**Ronaldlee Ejalu**

**CSC 555 Mining Big Data**

Assignment 4

1. Consider a Hadoop job that will result in 89 blocks of output to HDFS.

Suppose that writing one output block to HDFS takes 1 minute. The HDFS replication factor is set to 3 unless otherwise noted (the cost of writing output should include the replication blocks, although the copies are written by HDFS rather than by reducer. In fact, reducer doesn’t write any of the blocks, it just produces the computation result).

* 1. How long will it take for the reducer to write the job output on a 5-node Hadoop cluster? (ignoring the cost of Map processing, but counting replication cost in the output writing).

Writing one output block takes 1 min

Also, reading one output block takes 1 min.

HDFS replication factor is set to 3.

There are 89 blocks of output to be read and written.

So, a 5 node Hadoop cluster:

89/5 = 17.8

It takes 1 min to read a block and 1 min to write a block therefore:

17.8 min \* 3 = 53.4 mins.

* 1. How long will it take for reducer(s) to write the job output to 10 Hadoop worker nodes? (Assume that data is distributed evenly and replication factor is set to 1)

Replication factor = 1

10 nodes to process 89 blocks

So 89/10 = 8.9mins

Since the replication factor is 1 so 8.9min \* 1 = 8.9mins

* 1. How long will it take for reducer(s) to write the job output to 10 Hadoop worker nodes? (Assume that data is distributed evenly and replication factor is set to 3)

10 Hadoop worker nodes

Node 89/10 = 8.9 mins

Since Replication factor = 3

8.9 mins \* 3 = 26.7 mins

* 1. How long will it take for reducer(s) to write the job output to 100 Hadoop worker nodes? (Assume that data is distributed evenly and replication factor is set to 1)

For 100 Hadoop worker nodes with a replication factor of 1 will take 1 min.

* 1. How long will it take for reducer(s) to write the job output to 100 Hadoop worker nodes? (Assume that data is distributed evenly and replication factor is set to 3)

Node: 89/100 mins

Replication factor of 3 is equal: (89/100) \* 3 = 2.67mins

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

1. Repeat the exercise from Lecture 3 (RSA examples)
   1. Select two (small) primes and generate a public-private key pair.

* p = 5 , q = 11
* Compute their system modulus n = p \* q = 5 \* 11 = 55
* Compute Ø(n) = (p – 1)(q – 1) = (5-1)(11-1) = 40
* Select e such it is co-prime with Ø(n) = 40 , N = 55 and it must be less than Ø(n), 40, where gcd(e, 40) = 1 so let e = 3.
* We determine d: de = 1 mod 40 and d < 40

We got pick a number such that 3d (mod 40) = 1

3, 9, 12, 15, 18, 21, 24, 27, 30…………….

So d = 27

Where 3\* 27 mod 40 = 1

81 mod 40 =1

* Published public key KU = {3, 55}
* Private key = {27, 55}
  1. Compute a sample ciphertext using your public key

Given message M = 12 (12 < 55)

Encryption KU = {3, 55} such that

C = 12^3 mod 55 = 23

* 1. Decrypt your ciphertext from 2-b using the private key

Decryption KR = {27, 55}

M = 23^27 mod 55

* [(23^4 mod 55) \* (23^4 mod 55) \* (23^2 mod 55) \* (23^4 mod 55) \* (23^4 mod 55) \* (23^2 mod 55)] mod 55 = 12
  1. Why can’t the encrypted message sent through this mechanism be larger than the value of n?

All the computation takes the remainder of 55, a single message can’t exceed n, 55, because if you send 56, it will be truncated to 1, which usually makes the computation expensive, and this is used to exchange a regular password.

If it is larger than the value of n, it will be split.

1. Given the following keys: 1, 4, 5, 8, 11, 12, 14, 15, 17, 25, 26, 28, 50, 51, 59, 87, 89, 93, 98, design the following:
   1. A distribution of these keys across 3 reducers using the default key partitioner (% 3)

The keys are distributed across 3 reducers based on the default key partitioner (%3) where the remainder is 0, 1, 2 distributed amongst three reducers R0, R1 and R3 respectively as below:

R0 = {12, 15, 51, 87, 93}

R1 = {1, 4, 25, 28}

R2 = {5, 8, 11, 14, 17, 26, 50, 59, 89, 98}

* 1. Design a custom sorting partitioner instead of the default one and describe the resulting output across the same 3 reducers

If key <= 25: /\*all keys less than or equal to 25 will be assigned\*/

Reducer\_0

Elfif key > 25 and key <= 50:

Reducer\_1

else:

Reducer\_2

Reducer\_0, Reducer\_1, Reducer\_2 will be the output of the custom Partitioner and the keys will be distributed to the reducers in the given ranges above.

