Robert Kaszubski

CSC 555 Project Phase 2

**Part 1)**

Load in data:

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);

customer = LOAD '/user/ec2-user/customer.tbl' USING PigStorage('|') AS (c\_custkey:int, c\_name:chararray, c\_address:chararray, c\_city:chararray, c\_nation:chararray, c\_region:chararray, c\_phone:chararray, c\_mktsegment:chararray);

Testing:

lineorderG = GROUP lineorder ALL;

Count = FOREACH lineorderG GENERATE COUNT(lineorder);

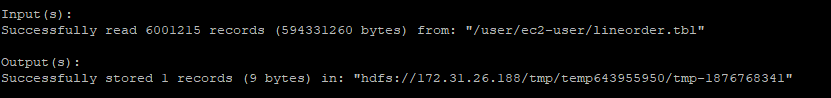
DUMP Count;

customerG = GROUP customer ALL;

Count = FOREACH customerG GENERATE COUNT(customer);

DUMP Count;

Data loads successfully:



A screenshot of a computer screen

Description automatically generated with medium confidence

PIG QUERY:

cfilter = FILTER customer BY c\_region == 'AFRICA';

lofilter = FILTER lineorder BY (lo\_discount == 6 OR lo\_discount > 8);

CLJoin = JOIN cfilter BY (c\_custkey), lofilter BY (lo\_custkey);

GByNation = GROUP CLJoin BY c\_nation;

Agg = FOREACH GByNation GENERATE group,

AVG(CLJoin.lo\_extendedprice) as AVGL;

SAgg = ORDER Agg BY $1 DESC;

STORE SAgg INTO 'Final1' using PigStorage(',');

Text

Description automatically generated

Output (DUMP SAgg;):

Text

Description automatically generated

Output written to HDFS:

Graphical user interface, text

Description automatically generated

**Part 2)**

Pass 1 (Join Lineorder and Part tables)

Pass1Mapper.py

Text

Description automatically generated

Key = partkey

Value = brand1\_part or revenue\_orderdate\_lo

Pass1Reducer.py

Perform the Join

Text

Description automatically generated

Key = partkey

Value = brand\_revenue\_orderdate

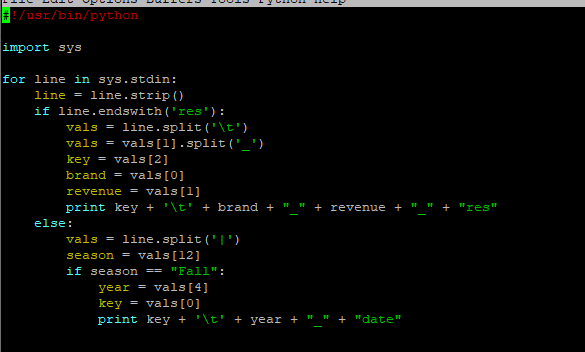
Sample Run using smaller lineorder table and saving Results:

Text

Description automatically generated

Pass 2 (join results + dwdate)

Pass2Mapper.py



Key: orderdate or datekey

Value: year\_date or brand\_revenue\_res

Pass2Reducer.py

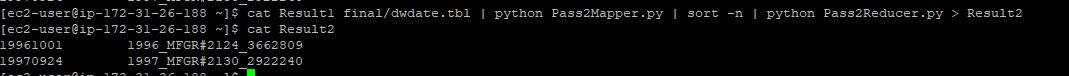
Key: orderdate/datekey

Value: year\_brand\_revenue

Text

Description automatically generated

Test run of Pass 2 using results of Pass1 (with smaller lineorder tbl)



Pass 3 (Group)

Pass3Mapper.py

Key = year\_brand

Value = revenue

A screenshot of a computer

Description automatically generated with medium confidence

Pass3Recuder.py

Key = year\_brand

Value = sum(revenue)

Text

Description automatically generated



RUNNING WITH HADOOP STREAMING:

Using full lineorder table now

Pass 1:

Create Pass 1 directory and place the lineorder and part tables in it.

