DSC 333 Mining Big Data

Assignment 3

Nikhil Bhalala

1) Describe how to implement the following queries in MapReduce:

a) SELECT a.First, a.Last, e.EID, a.AID, e.Age
FROM Employee as emp, Agent as a
WHERE emp.Last = a.Last AND emp.First = a.First;

⇨ Map: Each tuple creates a key pair. (key=(emp,Agent),value=(a.First, a.Last, e.EID, a.AID, e.Age))
Reduce: if an only keys that meet the condition "emp.Last = a.Last AND emp.First = a.First" will only create an output a.First, a.Last, e.EID, a.AID, e.Age

b) SELECT lo_quantity, COUNT(lo_extendedprice)
FROM lineorder, dwdate
WHERE lo_orderdate = d_datekey
AND d_yearmonth = 'Feb1995'
AND lo_discount = 6
GROUP BY lo_quantity;

⇨ Map : key pair(lineorder ,dwdate) and value is (lo_quantity , COUNT(lo_extendedprice)) with group by lo_quantity;
Reduce: keys only meet the condition lo_orderdate = d_datekey , d_yearmonth = 'Feb1995' lo_discount = 6 create output lo_quantity and aggregate count COUNT(lo_extendedprice)

c) SELECT d_month, AVG(d_year)
FROM dwdate
GROUP BY d_month
ORDER BY AVG(d_year)

⇨ Map : key = dwdate value = d_month, AVG(d_year) GROUP BY d_month
Where for every dwdate make d_month the key and set month and year as value
Reduce:for every dwdat compute the aggregate Avg(d_year)

2) Consider a Hadoop job that processes an input data file of size equal to 179 disk blocks (179 different blocks, not considering HDFS replication factor). The mapper in this job requires 1 minute to read and fully process a single block of data. Reducer requires 1 second (**not** minute) to produce an answer for one key worth of values and there are a total of 3000

DSC 333 Mining Big Data

Assignment 3

Nikhil Bhalala

distinct keys (mappers generate a lot more key-value pairs, but keys only occur in the 1-3000 range for a total of 3000 unique entries). Assume that each node has a reducer and that the keys are distributed evenly.
The total cost will consist of time to perform the Map phase plus the cost to perform the Reduce phase.

a) How long will it take to complete the job if you only had one Hadoop worker node? For simplicity, assume that that only one mapper and only one reducer are created on every node.

⇨ Map = no.of.block*time= 179*1=179min
Reduce time*Unique key=(1/60)*3000 = 50
Total=Map+ Reduce =179+50 =229

b) 30 Hadoop worker nodes?

⇨ At 4 block
Map = 4*1 = 4min
Reduce (3000/30)*(1/60)= 1.67min
Total= 4+1.67=5.67min

c) 60 Hadoop worker nodes?

⇨ At 3 block
Map = 3*1 = 3min
Reduce (3000/60)*(1/60)=0.833min
Total= 3+0.833min=3.83min

d) 100 Hadoop worker nodes?

⇨ At 2 block
Map = 2*1 = 2min
Reduce (3000/100)*(1/60)=0.5min
Total= 2+0.5 =2.5min

e) Would changing the replication factor have any affect your answers for a-d?

⇨ There is no influence of the replication factor on a-d because we ignore network cost.

You can ignore the network transfer costs as well as the possibility of node failure.

3)

a) Suppose you have an 8-node cluster with replication factor of 3. Describe what MapReduce has to do after it determines that a node has crashed while a job is being processed. For simplicity, assume that the failed node is not replaced and your cluster is reduced to 7 nodes. Specifically:

DSC 333 Mining Big Data

Assignment 3

Nikhil Bhalala

i) What does HDFS (the storage layer/NameNode) have to do in response to node failure in this case? I.e., what is the guarantee that HDFS has to maintain?

⇨ No one can utilize the filesystem without the name node. There are two approaches to prevent data loss in the event of failure: 1, backup metadata files (replication factor). 2. To stop the edit log from growing too big, run a supplementary name node that enables merging the namespace image with the edit log.

ii) What does MapReduce engine (the execution layer) have to do to respond to the node failure? Assume that there was a job in progress at the time of the crash (because MapReduce engine only needs to take action if a job was in progress).

⇨ The entire job must be restarted if the name node fails. If the data node fails, the name node notices the failure, and all map tasks are redone while the failed node is set to idle and another worker is assigned to complete the task. The name node idles the node and reschedules for a different reduce worker if the data node for the reduce operation fails.

b) Where does the Mapper store output key-value pairs before they are sent to Reducers?

⇨ Keys are stored locally

c) Can Reducers begin processing before Mapper phase is complete? **Why or why not?**

⇨ Because they require the keys to complete their action, the reducers must wait for the mapper.