Reducer\_0 = {1, 4, 5, 8, 11, 12, 14, 15, 17, 25}

Reducer\_1 = {26, 28, 50}

Reducer\_2 = {51, 59, 87, 89, 93, 98}

* 1. What is the downside (i.e., extra overhead) of employing a custom partitioner?

You need to run sampling in order to create a custom partitioner; this involves sampling the data and finding a distribution so that you can come up with a decision. The process ends up being so expensive and time consuming, especially when you end up with a wrong distribution; you will have to do it again until you get it right.

1. Implement the following query using Hadoop streaming and python with the lineorder table.

<http://cdmgcsarprd01.dpu.depaul.edu/CSC555/SSBM1/SSBM_schema_hive.sql>

<http://cdmgcsarprd01.dpu.depaul.edu/CSC555/SSBM1/lineorder.tbl>

SELECT lo\_shipmode, STDDEV(lo\_tax)

FROM lineorder

WHERE lo\_quantity BETWEEN 17 AND 24

GROUP BY lo\_shipmode;

STDDEV is standard deviation.

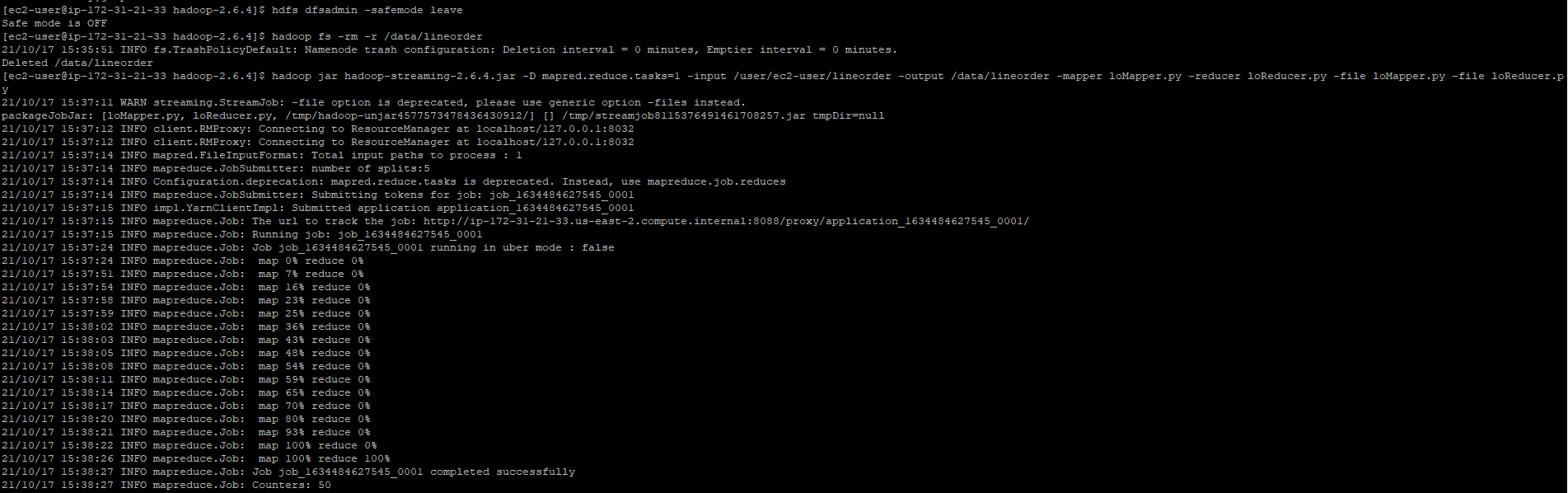
loMapper.py

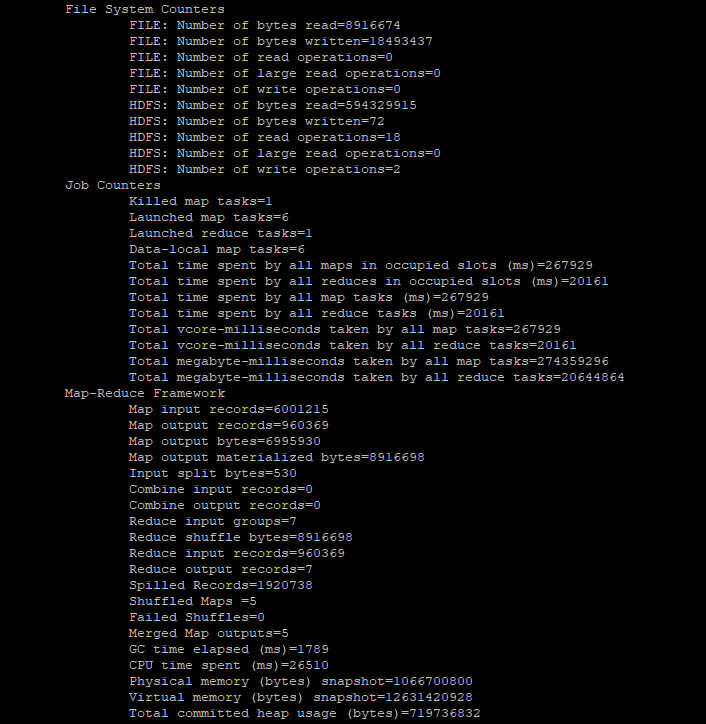


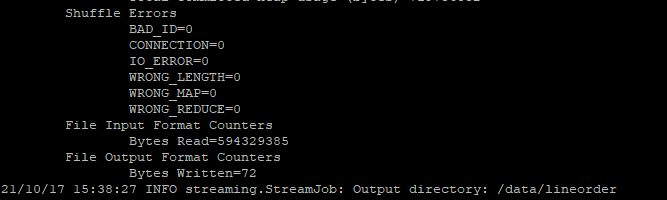
loReducer.py

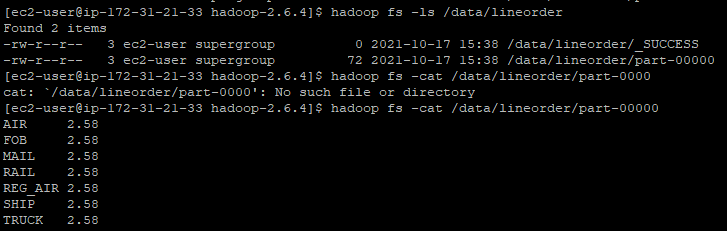


Don’t forget to submit your python code and the command lines you used to execute Hadoop streaming. I also recommend submitting the screenshot of execution to simplify the grader’s job.