A screenshot of a computer

Description automatically generated with medium confidence

Command:

hadoop jar hadoop-streaming-2.6.4.jar -input /data/Pass1 -mapper Pass1Mapper.py -file ../Pass1Mapper.py -reducer Pass1Reducer.py -file ../Pass1Reducer.py -output /data/Pass1/result

Results will be saved to /data/Pass1/result

Text

Description automatically generated with medium confidence

Text

Description automatically generated

Sample output of Pass 1:

Text

Description automatically generated with medium confidence

Pass 2:

Create directory for Pass 2

Move dwdate table and Result file from Pass 1:

Text

Description automatically generated

Command:

hadoop jar hadoop-streaming-2.6.4.jar -input /data/Pass2 -mapper Pass2Mapper.py -file ../Pass2Mapper.py -reducer Pass2Reducer.py -file ../Pass2Reducer.py -output /data/Pass2/result

Text

Description automatically generated

Results saved to /data/Pass2/result

Graphical user interface, text

Description automatically generated

Sample Output:

Graphical user interface, text

Description automatically generated

Pass 3:

Create directory for pass 3 and transfer Pass 2 results:



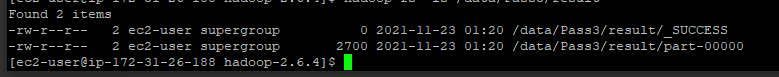
Command:

hadoop jar hadoop-streaming-2.6.4.jar -input /data/Pass3 -mapper Pass3Mapper.py -file ../Pass3Mapper.py -reducer Pass3Reducer.py -file ../Pass3Reducer.py -output /data/Pass3/result

A computer screen capture

Description automatically generated with medium confidence

Results saved to /data/Pass3/result



Final Output:

Graphical user interface

Description automatically generated with medium confidence

**Part 3)**

I generated my data using the following python code numgenfinal.py:

Text

Description automatically generated

Ran with the following command:



Part3data file:

A picture containing text, computer, keyboard

Description automatically generated

I also wrote a python file called rngcenters.py that read the data generated and picked 4 random points as the initial centers. It wrote them into a file, numbering each cluster with a tab between them just as the reducer will later return.

Text

Description automatically generated

Text

Description automatically generated

Mapper:

Key= Cluster#

Val = DataPoints

Text

Description automatically generated

A picture containing text

Description automatically generated

With sort -n

A picture containing text

Description automatically generated

Reducer

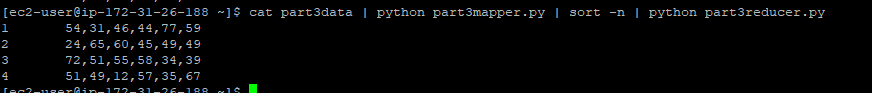
Key= Cluster#

Val = CenterPoints

Text

Description automatically generated

Output from reducer: new centers



Running with Hadoop Streaming:

I made a new directory in hadoop with data file

A screenshot of a computer

Description automatically generated with medium confidence

ITERATION 1:

Initial Centers:

Text

Description automatically generated

Command:

hadoop jar hadoop-streaming-2.6.4.jar -input /data/part3/part3data -file ../centers.txt -mapper part3mapper.py -file ../part3mapper.py -reducer part3reducer.py -file ../part3reducer.py -output /data/part3/iter1c

Graphical user interface, text

Description automatically generated

New Centers:

A screenshot of a computer

Description automatically generated with medium confidence

ITERATION 2

Replacing centers.txt with new centers:



Command:

hadoop jar hadoop-streaming-2.6.4.jar -input /data/part3/part3data -file ../centers.txt -mapper part3mapper.py -file ../part3mapper.py -reducer part3reducer.py -file ../part3reducer.py -output /data/part3/iter2

Text

Description automatically generated

New Centers:

Text

Description automatically generated

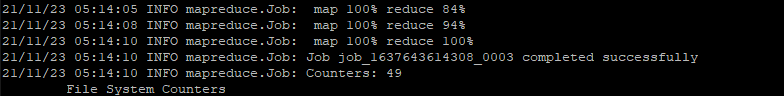
ITERATION 3

Replacing center.txt file



Command

hadoop jar hadoop-streaming-2.6.4.jar -input /data/part3/part3data -file ../centers.txt -mapper part3mapper.py -file ../part3mapper.py -reducer part3reducer.py -file ../part3reducer.py -output /data/part3/iter3



New Centers:

A picture containing text

Description automatically generated

ITERATION 4:

Replacing file:



Command

hadoop jar hadoop-streaming-2.6.4.jar -input /data/part3/part3data -file ../centers.txt -mapper part3mapper.py -file ../part3mapper.py -reducer part3reducer.py -file ../part3reducer.py -output /data/part3/iter4

Text

Description automatically generated

New Centers:

Text

Description automatically generated