4) Repeat the RSA computation examples by

a) Select two (small) primes and generate a public-private key pair.

⇨ R=19 S= 11
N= 209
(n)=(r-1)(s-1) = 18*10 =180
Gcd(e,180)—e = 7
De= 1 Mod 180
D=103
Ku= {7,209}
Kr={103, 209}

b) Compute a sample ciphertext using your public key

⇨ 10^7 MOD 209= 186

c) Decrypt your ciphertext from 4-b using the private key

=> 186^3 MOD 209 = 10

d) Why can't the encrypted message sent through this mechanism be larger than the value of n?

⇨ Due to the MOD procedure, the message delivered cannot be greater than n.

DSC 333 Mining Big Data

Assignment 3

Nikhil Bhalala

5) Using the SSBM schema (http://cdmgcsarprd01.dpu.depaul.edu/CSC555/SSBM1/SSBM_schema_hive.sql) load the Part table into Hive (data available at http://cdmgcsarprd01.dpu.depaul.edu/CSC555/SSBM1/part.tbl)

   NOTE: By default Hive assumes '\t' separated tables. You will need to modify your CREATE TABLE statement provided above to account for the '|' delimiter in the data.

   Use Hive user defined function to perform the following transformation on Part table (creating a new PartSwapped table with the same number of columns): in the 7th column/p_type swap the first and last word in the column and replace the spaces by a comma. For example, STANDARD BRUSHED TIN would become TIN, BRUSHED,STANDARD. For the rest of the columns, where applicable, replace # character by a '^' character, e.g., so that MFGR#4 becomes MFGR^4.

   Keep in mind that your transform python code (split/join) should **always** use tab ('\t') between fields even if the source data is |-separated. You can also take a look at the transform example included with this assignment for your reference (Examples_Assignment3.doc) which deliberately uses a different delimiter ('?').

```
create table part (
  p_partkey     int,
  p_name        varchar(22),
  p_mfgr        varchar(6),
  p_category    varchar(7),
  p_brand1      varchar(9),
  p_color       varchar(11),
  p_type        varchar(25),
  p_size        int,
  p_container   varchar(10)
) ROW FORMAT DELIMITED FIELDS TERMINATED BY '|' STORED AS TEXTFILE;
LOAD DATA LOCAL INPATH '/home/ec2-user/part.tbl' OVERWRITE INTO TABLE
part;
```

DSC 333 Mining Big Data

Assignment 3

Nikhil Bhalala

```
hive> create table part (
    >   p_partkey     int,
    >   p_name        varchar(22),
    >   p_mfgr        varchar(6),
    >   p_category    varchar(7),
    >   p_brand1      varchar(9),
    >   p_color       varchar(11),
    >   p_type        varchar(25),
    >   p_size        int,
    >   p_container   varchar(10)
    > ) ROW FORMAT DELIMITED FIELDS TERMINATED BY '|' STORED AS TEXTFILE;
OK
Time taken: 2.94 seconds
hive> LOAD DATA LOCAL INPATH '/home/ec2-user/part.tbl '
    > OVERWRITE INTO TABLE part;
FAILED: SemanticException Line 1:23 Invalid path ''/home/ec2-user/part.tbl '': No files mat
ching path file:/home/ec2-user/part.tbl%20
hive> LOAD DATA LOCAL INPATH '/home/ec2-user/part.tbl' OVERWRITE INTO TABLE part;
Loading data to table default.part
OK
Time taken: 1.258 seconds
hive> SELECT * FROM part  LIMIT 5
    > SELECT * FROM part  LIMIT 5;
FAILED: ParseException line 2:0 missing EOF at 'SELECT' near '5'
hive> SELECT *FROM part LIMIT 10;
OK
1       lace spring     MFGR#1  MFGR#11 MFGR#1121       goldenrod       PROMO BURNISHED COP
PER     7       JUMBO PKG
2       rosy metallic   MFGR#4  MFGR#43 MFGR#4318       blush   LARGE BRUSHED BRASS     1 L
G CASE
```

/my_python.py

```
#!/usr/bin/python
import sys
for line in sys.stdin:

        line=line.strip()
        v= line.split('\t')
        val6=v[6]
        val2=v[2]
        val3=v[3]
        val4=v[4]
        w1,w2,w3=val6.split(' ')
        w4,w5=val2.split('#')
        w6,w7=val3.split('#')
        w8,w9=val4.split('#')
        v[2]=w4 +'^'+w5
        v[3]=w6+'^'+w7
        v[4]=w8+'^'+w9
        v[6]=w3+','+w2+','+w1
        print( '\t'.join(v))
```