1. In this section you will practice using HBase. Note that HBase runs on top of HDFS, bypassing the MapReduce engine.

cd

(Download HBase)

wget http://dbgroup.cdm.depaul.edu/Courses/CSC555/hbase-0.90.3.tar.gz

gunzip hbase-0.90.3.tar.gz

tar xvf hbase-0.90.3.tar

cd hbase-0.90.3

(Start HBase service, there is a corresponding stop service and this assumes Hadoop home is set)

bin/start-hbase.sh

(Open the HBase shell – at this point jps should show HMaster)

bin/hbase shell

(Create an employee table and two column families – private and public. Please watch the quotes, if **'** turns into **‘**, the commands will not work)

**create 'employees', {NAME=>** **'private'}, {NAME=>** **'public'}**

**put 'employees', 'ID1', 'private:ssn', '111-222-334'**

**put 'employees', 'ID2', 'private:ssn', '222-338-446'**

**put 'employees', 'ID3', 'private:address', '123 State St.'**

**put 'employees', 'ID1', 'private:address', '243 N. Wabash Av.'**

**scan 'employees'**

Now that we have filled in a couple of values, add 3 new columns to the private family, 1 new column to the public family and create a brand new family with at least 2 columns. For each of these you should introduce at least 2 values -- so a total of (3+1+2) \* 2 = 12 values inserted. Verify that the table has been filled in properly with scan command and submit a screenshot.

