**CSC 555: Mining Big Data**

Project, Phase 1 (due Wednesday, February 20th)

In this part of the project (which will also serve as our take-home midterm), you will 1) Set up a 4-node cluster and 2) perform data warehousing and transformation queries using Hive, Pig and Hadoop streaming. The modified Hive-style schema is at:

http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/SSBM1/SSBM\_schema\_hive.sql

It is based on SSBM benchmark (derived from industry standard TPCH benchmark). The data is at Scale1, or the smallest unit – lineorder is the largest table at about 0.6GB. You can use wget to download the following links. Keep in mind that data is |-separated (not csv).

<http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/SSBM1/dwdate.tbl>

<http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/SSBM1/lineorder.tbl>

<http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/SSBM1/part.tbl>

<http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/SSBM1/supplier.tbl>

<http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/SSBM1/customer.tbl>

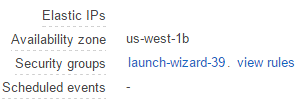
Please be sure to submit all code (pig, python and SQL).

# Part 1: Multi-node cluster

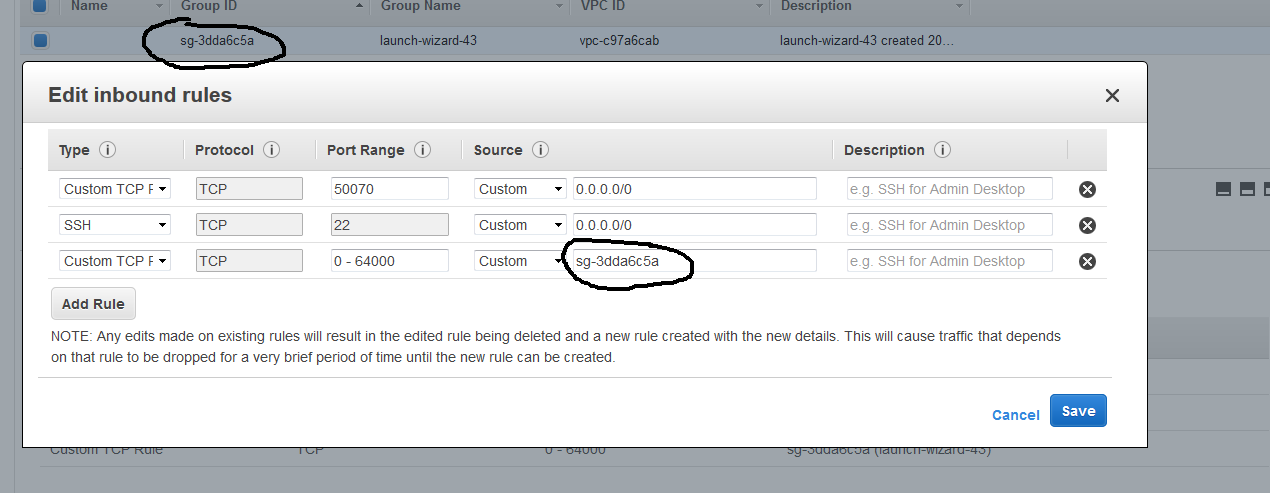
1. Your first step is to setup a multi-node cluster and re-run a simple wordcount. For this part, you will create a 4-node cluster (with a total of 1 master + 3 worker nodes). Include your master node in the “slaves” file, to make sure all 4 nodes are working.

You need to perform the following steps:

1. Create a new node of a medium size (you can always switch the size of the node). It is possible, but I do not recommend trying to reconfigure your existing Hadoop into this new cluster (it is much easier to make 4 new nodes for a total of 5 in your AWS account).
   1. **When creating a node I recommend changing the default 8G hard drive to 30G on all nodes.**
   2. Change your security group setting to open firewall access. We need to open the ports in two different ways. We will open port 50070 for the web interface in order to be able to see the cluster status in a browser. We will also set 0-64000 range opening up all ports. However, we will ensure that the ports are open only **within** the cluster and not to the world.

