, the function "PigStorage" must be input as this format, Can't be edited in upper or lower case.

1.
Parse the followings on the script and then execute in the PIG directory

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

2.
```
describe lod
```

3.
```
lodDis = filter lod by ((lo_discount>=4.0) and (lo_discount<=6.0));
```

4.
```
fiveCol = foreach lodDis generate lo_discount, lo_quantity, lo_extendedprice, lo_ordertotalprice,
lo_revenue;
```

5
```
dump fiveCol
```

6.
```
store fiveCol into '/fiveCol1b_pig' using PigStorage (' ');
```

7.
```
Hadoop fs -copyToLocal '/fiveCol1b_pig'
```

8.
```
cat part-m-00000 part-m-00000 part-m-00001 part-m-00002 part-m-00003 part-m-00004 part-m-00005
part-m-00006 part-m-00007 part-m-00008 part-m-00009 part-m-00010 part-m-00011 part-m-00012
part-m-00013 part-m-00014 part-m-00015 part-m-00016 part-m-00017 >> fiveCol1bpig_one
```

9. (Prepare for Question 3)
```
hadoop fs -put fiveCol1bpig_one
```

Pig output:

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.4   0.15.0  ec2-user        2017-11-15 22:56:25     2017-11-15 22:57:23     FILTER

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   Max
ReduceTime      MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Out
puts
job_1510781966365_0010  18      0       45      10      39      43      0       0       0 f
iveCol,lod,lodDis       MAP_ONLY        /fiveCol1b_pig,

Input(s):
Successfully read 23996604 records (2417832945 bytes) from: "/user/ec2-user/lineorder.tbl"

Output(s):
Successfully stored 6543471 records (249054913 bytes) in: "/fiveCol1b_pig"

Counters:
Total records written : 6543471
Total bytes written : 249054913
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
```

```
2017-11-15 22:57:23,035 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to
ResourceManager at /172.31.18.209:8032
2017-11-15 22:57:23,039 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Appli
cation state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history ser
ver
2017-11-15 22:57:23,130 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to
ResourceManager at /172.31.18.209:8032
2017-11-15 22:57:23,134 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Appli
cation state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history ser
ver
2017-11-15 22:57:23,166 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to
ResourceManager at /172.31.18.209:8032
2017-11-15 22:57:23,170 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Appli
cation state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history ser
ver
2017-11-15 22:57:23,221 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduc
eLayer.MapReduceLauncher - Success!
grunt>
```

```
[ec2-user@ip-172-31-18-209 ~]$ hadoop fs -cat /fiveCol1b_pig/part-m-00000 | more
5.0 42 4889724.0 4923951.0 4645237.0
4.0 28 5140184.0 2.5095892E7 4934576.0
6.0 28 4644248.0 2.5095892E7 4365593.0
5.0 20 2196340.0 2.5095892E7 2086523.0
5.0 4 665440.0 800181.0 632168.0
```

C. Create a pre-join (i.e. a new data file) that corresponds to the following query below. You can think of it as a materialized view. What is the size of the new file? Use Hive and Pig (2 different solutions).

```
SELECT lo_partkey, lo_suppkey, lo_discount, d_year, lo_revenue
FROM lineorder, dwdate
WHERE lo_orderdate = d_datekey;
```

Ans:

Hive (Solution 1):

```
wget http://rasinsrv07.cstcis.cti.depaul.edu/CSC553/data/dwdate.tbl
1.
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 '|';
2.
LOAD DATA LOCAL INPATH '/home/ec2-user/dwdate.tbl' OVERWRITE INTO TABLE dwdate;
3.
INSERT OVERWRITE DIRECTORY 'preJoin_hive_1c'
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;
```

Hive Output:

```
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-lo
g4j12-1.7.5.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_20171114194539_b5650029-3292-4222-a722-449cf33b40df.l
og
2017-11-14 19:45:46     Starting to launch local task to process map join;      maximum memory
 = 477102080
2017-11-14 19:45:47     Dump the side-table for tag: 1 with group count: 2556 into file: file:
/tmp/ec2-user/04360607-6bb3-4bdb-9773-c4cd4a0b74c5/hive_2017-11-14_19-45-39_951_73728203051132
8078-1/-local-10002/HashTable-Stage-3/MapJoin-mapfile01--.hashtable
2017-11-14 19:45:47     Uploaded 1 File to: file:/tmp/ec2-user/04360607-6bb3-4bdb-9773-c4cd4a0
b74c5/hive_2017-11-14_19-45-39_951_73728203051132 8078-1/-local-10002/HashTable-Stage-3/MapJoin
-mapfile01--.hashtable (67039 bytes)
2017-11-14 19:45:47     End of local task; Time Taken: 1.204 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1510686994714_0005, Tracking URL = http://ip-172-31-18-209.us-west-1.comput
e.internal:8088/proxy/application_1510686994714_0005/
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job  -kill job_1510686994714_0005
Hadoop job information for Stage-3: number of mappers: 10; number of reducers: 0
2017-11-14 19:45:53,599 Stage-3 map = 0%,  reduce = 0%
2017-11-14 19:46:06,888 Stage-3 map = 10%,  reduce = 0%, Cumulative CPU 25.79 sec
2017-11-14 19:46:09,115 Stage-3 map = 30%,  reduce = 0%, Cumulative CPU 54.92 sec
2017-11-14 19:46:11,343 Stage-3 map = 55%,  reduce = 0%, Cumulative CPU 70.61 sec
2017-11-14 19:46:13,468 Stage-3 map = 60%,  reduce = 0%, Cumulative CPU 73.17 sec
2017-11-14 19:46:14,525 Stage-3 map = 80%,  reduce = 0%, Cumulative CPU 84.54 sec
2017-11-14 19:46:15,576 Stage-3 map = 85%,  reduce = 0%, Cumulative CPU 85.22 sec
2017-11-14 19:46:19,774 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 95.69 sec
MapReduce Total cumulative CPU time: 1 minutes 35 seconds 690 msec
Ended Job = job_1510686994714_0005
Moving data to: preJoin_hive_1c
MapReduce Jobs Launched:
Stage-Stage-3: Map: 10   Cumulative CPU: 95.69 sec   HDFS Read: 2417935760 HDFS Write: 6574552
11 SUCCESS
Total MapReduce CPU Time Spent: 1 minutes 35 seconds 690 msec
OK
Time taken: 40.982 seconds
hive>
```

```
[ec2-user@ip-172-31-18-209 ~]$ hadoop fs -cat preJoin_hive_1c//000000_0 | more
310379,16546,4,1996,2267435
134619,3259,9,1996,5417226
127400,1418,10,1996,1027728
4263,18842,9,1996,2974178
48054,32491,10,1996,2164428
31269,27344,7,1996,3571973
212340,21314,0,1996,4758854
8594,39169,6,1993,6355955
38071,33331,10,1993,4449998
256897,28180,6,1993,4705147
58760,14834,1,1993,340314
366189,32134,4,1993,3373896
```

```
[ec2-user@ip-172-31-18-209 ~]$ hadoop fs -ls preJoin_hive_1c
Found 10 items
-rwxr-xr-x   3 ec2-user supergroup   73693378 2017-11-14 19:46 preJoin_hive_1c/000000_0
-rwxr-xr-x   3 ec2-user supergroup   73189463 2017-11-14 19:46 preJoin_hive_1c/000001_0
-rwxr-xr-x   3 ec2-user supergroup   73390828 2017-11-14 19:46 preJoin_hive_1c/000002_0
-rwxr-xr-x   3 ec2-user supergroup   73379423 2017-11-14 19:46 preJoin_hive_1c/000003_0
-rwxr-xr-x   3 ec2-user supergroup   72663574 2017-11-14 19:46 preJoin_hive_1c/000004_0
-rwxr-xr-x   3 ec2-user supergroup   72658783 2017-11-14 19:46 preJoin_hive_1c/000005_0
-rwxr-xr-x   3 ec2-user supergroup   72663767 2017-11-14 19:46 preJoin_hive_1c/000006_0
-rwxr-xr-x   3 ec2-user supergroup   72660094 2017-11-14 19:46 preJoin_hive_1c/000007_0
-rwxr-xr-x   3 ec2-user supergroup   72658860 2017-11-14 19:46 preJoin_hive_1c/000008_0
-rwxr-xr-x   3 ec2-user supergroup     497041 2017-11-14 19:46 preJoin_hive_1c/000009_0
[ec2-user@ip-172-31-18-209 ~]$ hadoop fs -du -s -h  preJoin_hive_1c
627.0 M  preJoin_hive_1c
[ec2-user@ip-172-31-18-209 ~]$
```

The size of file(s) is 627M

Pig (Solution 2):

1.