**put 'employees', 'ID1', 'private:firstname', 'Ronaldlee'**

**put 'employees', 'ID1', 'private:lastname', 'Ejalu'**

**put 'employees', 'ID1', 'private:age', '35'**

**put 'employees', 'ID2', 'private:firstname', 'Stevens'**

**put 'employees', 'ID2', 'private:lastname', 'Smith'**

**put 'employees', 'ID2', 'private:age', '46'**

**put 'employees', 'ID1', 'public:residentialstatus', 'Yes'**

**put 'employees', 'ID2', 'public:residentialstatus', 'Yes'**

**disable 'employees'**

**alter 'employees', 'department details'**

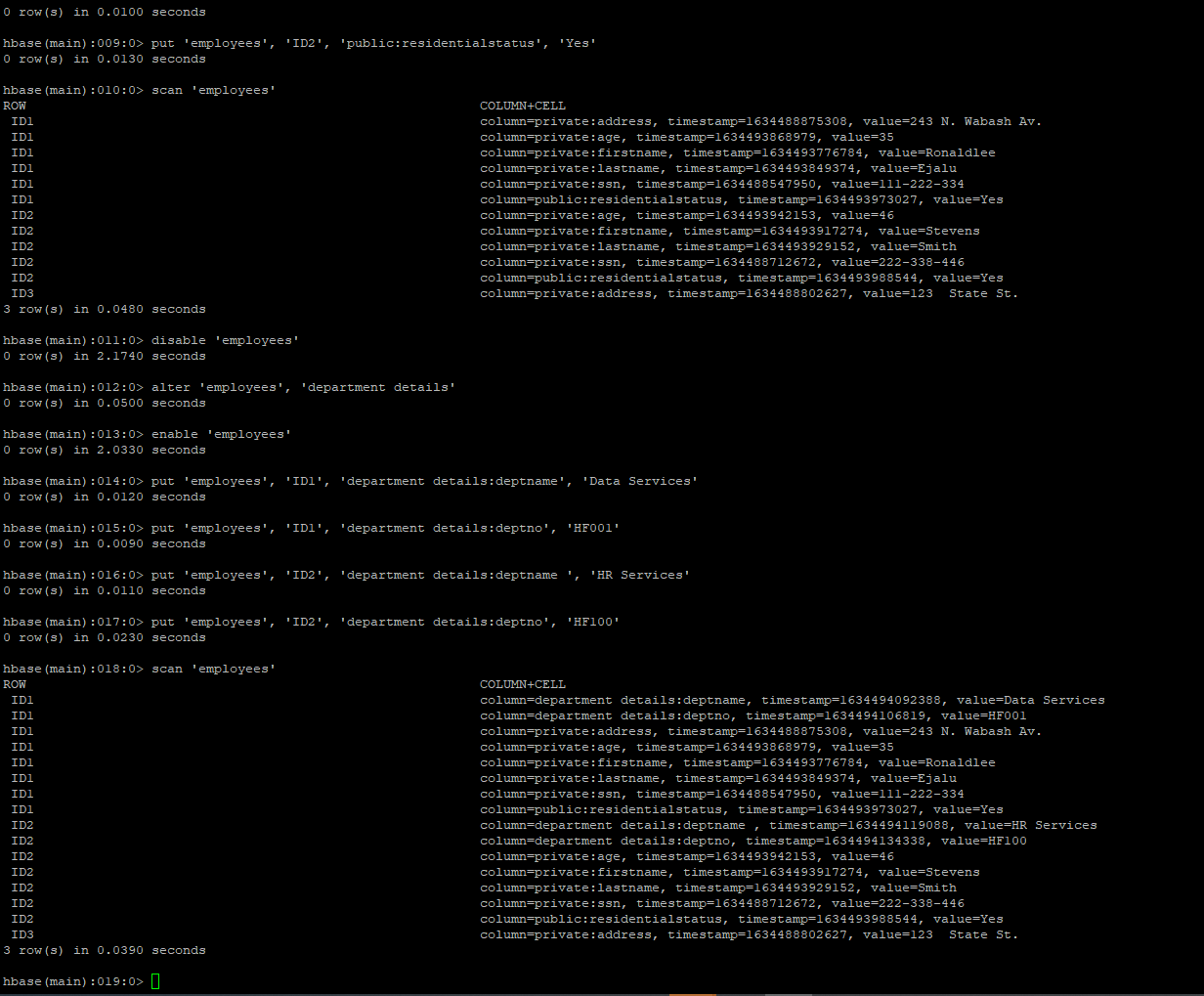
**enable 'employees'**

**put 'employees', 'ID1', 'department details:deptname', 'Data Services'**

**put 'employees', 'ID1', 'department details:deptno', 'HF001'**

**put 'employees', 'ID2', 'department details:deptname ', 'HR Services'**

**put 'employees', 'ID2', 'department details:deptno', 'HF100'**



NOTE: In order to add a new column family to an HBase table called test, you would need to run the following commands

**disable 'test'**

**alter 'test', 'myNewFavoriteColumnFamily'**

**enable 'test'**

Submit a single document containing your written answers. Be sure that this document contains your name and “CSC 555 Assignment 4” at the top.