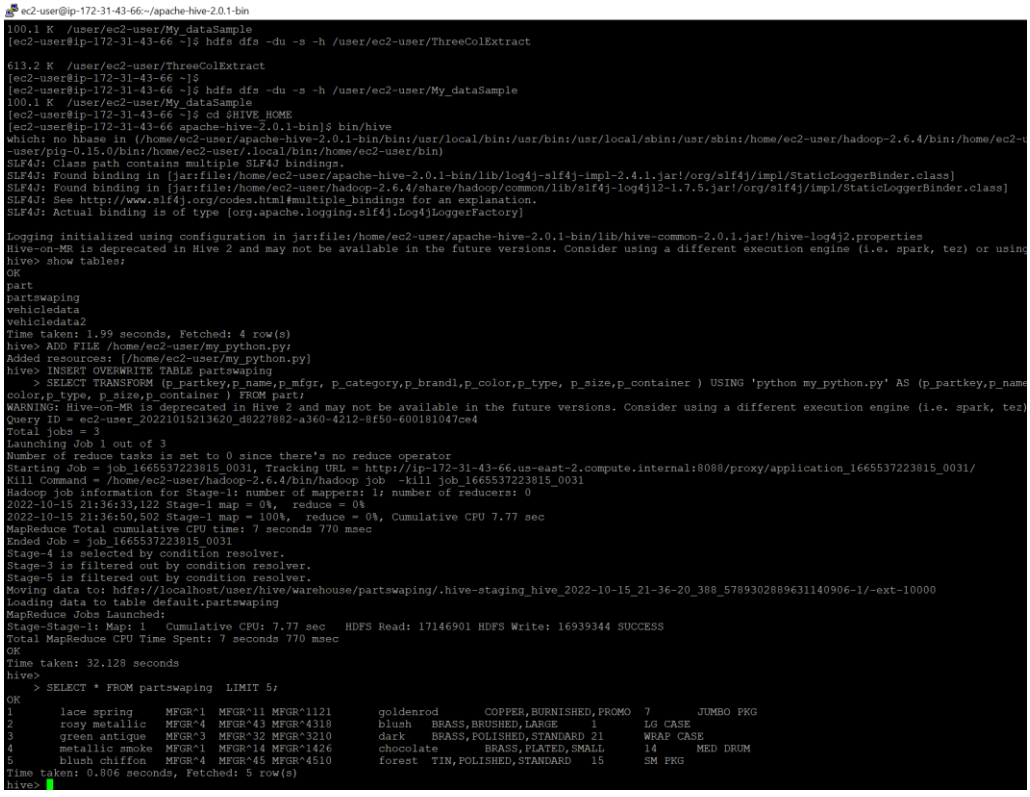
```
create table partswaping (
 p_partkey int,
 p_name varchar(22),
p_mfgr varchar(6),
 p_category varchar(7),
 p_brand1 varchar(9),
```

```
 p_color varchar(11),
 p_type varchar(25),
 p_size int,
 p_container varchar(10)
)

ROW FORMAT DELIMITED FIELDS
TERMINATED BY '\t' STORED AS TEXTFILE;

ADD FILE /home/ec2-user/my_python.py;

INSERT OVERWRITE TABLE partswaping
SELECT TRANSFORM (p_partkey,p_name,p_mfgr, p_category,p_brand1,p_color,p_type,
p_size,p_container ) USING 'python my_python.py' AS (p_partkey,p_name,p_mfgr,
p_category,p_brand1,p_color,p_type, p_size,p_container ) FROM part;
```



6) Download and install Pig:

```
cd
wget http://cdmgcsarprd01.dpu.depaul.edu/CSC555/pig-0.15.0.tar.gz
gunzip pig-0.15.0.tar.gz
tar xvf pig-0.15.0.tar
```

DSC 333 Mining Big Data

Assignment 3

Nikhil Bhalala

set the environment variables (this can also be placed in ~/.bashrc to make it permanent)

```
export PIG_HOME=/home/ec2-user/pig-0.15.0
export PATH=$PATH:$PIG_HOME/bin
```

Use the same vehicles file. Copy the vehicles.csv file to the HDFS if it is not already there.

Now run pig (and use the pig home variable we set earlier):

```
cd $PIG_HOME
bin/pig
```

Create the same table as what we used in Hive, assuming that vehicles.csv is in the home directory on HDFS:

```
VehicleData = LOAD '/user/ec2-user/vehicles.csv' USING PigStorage(',')
AS (barrels08:FLOAT, barrelsA08:FLOAT, charge120:FLOAT, charge240:FLOAT, city08:FLOAT);
```

You can see the table description by

```
DESCRIBE VehicleData;
```

Verify that your data has loaded by running:

```
VehicleG = GROUP VehicleData ALL;
Count = FOREACH VehicleG GENERATE COUNT(VehicleData);
DUMP Count;
```

How many rows did you get? (if you get an error here, it is likely because vehicles.csv is not in HDFS)

⇨ 34174

Create the same ThreeColExtract file that you have in the previous assignment, by placing barrels08, city08 and charge120 into a new file using PigStorage. You want the STORE command to record output in HDFS. (discussed in p457, Pig Chapter, "Data Processing Operator section)

For example, you can use this to get one column (multiple columns are comma-separated)

```
OneCol = FOREACH VehicleData GENERATE barrels08;
```

Verify that the new file has been created and report the size of the newly created file. (you can use **quit** to exit the grunt shell)

⇨ 100.1 k report the size of the newly created file.