```
Hadoop fs -put dwdate.tbl;
Hadoop fs -put lineorder.tbl;
```

2.

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

3.

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

4.

```
joinDate = JOIN lod BY lo_orderdate, dwd BY d_datekey;
dump joinDate;
```

5.

```
result = foreach joinDate generate lo_partkey , lo_suppkey , lo_discount , d_year , lo_revenue;
```

6.
dump result;
7.
store result into 'preJoin_pig_1c' using PigStorage (',');

Pig Output

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.4   0.15.0  ec2-user        2017-11-14 21:58:59     2017-11-14 22:06:36     HASH_JOIN

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduce
Time    MedianReducetime        Alias   Feature Outputs
job_1510686994714_0040  19      3       64      21      45      44      218     217     218     218     dwd,joinDate,lod,result H
ASH_JOIN        hdfs://172.31.18.209/user/ec2-user/preJoin_pig_1c,

Input(s):
Successfully read 2556 records from: "/user/ec2-user/dwdate.tbl"
Successfully read 23996604 records from: "/user/ec2-user/lineorder.tbl"

Output(s):
Successfully stored 56900543 records (2127956273 bytes) in: "hdfs://172.31.18.209/user/ec2-user/preJoin_pig_1c"

Counters:
Total records written : 56900543
Total bytes written : 2127956273
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1510686994714_0040


2017-11-14 22:06:36,367 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /172.31.18.209:8032
2017-11-14 22:06:36,372 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplica
tionStatus=SUCCEEDED. Redirecting to job history server
2017-11-14 22:06:36,460 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /172.31.18.209:8032
2017-11-14 22:06:36,464 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplica
tionStatus=SUCCEEDED. Redirecting to job history server
2017-11-14 22:06:36,543 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /172.31.18.209:8032
2017-11-14 22:06:36,548 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplica
tionStatus=SUCCEEDED. Redirecting to job history server
2017-11-14 22:06:36,579 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
grunt>
```

```
[ec2-user@ip-172-31-18-209 ~]$ hadoop fs -cat preJoin_pig_1c/part-r-00000 | more
167160.0,16611.0,2.0,1992.0,4690205.0
308001.0,4814.0,6.0,1992.0,1327830.0
68731.0,9153.0,2.0,1992.0,5496926.0
178566.0,24685.0,8.0,1992.0,151299.0
159153.0,3229.0,9.0,1992.0,4522531.0
78958.0,21553.0,3.0,1992.0,3569798.0
192146.0,25703.0,7.0,1992.0,2302940.0
359457.0,2220.0,6.0,1992.0,997817.0
302809.0,37301.0,6.0,1992.0,3406165.0
15784.0,8935.0,6.0,1992.0,6071614.0
```

```
[ec2-user@ip-172-31-18-209 ~]$ hadoop fs -du -s -h preJoin_pig_1c
2.0 G  preJoin_pig_1c
[ec2-user@ip-172-31-18-209 ~]$
```

file size is 2.0G

Note
Based on the above reslt, the PIG output files are larger than the HIVE files. I would think that the reason is PIG Tables are created with 'float' numeric values , but HIVE table was created with 'Integer' Values.

Part 2: Querying

All queries from SSBM benchmark are available here:
http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/SSBM1/SSBM_queries_all.sql
Using Scale4 data perform the following data processing and don't forget to time your results.

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

A. Run SSBM queries 2.2, 3.2 and 4.2 using Hive only.

Ans:

2.2

```
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_brand1 between 'MFGR#2221'
  and 'MFGR#2228'
  and s_region = 'ASIA'
group by d_year, p_brand1
order by d_year, p_brand1;
```

Output: (4-nodes)

```
2740118896      1996    MFGR#2226
2520128511      1996    MFGR#2227
2958335540      1996    MFGR#2228
2538388145      1997    MFGR#2221
2456017205      1997    MFGR#2222
2731016064      1997    MFGR#2223
2290329277      1997    MFGR#2224
2478673421      1997    MFGR#2225
2760849777      1997    MFGR#2226
2801322559      1997    MFGR#2227
2773104030      1997    MFGR#2228
1640853482      1998    MFGR#2221
1464532062      1998    MFGR#2222
1388416942      1998    MFGR#2223
1513940667      1998    MFGR#2224
1554728340      1998    MFGR#2225
1614687724      1998    MFGR#2226
1530903188      1998    MFGR#2227
1791190203      1998    MFGR#2228
Time taken: 164.003 seconds, Fetched: 56 row(s)
hive>
```

Time taken: 164.003 sec with 56 rows result.

3.2

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

Output: (4-nodes)

```
UNITED ST5      UNITED ST5      1997    245515678
UNITED ST7      UNITED ST6      1997    245708166
UNITED ST1      UNITED ST6      1997    246102843
UNITED ST9      UNITED ST5      1997    247390190
UNITED ST1      UNITED ST5      1997    255552893
UNITED ST6      UNITED ST4      1997    256519276
UNITED ST8      UNITED ST0      1997    257634026
UNITED ST6      UNITED ST6      1997    257725690
UNITED ST3      UNITED ST7      1997    257938073
UNITED ST0      UNITED ST5      1997    260361000
UNITED ST6      UNITED ST0      1997    262613049
UNITED ST5      UNITED ST3      1997    266054384
UNITED ST4      UNITED ST5      1997    273729195
UNITED ST5      UNITED ST1      1997    274973084
UNITED ST9      UNITED ST0      1997    275396675
UNITED ST4      UNITED ST6      1997    278555397
UNITED ST5      UNITED ST4      1997    294193681
UNITED ST5      UNITED ST7      1997    296752211
UNITED ST5      UNITED ST0      1997    316035257
Time taken: 84.701 seconds, Fetched: 600 row(s)
hive>
```

Time taken is 84.701 sec with 600 rows result.

4.2

```
--Q4.2  Removed second match of OR conditions, expression in sum
select d_year, s_nation, p_category, sum(lo_revenue) as profit1
from lineorder , customer , supplier , part, dwdate
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_region = 'AMERICA'
  and d_year = 1997
  and p_mfgr = 'MFGR#1'
group by d_year, s_nation, p_category;
```

Ans:

```
Starting Job = job_1511043867404_0004, Tracking URL = http://ip-172-31-30-37.us-west-1.compute.internal:8088/proxy/
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job  -kill job_1511043867404_0004
Hadoop job information for Stage-5: number of mappers: 1; number of reducers: 1
2017-11-18 22:45:05,842 Stage-5 map = 0%,  reduce = 0%
2017-11-18 22:45:12,028 Stage-5 map = 100%,  reduce = 0%, Cumulative CPU 2.79 sec
2017-11-18 22:45:17,193 Stage-5 map = 100%,  reduce = 100%, Cumulative CPU 3.99 sec
MapReduce Total cumulative CPU time: 3 seconds 990 msec
Ended Job = job_1511043867404_0004
MapReduce Jobs Launched:
Stage-Stage-15: Map: 10   Cumulative CPU: 78.63 sec   HDFS Read: 2417923521 HDFS Write: 159158860 SUCCESS
Stage-Stage-14: Map: 3   Cumulative CPU: 19.97 sec   HDFS Read: 159177226 HDFS Write: 37276493 SUCCESS
Stage-Stage-3: Map: 3  Reduce: 1   Cumulative CPU: 21.58 sec   HDFS Read: 88342019 HDFS Write: 8266710 SUCCESS
Stage-Stage-5: Map: 1  Reduce: 1   Cumulative CPU: 3.99 sec   HDFS Read: 8282153 HDFS Write: 815 SUCCESS
Total MapReduce CPU Time Spent: 2 minutes 4 seconds 170 msec
OK
1997    ARGENTINA       MFGR#11 4234103096
1997    ARGENTINA       MFGR#12 4489420815
1997    ARGENTINA       MFGR#13 4445657741
1997    ARGENTINA       MFGR#14 4165351679
1997    ARGENTINA       MFGR#15 4574444921
1997    BRAZIL  MFGR#11 4057698290
1997    BRAZIL  MFGR#12 4124645435
1997    BRAZIL  MFGR#13 4048948816
1997    BRAZIL  MFGR#14 4080535388
1997    BRAZIL  MFGR#15 4265134120
1997    CANADA  MFGR#11 4365839263
1997    CANADA  MFGR#12 4221266344
1997    CANADA  MFGR#13 4079946458
1997    CANADA  MFGR#14 4255004665
1997    CANADA  MFGR#15 4545035148
1997    PERU    MFGR#11 4377522939
1997    PERU    MFGR#12 4358488411
1997    PERU    MFGR#13 4079223279
1997    PERU    MFGR#14 4383949788
1997    PERU    MFGR#15 4229817375
1997    UNITED STATES   MFGR#11 4182140089
1997    UNITED STATES   MFGR#12 4224160699
1997    UNITED STATES   MFGR#13 4391310016
1997    UNITED STATES   MFGR#14 4180428267
1997    UNITED STATES   MFGR#15 3975901975
Time taken: 96.094 seconds, Fetched: 25 row(s)
hive>
```

In this particular question, I have experienced that the sequence of calling the table in the query is really matter affecting the speed of the process. The sequence of calling the tables should be in the SAME order according to the conditions (“where” statement) statements.

B. For this part use Hive and Pig (two different solutions) to run Q2.1 using what you have created in 1-C (i.e. use PreJoin1 instead of lineorder and dwdate tables in the from clause). You would need to rewrite the query accordingly. (e.g. something like,

```
select sum(lo_revenue), d_year, p_brand1
from MyNewStructureFrom1C, part, supplier
where 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: 4 –Nodes

1. From part 1c:

```
INSERT OVERWRITE DIRECTORY 'preJoin_hive_1c'
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;
```

2.

```
Create Table preJoin1c(
lo_partkey int,
lo_suppkey int,
lo_discount int,
d_year int,
lo_revenue int
) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
```

2. Copy the hdfs file to the local file system

```
hadoop fs -copyToLocal preJoin_hive_1c/000000_0 > preJoin_hive_0
hadoop fs -copyToLocal preJoin_hive_1c/000001_0 > preJoin_hive_1
hadoop fs -copyToLocal preJoin_hive_1c/000002_0 > preJoin_hive_2
hadoop fs -copyToLocal preJoin_hive_1c/000003_0 > preJoin_hive_3
hadoop fs -copyToLocal preJoin_hive_1c/000004_0 > preJoin_hive_4
hadoop fs -copyToLocal preJoin_hive_1c/000005_0 > preJoin_hive_5
hadoop fs -copyToLocal preJoin_hive_1c/000006_0 > preJoin_hive_6
hadoop fs -copyToLocal preJoin_hive_1c/000007_0 > preJoin_hive_7
hadoop fs -copyToLocal preJoin_hive_1c/000008_0 > preJoin_hive_8
hadoop fs -copyToLocal preJoin_hive_1c/000009_0 > preJoin_hive_9
```

3. Copy the file content to the local file

```
cp 000000_0 preJoin_hive_0
cp 000000_1 preJoin_hive_1
cp 000000_2 preJoin_hive_2
```

```
cp 000000_3 preJoin_hive_3
cp 000000_4 preJoin_hive_4
cp 000000_5 preJoin_hive_5
cp 000000_6 preJoin_hive_6
cp 000000_7 preJoin_hive_7
cp 000000_8 preJoin_hive_8
cp 000000_9 preJoin_hive_9
```

5. Insert the local data to the preJoin1c table

```
LOAD DATA LOCAL INPATH '/home/ec2-user/preJoin_hive_0' INTO TABLE preJoin1c;
LOAD DATA LOCAL INPATH '/home/ec2-user/preJoin_hive_1' INTO TABLE preJoin1c;
LOAD DATA LOCAL INPATH '/home/ec2-user/preJoin_hive_2' INTO TABLE preJoin1c;
LOAD DATA LOCAL INPATH '/home/ec2-user/preJoin_hive_3' INTO TABLE preJoin1c;
LOAD DATA LOCAL INPATH '/home/ec2-user/preJoin_hive_4' INTO TABLE preJoin1c;
LOAD DATA LOCAL INPATH '/home/ec2-user/preJoin_hive_5' INTO TABLE preJoin1c;
LOAD DATA LOCAL INPATH '/home/ec2-user/preJoin_hive_6' INTO TABLE preJoin1c;
LOAD DATA LOCAL INPATH '/home/ec2-user/preJoin_hive_7' INTO TABLE preJoin1c;
LOAD DATA LOCAL INPATH '/home/ec2-user/preJoin_hive_8' INTO TABLE preJoin1c;
LOAD DATA LOCAL INPATH '/home/ec2-user/preJoin_hive_9' INTO TABLE preJoin1c;
```

6.

```
select sum(lo_revenue), d_year, p_brand1
from preJoin1c, part, supplier
where 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 Output: (4-node)

```
1640583696      1998    MFGR#1226
1565657860      1998    MFGR#1227
1607890751      1998    MFGR#1228
1350601347      1998    MFGR#1229
1470503353      1998    MFGR#123
1441898473      1998    MFGR#1230
1445039464      1998    MFGR#1231
1710140678      1998    MFGR#1232
1538979218      1998    MFGR#1233
1532309319      1998    MFGR#1234
1598713364      1998    MFGR#1235
1577658136      1998    MFGR#1236
1532687418      1998    MFGR#1237
1285428693      1998    MFGR#1238
1459545128      1998    MFGR#1239
1525737275      1998    MFGR#124
1587370161      1998    MFGR#1240
1477715730      1998    MFGR#125
1466946762      1998    MFGR#126
1686460729      1998    MFGR#127
1538644707      1998    MFGR#128
1207004714      1998    MFGR#129
Time taken: 128.035 seconds, Fetched: 280 row(s)
hive>
```

PIG: At least 4 –Nodes

1

```
hadoop fs -put part.tbl;
hadoop fs -put supplier.tbl;
```

2

```
part= LOAD '/user/ec2-user/part.tbl' USING PigStorage('|')
AS (p_partkey     :float,
p_name        :chararray,
p_mfgr        :chararray,
p_category    :chararray,
p_brand1      :chararray,
p_color       :chararray,
p_type        :chararray,
p_size        :float,
p_container   :chararray);
```