DSC 333 Mining Big Data

Assignment 3

Nikhil Bhalala

Submit a single document containing your written answers. Be sure that this document contains your name and "CSC 555 Assignment 3" at the top.

ec2-user@ip-172-31-43-66:~/pig-0.15.0

```
 - Script Statistics:

HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.4   0.15.0  ec2-user        2022-10-15 19:28:24     2022-10-15 19:28:56     GROUP_BY

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   Max
ReduceTime      MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Out
puts
job_1665537223815_0029  1       1       6       6       6       6       5       5       5 5
Count,VehicleData,VehicleG      GROUP_BY,COMBINER       hdfs://localhost/tmp/temp1646436175
/tmp1688766387,

Input(s):
Successfully read 34175 records (11766951 bytes) from: "/user/ec2-user/vehicles.csv"

Output(s):
Successfully stored 1 records (9 bytes) in: "hdfs://localhost/tmp/temp1646436175/tmp1688766
387"

Counters:
Total records written : 1
Total bytes written : 9
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1665537223815_0029


2022-10-15 19:28:56,678 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to
ResourceManager at localhost/127.0.0.1:8032
2022-10-15 19:28:56,682 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Appli
cation state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history ser
ver
2022-10-15 19:28:56,741 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to
ResourceManager at localhost/127.0.0.1:8032
2022-10-15 19:28:56,749 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Appli
cation state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history ser
ver
2022-10-15 19:28:56,799 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to
ResourceManager at localhost/127.0.0.1:8032
2022-10-15 19:28:56,810 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Appli
cation state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history ser
ver
2022-10-15 19:28:56,883 [main] WARN  org.apache.pig.backend.hadoop.executionengine.mapReduc
eLayer.MapReduceLauncher - Encountered Warning FIELD_DISCARDED_TYPE_CONVERSION_FAILED 5 tim
e(s).
2022-10-15 19:28:56,889 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduc
eLayer.MapReduceLauncher - Success!
2022-10-15 19:28:56,892 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.
default.name is deprecated. Instead, use fs.defaultFS
2022-10-15 19:28:56,895 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.sche
matuple] was not set... will not generate code.
2022-10-15 19:28:56,927 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat
- Total input paths to process : 1
2022-10-15 19:28:56,932 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.Map
RedUtil - Total input paths to process : 1
(34174)
grunt>
```

DSC 333 Mining Big Data

Assignment 3

Nikhil Bhalala

```
2022-10-15 19:45:06,807 [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        2022-10-15 19:44:45     2022-10-15 19:45:06     UNKNOWN

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_1665537223815_0030  1       0       7       7       7       7       0       0       0       0       My_data MAP_ONLY        hdfs://localhost/user/ec2-user/My_dataSample,

Input(s):
Successfully read 34175 records (628255 bytes) from: "hdfs://localhost/user/ec2-user/ThreeColExtract"

Output(s):
Successfully stored 34175 records (102525 bytes) in: "hdfs://localhost/user/ec2-user/My_dataSample"

Counters:
Total records written : 34175
Total bytes written : 102525
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1665537223815_0030


2022-10-15 19:45:06,809 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at localhost/127.0.0.1:8032
2022-10-15 19:45:06,823 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2022-10-15 19:45:06,893 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at localhost/127.0.0.1:8032
2022-10-15 19:45:06,903 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2022-10-15 19:45:06,948 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at localhost/127.0.0.1:8032
2022-10-15 19:45:06,957 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2022-10-15 19:45:07,015 [main] WARN  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning FIELD_DISCARDED_TYPE_CONVERSION_FAILED 34175 ti
me(s).
2022-10-15 19:45:07,019 [main] WARN  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning ACCESSING_NON_EXISTENT_FIELD 68350 time(s).
2022-10-15 19:45:07,019 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
grunt>
```

```
[ec2-user@ip-172-31-43-66 ~]$
[ec2-user@ip-172-31-43-66 ~]$ hdfs dfs -du -s -h /user/ec2-user/My_dataSample
100.1 K  /user/ec2-user/My_dataSample
[ec2-user@ip-172-31-43-66 ~]$
```