3

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

4.

```
preJoin = LOAD 'preJoin_pig_1c' USING PigStorage(',')
AS (lo_partkey :float
, lo_suppkey :float
, lo_discount :float
, d_year :chararray
 , lo_revenue :float);
```

5.

```
pre_supplier_join= JOIN preJoin BY lo_suppkey , supplier BY s_suppkey;
```

6.

```
pre_supplier_part_join = JOIN pre_supplier_join BY lo_partkey, part BY p_partkey;
```

7.

```
pre_supplier_part_join_filter = FILTER pre_supplier_part_join BY ( p_category == 'MFGR#12' )
and ( s_region == 'AMERICA');
```

8.

```
newtable = FOREACH pre_supplier_part_join_filter generate lo_revenue, d_year, p_brand1;
```

9.

```
STORE newtable INTO 'new_dataJoin2' using PigStorage(',');
```

First Phase:

```
2017-11-16 08:14:21,297 [main] INFO  org.apache.pig.tools.pigstats.mapreduce.SimplePigStats - Script Statis
tics:

HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.4   0.15.0  ec2-user        2017-11-16 08:10:26     2017-11-16 08:14:21     HASH_JOIN,FILTER

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   Min
ReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_1510819401388_0001  17      3       101     9       78      90      133     130     132     132     pre
Join,pre_supplier_join,supplier      HASH_JOIN
job_1510819401388_0002  4       1       39      14      32      38      40      40      40      40      new
table,part,pre_supplier_part_join,pre_supplier_part_join_filter      HASH_JOIN      hdfs://172.31.18.209/us
er/ec2-user/new_dataJoin2,

Input(s):
Successfully read 56900543 records from: "hdfs://172.31.18.209/user/ec2-user/preJoin_pig_1c"
Successfully read 40000 records from: "/user/ec2-user/supplier.tbl"
Successfully read 600000 records from: "/user/ec2-user/part.tbl"

Output(s):
Successfully stored 451185 records (12017904 bytes) in: "hdfs://172.31.18.209/user/ec2-user/new_dataJoin2"
```

Time taken for phase 1 is about 03min 55sec

```
[ec2-user@ip-172-31-18-209 ~]$ hadoop fs -cat new_dataJoin2/part-r-00000 | more
3371060.0,1993.0,MFGR#1224
279083.0,1998.0,MFGR#1224
279083.0,1998.0,MFGR#1224
279083.0,1998.0,MFGR#1224
176465.0,1992.0,MFGR#1224
176465.0,1992.0,MFGR#1224
352930.0,1994.0,MFGR#1224
352930.0,1994.0,MFGR#1224
176465.0,1992.0,MFGR#1224
```

10.

```
newjoin2 = LOAD 'new_dataJoin2' USING PigStorage (',')
 AS (lo_revenue :float,
 d_year :chararray ,
 p_brand1  :chararray);
```

11.

```
join_group = GROUP newjoin2 BY (d_year, p_brand1);
```

12.

```
result = FOREACH join_group GENERATE group, SUM(newjoin2.lo_revenue);
```

13.

```
result2 = ORDER result BY group;
```

14.

```
STORE result2 into 'final_pig_2b' using PigStorage(',');
```

Second Phase

```
2017-11-16 08:21:52,979 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 100% complete
2017-11-16 08:21:53,016 [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        2017-11-16 08:20:49     2017-11-16 08:21:53     GROUP_BY,ORDER_BY

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTim
e   MedianReducetime        Alias   Feature Outputs
job_1510819401388_0003  1       1       6       6       6       6       3       3       3       3       join_group,newjoin2,result
      GROUP_BY,COMBINER
job_1510819401388_0004  1       1       2       2       2       2       2       2       2       2       result2 SAMPLER
job_1510819401388_0005  1       1       3       3       3       3       3       3       3       3       result2 ORDER_BY        hdfs
://172.31.18.209/user/ec2-user/final_pig_2b,

Input(s):
Successfully read 451185 records (12018292 bytes) from: "hdfs://172.31.18.209/user/ec2-user/new_dataJoin2"

Output(s):
Successfully stored 280 records (9145 bytes) in: "hdfs://172.31.18.209/user/ec2-user/final_pig_2b"

Counters:
Total records written : 280
Total bytes written : 9145
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
```

Time taken for 2nd phase is about 01min 04sec

```
[ec2-user@ip-172-31-18-209 ~]$ hadoop fs -cat final_pig_2b/part-r-00000 | more
(1992.0,MFGR#121),6.163246747E9
(1992.0,MFGR#1210),6.113279666E9
(1992.0,MFGR#1211),6.274516322E9
(1992.0,MFGR#1212),6.444266136E9
(1992.0,MFGR#1213),6.024458571E9
(1992.0,MFGR#1214),6.121549826E9
(1992.0,MFGR#1215),6.39141868E9
(1992.0,MFGR#1216),6.053234748E9
(1992.0,MFGR#1217),6.681688577E9
(1992.0,MFGR#1218),5.849328282E9
(1992.0,MFGR#1219),5.764990555E9
```

The total time taken is about 04 min 59 sec

Part 3: Clustering

Using the file you have created in 1-B, run KMeans clustering using 7 clusters.

A. Using Mahout synthetic clustering as you have in a previous assignment on sample data.

Command line input: the input file 'fiveCol1bhive_one' was nomarlized into (0,1) sclae in part 2b

time mahout org.apache.mahout.clustering.syntheticcontrol.kmeans.Job --maxIter 10 --numClusters 7 --t1 0.3 --t2 0.5 --input fiveCol1bhive_one --output cluster_output

Output:

```
        at java.io.DataInputStream.readUTF(DataInputStream.java:564)
        at org.apache.mahout.clustering.classify.WeightedPropertyVectorWritable.readFields(
WeightedPropertyVectorWritable.java:61)
        at org.apache.hadoop.io.SequenceFile$Reader.getCurrentValue(SequenceFile.java:2254)
        at org.apache.hadoop.io.SequenceFile$Reader.next(SequenceFile.java:2382)
        at org.apache.mahout.common.iterator.sequencefile.SequenceFileIterator.computeNext(
SequenceFileIterator.java:101)
        at org.apache.mahout.common.iterator.sequencefile.SequenceFileIterator.computeNext(
SequenceFileIterator.java:40)
        at com.google.common.collect.AbstractIterator.tryToComputeNext(AbstractIterator.jav
a:143)
        at com.google.common.collect.AbstractIterator.hasNext(AbstractIterator.java:138)
        at com.google.common.collect.Iterators$5.hasNext(Iterators.java:543)
        at com.google.common.collect.ForwardingIterator.hasNext(ForwardingIterator.java:43)
        at org.apache.mahout.utils.clustering.ClusterDumper.readPoints(ClusterDumper.java:3
11)
        at org.apache.mahout.utils.clustering.ClusterDumper.init(ClusterDumper.java:262)
        at org.apache.mahout.utils.clustering.ClusterDumper.<init>(ClusterDumper.java:92)
        at org.apache.mahout.clustering.syntheticcontrol.kmeans.Job.run(Job.java:141)
        at org.apache.mahout.clustering.syntheticcontrol.kmeans.Job.run(Job.java:95)
        at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:70)
        at org.apache.mahout.clustering.syntheticcontrol.kmeans.Job.main(Job.java:54)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.jav
a:43)
        at java.lang.reflect.Method.invoke(Method.java:606)
        at org.apache.hadoop.util.ProgramDriver$ProgramDescription.invoke(ProgramDriver.jav
a:71)
        at org.apache.hadoop.util.ProgramDriver.run(ProgramDriver.java:144)
        at org.apache.hadoop.util.ProgramDriver.driver(ProgramDriver.java:152)
        at org.apache.mahout.driver.MahoutDriver.main(MahoutDriver.java:195)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.jav
a:43)
        at java.lang.reflect.Method.invoke(Method.java:606)

real    2m33.625s
user    1m20.692s
sys     0m2.432s
[ec2-user@ip-172-31-18-209 pig-0.15.0]$
```

B. Using Hadoop streaming perform one iteration manually with randomly chosen input centers. (This would require passing a text file with cluster centers using -file option, opening the centers.txt in the mapper with open('centers.txt', 'r') and assigning a key to each point based on which center is the closest to each particular point). Your reducer would need to compute the new centers, and at that point the iteration is done.

Command line

```
time hadoop jar /home/ec2-user/hadoop-2.6.4/share/hadoop/tools/lib/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 fiveCol1bhive_one -output 3b_kmean_output -mapper
3bmapper.py -reducer 3breducer.py -file 3breducer.py -file 3bmapper.py -file 3bcenters.txt
```

Mapper Code:

```
import math
import sys
center_lst=[]
center_dict={}

#Manually assign the first 7 clusters center
with open('3bcenters.txt','r') as ofile:
    lines=ofile.readlines()
    for line in lines:
        words=line.strip().split(' ')
        center_lst.append(map(float,words))

#Read from the stdin
for record in sys.stdin:
    instance=record.strip().split(' ')
    cal_lst=[]
    for c in center_lst:
        num=0
        for i in range(0,len(c)):
            #print(c[i])
            a=float(c[i])
            b=float(instance[i])
            ab=a-b
            #Calculate the Euclidean Distance
            num+= math.pow(ab,2)
            #print(num)
        cal_lst.append(round(math.sqrt(num),3))
    #Assign the key (Index of cluster,0-6) to the instance based on the shortest distance
    a=cal_lst.index(min(cal_lst))
    if a in center_dict.keys():
        center_dict[a].append(instance)
    else:
        center_dict[a]=[instance]

#Print out ALL the instance with key
for i in range(len(center_lst)):
    if i in center_dict.keys():
        for val in center_dict[i]:
            print '%d% 0.5f% 0.5f% 0.5f% 0.5f% 0.5f'%
(i,float(val[0]),float(val[1]),float(val[2]),float(val[3]),float(val[4]))
    else:
        print '%d% 0.5f% 0.5f% 0.5f% 0.5f%
0.5f'%(i,center_lst[i][0],center_lst[i][1],center_lst[i][2],center_lst[i][3],center_lst[i][4]
)
```

Reducer Code:

```
import math
import sys

#Initialize variables
new_center=[]
c=[0,0,0,0,0]
key=-1
counter=0.0
for line in sys.stdin:
    words = line.strip().split(' ')
    if key == words[0]: #Collect all the points belong to the same center
        for i in range(1,len(words)):
            c[i-1] += float(words[i]) #Add all the columns values among the SAME
cluster members
        counter += 1.0
    else: # Start here when countering the new clusters instance
        if counter != 0.0:
            aa=[0.0,0.0,0.0,0.0,0.0]
            aa = [c[i]/counter for i in range(len(c))]
            new_center.append(aa) #Store all the clusters centers in the list
        c=[0,0,0,0,0] #reinitialize
        counter=0
        key=words[0]
        for i in range(1,len(words)):
            c[i-1] += float(words[i]) #Add all the columns values among the SAME
cluster members
        counter+=1.0

aa = [c[i]/counter for i in range(len(c))]
new_center.append(aa)

#The first column is the number of cluster
#The rest of columns are the centers
for i,val in enumerate(new_center):
    print '%d% 0.5f% 0.5f% 0.5f% 0.5f% 0.5f'%
(i,float(val[0]),float(val[1]),float(val[2]),float(val[3]),float(val[4]))
```

Kmean Cluster output

```
17/11/16 06:48:33 INFO mapreduce.Job:  map 100% reduce 95%
17/11/16 06:48:36 INFO mapreduce.Job:  map 100% reduce 100%
17/11/16 06:48:37 INFO mapreduce.Job: Job job_1510808986074_0004 completed successful
17/11/16 06:48:37 INFO mapreduce.Job: Counters: 50
        File System Counters
                FILE: Number of bytes read=588912822
                FILE: Number of bytes written=883922486
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=530322280
                HDFS: Number of bytes written=301
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=4
                Launched reduce tasks=1
                Data-local map tasks=3
                Rack-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=418928
                Total time spent by all reduces in occupied slots (ms)=27689
                Total time spent by all map tasks (ms)=418928
                Total time spent by all reduce tasks (ms)=27689
                Total vcore-milliseconds taken by all map tasks=418928
                Total vcore-milliseconds taken by all reduce tasks=27689
                Total megabyte-milliseconds taken by all map tasks=428982272
                Total megabyte-milliseconds taken by all reduce tasks=28353536
        Map-Reduce Framework
                Map input records=6543471
                Map output records=6543475
                Map output bytes=281369425
                Map output materialized bytes=294456399
                Input split bytes=416
                Combine input records=0
                Combine output records=0
                Reduce input groups=6543467
                Reduce shuffle bytes=294456399
                Reduce input records=6543475
                Reduce output records=7
                Spilled Records=19630425
                Shuffled Maps =4
                Failed Shuffles=0
                Merged Map outputs=4
                GC time elapsed (ms)=763
                CPU time spent (ms)=424700
                Physical memory (bytes) snapshot=1221033984
                Virtual memory (bytes) snapshot=4978413568
                Total committed heap usage (bytes)=801112064
```

Kmean cluster output

```
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=530321864
        File Output Format Counters
                Bytes Written=301
17/11/16 06:48:37 INFO streaming.StreamJob: Output directory: 3b_kmean_output

real    2m22.915s
user    0m4.132s
sys     0m0.208s
[ec2-user@ip-172-31-18-209 ~]$
```

Total time taken 2min 22.915sec

The New cluster center, the first column is the index of the 7 clusters

```
[ec2-user@ip-172-31-18-209 ~]$ hadoop fs -cat 3b_kmean_output/part-00000 |more
0 0.20301 0.42503 0.29720 0.34499 0.29642
1 0.76325 0.86952 0.66319 0.44229 0.65320
2 0.71433 0.66696 0.46349 0.20237 0.45715
3 1.00000 0.18940 0.13855 0.26672 0.13570
4 0.39905 0.45043 0.38753 0.80936 0.38493
5 0.20000 0.10000 0.50000 0.60000 0.65000
6 1.00000 0.44442 0.31565 0.48748 0.30916
[ec2-user@ip-172-31-18-209 ~]$
```

The initial assigned cluster center (Manually assigned in Center.txt). The dataset was normalized in (0,1) scale in Part 1b

```
[ec2-user@ip-172-31-18-209 ~]$ hadoop fs -cat 3bcenters.txt | more
0.1 0.32 0.6 0.9 0.14
0.61 0.82 0.9 0.58 0.3
0.5 0.6 0.25 0.1 0.7
0.84 0.1 0.1 0.4 0.8
0.3 0.4 0.2 0.8 0.6
0.2 0.1 0.5 0.6 0.65
0.8 0.3 0.2 0.8 0.7
[ec2-user@ip-172-31-18-209 ~]$
```

NOTE: if you get a java.lang.OutOfMemoryError error, you will need to reconfigure Hadoop to supply the java virtual machine with more memory. You can do this by editing the mapred-site.xml (Mapper should not need much RAM):

```
<property>
  <name> mapreduce.reduce.java.opts</name>
  <value>-Xmx1024m</value>
</property>
```

The amount of memory can be tweaked (you can go higher, but keep in mind how much physical memory your machine has). Do not forget to restart Hadoop after any configuration file change. If you **still** run out of memory in 3-A submit the screenshot of that and you will get full credit for the question.

Part 4: Performance

Compare the performance given following combinations. If you already ran that combination before it is sufficient to copy the runtime for comparison.

A. All three of your solutions to Part-1A with

a. Scale4: single node and a cluster of at least 4 nodes

Ans: Hive – 4- node

```
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1510651862836_0003, Tracking URL = http://ip-172-31-18-209.us-wes
t-1.compute.internal:8088/proxy/application_1510651862836_0003/
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job  -kill job_1510651862836_0
003
Hadoop job information for Stage-1: number of mappers: 10; number of reducers: 0
2017-11-14 10:02:55,267 Stage-1 map = 0%,  reduce = 0%
2017-11-14 10:03:09,652 Stage-1 map = 10%,  reduce = 0%, Cumulative CPU 38.99 sec
2017-11-14 10:03:26,254 Stage-1 map = 15%,  reduce = 0%, Cumulative CPU 161.45 sec
2017-11-14 10:03:29,538 Stage-1 map = 20%,  reduce = 0%, Cumulative CPU 181.37 sec
2017-11-14 10:03:30,867 Stage-1 map = 25%,  reduce = 0%, Cumulative CPU 188.73 sec
2017-11-14 10:03:34,157 Stage-1 map = 30%,  reduce = 0%, Cumulative CPU 208.24 sec
2017-11-14 10:03:38,598 Stage-1 map = 35%,  reduce = 0%, Cumulative CPU 239.07 sec
2017-11-14 10:03:40,772 Stage-1 map = 40%,  reduce = 0%, Cumulative CPU 257.9 sec
2017-11-14 10:03:46,297 Stage-1 map = 45%,  reduce = 0%, Cumulative CPU 284.21 sec
2017-11-14 10:03:47,387 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 295.95 sec
2017-11-14 10:03:49,616 Stage-1 map = 65%,  reduce = 0%, Cumulative CPU 310.12 sec
2017-11-14 10:03:52,966 Stage-1 map = 70%,  reduce = 0%, Cumulative CPU 326.28 sec
2017-11-14 10:03:55,142 Stage-1 map = 75%,  reduce = 0%, Cumulative CPU 329.84 sec
2017-11-14 10:04:17,276 Stage-1 map = 80%,  reduce = 0%, Cumulative CPU 401.17 sec
2017-11-14 10:04:19,430 Stage-1 map = 85%,  reduce = 0%, Cumulative CPU 402.54 sec
2017-11-14 10:04:20,459 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 408.9 sec
MapReduce Total cumulative CPU time: 6 minutes 48 seconds 900 msec
Ended Job = job_1510651862836_0003
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://172.31.18.209/user/hive/warehouse/lineorder2/.hive-staging_hi
ve_2017-11-14_10-02-49_316_984824217666851051-1/-ext-10000
Loading data to table default.lineorder2
MapReduce Jobs Launched:
Stage-Stage-1: Map: 10   Cumulative CPU: 408.9 sec   HDFS Read: 2417934850 HDFS Writ
e: 2417757419 SUCCESS
Total MapReduce CPU Time Spent: 6 minutes 48 seconds 900 msec
OK
Time taken: 92.592 seconds
hive>
```

Ans: Hive – Single node

```
Loading data to table default.lineorder2
MapReduce Jobs Launched:
Stage-Stage-1: Map: 10   Cumulative CPU: 424.12 sec   HDFS Read: 2417934820 HDFS Write: 2417757419 SUCCESS
Total MapReduce CPU Time Spent: 7 minutes 4 seconds 120 msec
OK
Time taken: 319.199 seconds
hive>
```

Ans: Pig – 4-node

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.4   0.15.0  ec2-user        2017-11-17 00:53:57     2017-11-17 00:56:26     UNKNOWN

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTi
eature  Outputs
job_1510879763150_0001  18      0       91      29      58      50      0       0       0       0

Input(s):
Successfully read 23996604 records (2417832945 bytes) from: "/user/ec2-user/lineorder.tbl"

Output(s):
Successfully stored 23996604 records (3106341603 bytes) in: "/final1a_pig"

Counters:
Total records written : 23996604
Total bytes written : 3106341603
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1510879763150_0001
```

Time taken is 2min 29sec

Ans: Pig – Single node

```
r - 100% complete
2017-11-14 05:08:11,327 [main] INFO  org.apache.pig.tools.pigstats.mapreduce.SimplePigSta

HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.4   0.15.0  ec2-user        2017-11-14 05:04:33     2017-11-14 05:08:11     UNKNOWN

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   M
ime   AvgReduceTime   MedianReducetime        Alias   Feature     Outputs
job_1510632196402_0006  18      0       70      65      66      66      0       0       0
        MAP_ONLY        /final1a_pig,
```

time taken : 03min 38sec

Ans: Hadoop Streaming – 4- node

```
                Total time spent by all reduce tasks (ms)=114296
                Total vcore-milliseconds taken by all map tasks=928346
                Total vcore-milliseconds taken by all reduce tasks=114296
                Total megabyte-milliseconds taken by all map tasks=950626304
                Total megabyte-milliseconds taken by all reduce tasks=117039104
        Map-Reduce Framework
                Map input records=23996604
                Map output records=23996604
                Map output bytes=2441753167
                Map output materialized bytes=2489746483
                Input split bytes=1800
                Combine input records=0
                Combine output records=0
                Reduce input groups=23996604
                Reduce shuffle bytes=2489746483
                Reduce input records=23996604
                Reduce output records=23996604
                Spilled Records=83924039
                Shuffled Maps =18
                Failed Shuffles=0
                Merged Map outputs=18
                GC time elapsed (ms)=2601
                CPU time spent (ms)=237220
                Physical memory (bytes) snapshot=4819709952
                Virtual memory (bytes) snapshot=18846203904
                Total committed heap usage (bytes)=3379560448
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=2417826195
        File Output Format Counters
                Bytes Written=2441753167
17/11/14 09:56:01 INFO streaming.StreamJob: Output directory: /final1a_hs/

real    2m40.626s
user    0m4.180s
sys     0m0.184s
[ec2-user@ip-172-31-18-209 ~]$
```

Ans: Hadoop Streaming – Single- node

```
        File Output Format Counters
                Bytes Written=2441753167
17/11/14 05:00:36 INFO streaming.StreamJob: Output directory: /final1a_hs/

real    4m33.677s
user    0m4.624s
sys     0m0.208s
[ec2-user@ip-172-31-25-159 ~]$
```

b. Scale14: a cluster of at least 4 nodes (Steps and coding are EXACTLY the same as Question #1)

Hive Output:

```
hive> LOAD DATA LOCAL INPATH '/home/ec2-user/lineorder.tbl.1'  OVERWRITE INTO TABLE lineorder;
Loading data to table default.lineorder
OK
Time taken: 155.998 seconds
```

```
2017-11-16 08:55:02,745 Stage-1 map = 94%,  reduce = 0%, Cumulative CPU 1452.39 sec
2017-11-16 08:55:03,845 Stage-1 map = 95%,  reduce = 0%, Cumulative CPU 1453.32 sec
2017-11-16 08:55:06,007 Stage-1 map = 97%,  reduce = 0%, Cumulative CPU 1457.94 sec
2017-11-16 08:55:07,039 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1461.24 sec
MapReduce Total cumulative CPU time: 24 minutes 21 seconds 240 msec
Ended Job = job_1510821417449_0002
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://172.31.18.209/user/hive/warehouse/lineorder2/.hive-staging_hive_2017-11-16_08-52-43_237_4803619336523380232-1
/-ext-10000
Loading data to table default.lineorder2
MapReduce Jobs Launched:
Stage-Stage-1: Map: 33   Cumulative CPU: 1461.24 sec   HDFS Read: 8627800558 HDFS Write: 8543207696 SUCCESS
Total MapReduce CPU Time Spent: 24 minutes 21 seconds 240 msec
OK
Time taken: 145.542 seconds
hive>
```

Pig Output:

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.4   0.15.0  ec2-user        2017-11-16 19:33:18     2017-11-16 19:35:32     UNKNOWN

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime
job_1510859577880_0001  65      0       107     43      92      97      0       0       0       0

Input(s):
Successfully read 83988094 records (8627479603 bytes) from: "/user/ec2-user/lineorder.tbl.1"

Output(s):
Successfully stored 83988094 records (10969648711 bytes) in: "/final1a_pig"

Counters:
Total records written : 83988094
Total bytes written : 10969648711
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1510859577880_0001
```

time taken is 02min 14sec

Hadoop streaming Output:

```
17/11/16 09:16:46 INFO mapreduce.Job:  map 100% reduce 94%
17/11/16 09:16:52 INFO mapreduce.Job:  map 100% reduce 95%
17/11/16 09:16:55 INFO mapreduce.Job:  map 100% reduce 96%
17/11/16 09:17:04 INFO mapreduce.Job:  map 100% reduce 97%
17/11/16 09:17:07 INFO mapreduce.Job:  map 100% reduce 98%
17/11/16 09:17:13 INFO mapreduce.Job:  map 100% reduce 99%
17/11/16 09:17:19 INFO mapreduce.Job:  map 100% reduce 100%
17/11/16 09:17:23 INFO mapreduce.Job: Job job_1510821417449_0003 completed successfully
17/11/16 09:17:23 INFO mapreduce.Job: Counters: 51
        File System Counters
                FILE: Number of bytes read=26184581509
                FILE: Number of bytes written=35070953817
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=8627461728
                HDFS: Number of bytes written=8711181048
                HDFS: Number of read operations=198
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
```

```
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=8627455098
        File Output Format Counters
                Bytes Written=8711181048
17/11/16 09:17:23 INFO streaming.StreamJob: Output directory: /final1a_hs/

real    16m43.143s
user    0m6.632s
sys     0m0.372s
[ec2-user@ip-172-31-18-209 ~]$
```

```
[ec2-user@ip-172-31-18-209 ~]$ hadoop fs -cat /final1a_hs/part-00000 | more
1,1,103321,465569,11582,19960102,5-LOW,0,17,2608718,21280402,4,2504369,92072,2,19960212,TRUCK,
1,2,103321,201928,2281,19960102,5-LOW,0,36,6587676,21280402,9,5994785,109794,6,19960228,MAIL,
1,3,103321,191100,993,19960102,5-LOW,0,8,952880,21280402,10,857592,71466,2,19960305,REG AIR,
1,4,103321,6395,13190,19960102,5-LOW,0,28,3643892,21280402,9,3315941,78083,6,19960330,AIR,
1,5,103321,72080,22744,19960102,5-LOW,0,24,2524992,21280402,10,2272492,63124,4,19960314,FOB,
1,6,103321,46904,19141,19960102,5-LOW,0,32,5922880,21280402,7,5508278,111054,2,19960207,MAIL,
100,1,411610,186087,4388,19980228,4-NOT SPECI,0,28,3284624,21189463,4,3153239,70384,5,19980513,TRUCK,
100,2,411610,347935,11783,19980228,4-NOT SPECI,0,22,4362424,21189463,0,4362424,118975,7,19980412,SHIP,
100,3,411610,138450,9914,19980228,4-NOT SPECI,0,46,6846870,21189463,3,6641463,89307,4,19980410,SHIP,
```

B. Both of your solution for your 2-B

Scale 4: Single and a cluster of at least 4nodes

Hive (At least 4-cluster)

```
1640583696      1998    MFGR#1226
1565657860      1998    MFGR#1227
1607890751      1998    MFGR#1228
1350601347      1998    MFGR#1229
1470503353      1998    MFGR#123
1441898473      1998    MFGR#1230
1445039464      1998    MFGR#1231
1710140678      1998    MFGR#1232
1538979218      1998    MFGR#1233
1532309319      1998    MFGR#1234
1598713364      1998    MFGR#1235
1577658136      1998    MFGR#1236
1532687418      1998    MFGR#1237
1285428693      1998    MFGR#1238
1459545128      1998    MFGR#1239
1525737275      1998    MFGR#124
1587370161      1998    MFGR#1240
1477715730      1998    MFGR#125
1466946762      1998    MFGR#126
1686460729      1998    MFGR#127
1538644707      1998    MFGR#128
1207004714      1998    MFGR#129
Time taken: 128.035 seconds, Fetched: 280 row(s)
hive>
```

Hive : Single node:

```
1441898473      1998    MFGR#1230
1445039464      1998    MFGR#1231
1710140678      1998    MFGR#1232
1538979218      1998    MFGR#1233
1532309319      1998    MFGR#1234
1598713364      1998    MFGR#1235
1577658136      1998    MFGR#1236
1532687418      1998    MFGR#1237
1285428693      1998    MFGR#1238
1459545128      1998    MFGR#1239
1525737275      1998    MFGR#124
1587370161      1998    MFGR#1240
1477715730      1998    MFGR#125
1466946762      1998    MFGR#126
1686460729      1998    MFGR#127
1538644707      1998    MFGR#128
1207004714      1998    MFGR#129
Time taken: 202.004 seconds, Fetched: 280 row(s)
hive>
```

Pig: At Least 4 clusters:

First Phase:

```
2017-11-16 08:14:21,297 [main] INFO  org.apache.pig.tools.pigstats.mapreduce.SimplePigStats - Script Statis
tics:

HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.4   0.15.0  ec2-user        2017-11-16 08:10:26     2017-11-16 08:14:21     HASH_JOIN,FILTER

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   Min
ReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_1510819401388_0001  17      3       101     9       78      90      133     130     132     132     pre
Join,pre_supplier_join,supplier      HASH_JOIN
job_1510819401388_0002  4       1       39      14      32      38      40      40      40      40      new
table,part,pre_supplier_part_join,pre_supplier_part_join_filter      HASH_JOIN      hdfs://172.31.18.209/us
er/ec2-user/new_dataJoin2,

Input(s):
Successfully read 56900543 records from: "hdfs://172.31.18.209/user/ec2-user/preJoin_pig_1c"
Successfully read 40000 records from: "/user/ec2-user/supplier.tbl"
Successfully read 600000 records from: "/user/ec2-user/part.tbl"

Output(s):
Successfully stored 451185 records (12017904 bytes) in: "hdfs://172.31.18.209/user/ec2-user/new_dataJoin2"
```

Time taken for phase 1 is about 03min 55sec

Second Phase:

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.4   0.15.0  ec2-user        2017-11-15 02:24:26     2017-11-15 02:25:36     GROUP_BY,ORDER_BY

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   M
lias    Feature Outputs
job_1510712097546_0001  1       1       7       7       7       7       4       4       4       4       j
job_1510712097546_0002  1       1       4       4       4       4       3       3       3       3       r
job_1510712097546_0003  1       1       4       4       4       4       3       3       3       3       r
ec2-user/final_pig_2b,

Input(s):
Successfully read 451185 records (12018291 bytes) from: "hdfs://172.31.18.209/user/ec2-user/new_dataJoin"

Output(s):
Successfully stored 280 records (9145 bytes) in: "hdfs://172.31.18.209/user/ec2-user/final_pig_2b"
```

Time taken for phase 2 is about 01min 10sec

Pig -Single Cluster

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.4   0.15.0  ec2-user        2017-11-16 21:08:43     2017-11-16 21:15:31     HASH_JOIN,FILTER

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinR
job_1510865311285_0004  17      3       87      4       76      86      135     73      114     135     prel
job_1510865311285_0005  4       1       58      14      46      56      56      56      56      56      newt
1.18.209/user/ec2-user/new_dataJoin2,

Input(s):
Successfully read 56900543 records from: "hdfs://172.31.18.209/user/ec2-user/preJoin_pig_1c"
Successfully read 40000 records from: "/user/ec2-user/supplier.tbl"
Successfully read 600000 records from: "/user/ec2-user/part.tbl"

Output(s):
Successfully stored 451185 records (12017904 bytes) in: "hdfs://172.31.18.209/user/ec2-user/new_dataJoin2"

Counters:
Total records written : 451185
Total bytes written : 12017904
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1510865311285_0004  ->      job_1510865311285_0005,
job_1510865311285_0005
```

Time taken for phase 1 is 06min 48sec

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.4   0.15.0  ec2-user        2017-11-16 21:22:42     2017-11-16 21:23:50     GROUP_BY,ORDER_BY

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinRe
job_1510865311285_0006  1       1       6       6       6       6       3       3       3       3       join_
job_1510865311285_0007  1       1       3       3       3       3       3       3       3       3       resul
job_1510865311285_0008  1       1       3       3       3       3       3       3       3       3       resul

Input(s):
Successfully read 451185 records (12018292 bytes) from: "hdfs://172.31.18.209/user/ec2-user/new_dataJoin2"

Output(s):
Successfully stored 280 records (9145 bytes) in: "hdfs://172.31.18.209/user/ec2-user/final_pig_2b"

Counters:
Total records written : 280
Total bytes written : 9145
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1510865311285_0006  ->      job_1510865311285_0007,
job_1510865311285_0007  ->      job_1510865311285_0008,
job_1510865311285_0008
```

Time taken for phase 2 is 01min 08sec

Summarize the results and cluster performance/scaling in at least a paragraph.
Based on the above result, the performance of running task in different clusters set up achieve different performance. The multiple cluster setup apparently offers a better performance than the single node cluster in terms of process speed. In general speaking, the result above shows the time taken from the multi-nodes is about approximately 2-3 times faster than the Single node cluster. This result is very reasonable because the more worker clusters would share more works. However, I would say that the running speed also depends on the storage size and available space in the master and worker cluster. The limited size of storage size would hinder the cluster running performance.

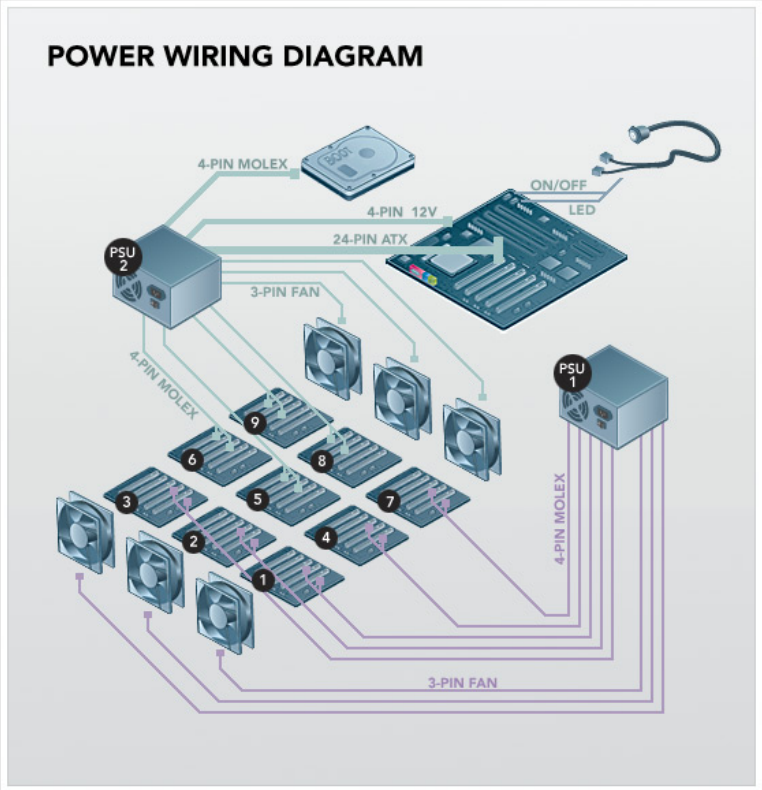
The size of dataset is also the matter to the cluster performance. Although the Scale14 dataset is about 4 times larger than the Scale4 dataset, the time consuming of Scale14 data is not 4 times slower than the Scale 4 dataset as shown above. As I said, the process speed also depends on the number of worker nodes, storage size and memory ram etc.

Extra Credit

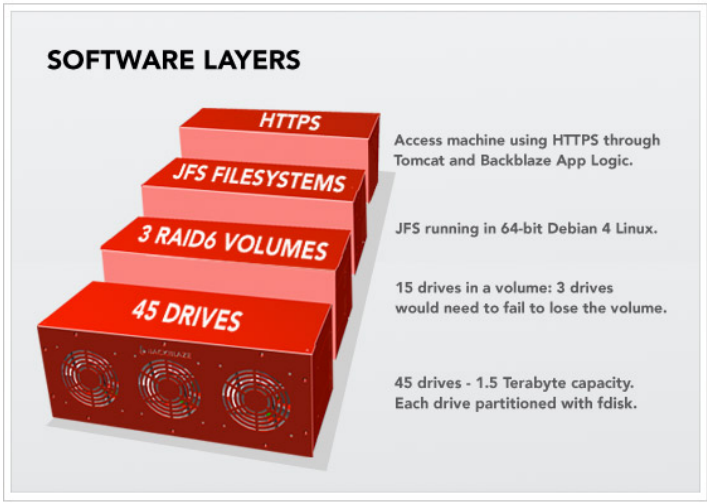
Research and describe the most affordable way to build a 1-Petabyte drive. Note that the setup has to be self-sufficient (i.e. easily usable) and include references. Buying 250 of 4TB drives is not enough because you still need a way to use it. The drive should be built to own, not to rent (Dropbox or similar services doesn't count, even if it does say "unlimited" storage).

Ans

1-Petabyte drive which means 1000 Terabyte equals to 1 milion of Gigabyte. Based on the amazon general pricing, the cost of 1TB hard drive cost about $45, so that the only cost of hard drive is about $4500 which does not include other parts of the system. Based on the online research, there is a brand which is called BLACKBLAZE provides the design for the 1-Petabyte hard drive system for about $110000. The structure of the system includes 15 pod stack in a rack. Each pod includes the following main parts: "one pod contains one Intel Motherboard with four SATA cards plugged into it. The nine SATA cables run from the cards to nine port multiplier backplanes that each have five hard drives plugged directly into them (45 hard drives in total)." For each hard drive, there is nylon and rubber band to helps on dampen vibration problem. The following was the block parts diagram from BLACKBLAZE. The cost of the system is dominated by the hard drive , the detail list of the cost is listed in the provided website link.



The Storage Pod is configured with free software includes 64-bit Debian 4 Linux and the JFS file system. Also, "they are self-contained appliances, where all access to and from the pods is through HTTPS". However, the pods functions does not include the iSCSI, no NFS, no SQL, no Fibre Channel.



By building stack of pod to have the 1-Petabyte hard drive is cheaper than other cloud provider e.g. Dell, Amazon. However, there is also some other cost (e.g. Space and Electrical and Maintenance fee etc) associated with this system that we need to take into account before building this system.

Reference: https://www.backblaze.com/blog/petabytes-on-a-budget-how-to-build-cheap-cloud-storage/

Submit a single document containing your written answers. Be sure that this document contains your name and "CSC 555 Project Phase 2" at the top.