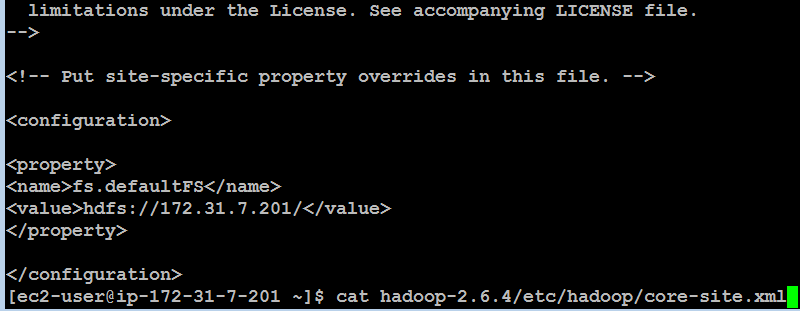
In order to make changes, you need to do the following. Access the cluster security group (launch-wizard-xx). Right click on the security group and choose Edit inbound rules

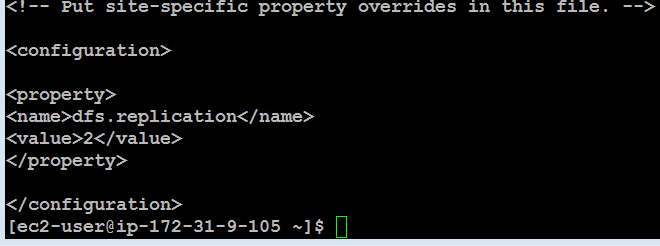
Note that the first line below is opening port 50070. The second line below is the default (port 22 is required for regular SSH connections). The third line opens all ports but ONLY for the same security group (assuming that all of your nodes in the cluster share the same security group – that will happen automatically if you use the “create more like this” option when creating instances as specified in part 1-c below). We had some issues with machines being hacked and disabled without that last limitation, so make sure you include it.



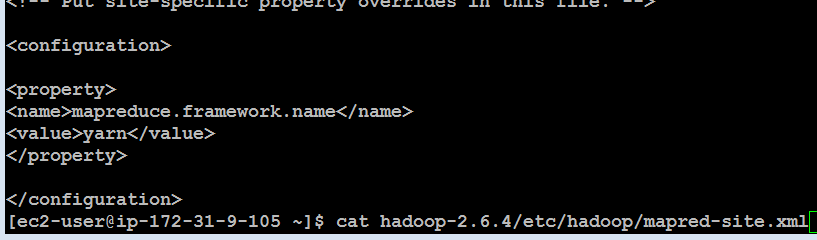
* 1. Right click on the Master node and choose “create more like this” to create 3 more nodes with same settings. If you configure the network settings on master first, security group information will be copied.

NOTE: Hard drive size will not be copied and default to 8G unless you change it.

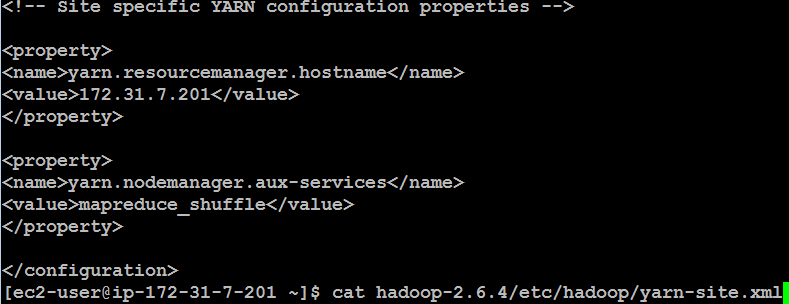
1. Connect to the master and set up Hadoop similarly to what you did previously. Do not attempt to repeat these steps on workers yet – you will only need to set up Hadoop once.
   1. Configure core-site.xml, adding the **PrivateIP** (do not use public IP) of the master. 
   2. Configure hdfs-site and set replication factor to 2.



* 1. cp hadoop-2.6.4/etc/hadoop/mapred-site.xml.template hadoop-2.6.4/etc/hadoop/mapred-site.xml and then configure mapred-site.xml



* 1. Configure yarn-site.xml (once again, use PrivateIP of the master)



Finally, edit the slaves file and list your 4 nodes (master and 3 workers) using Private IPs

[ec2-user@ip-172-31-7-201 ~]$ cat hadoop-2.6.4/etc/hadoop/slaves

172.31.7.201

172.31.5.246

…

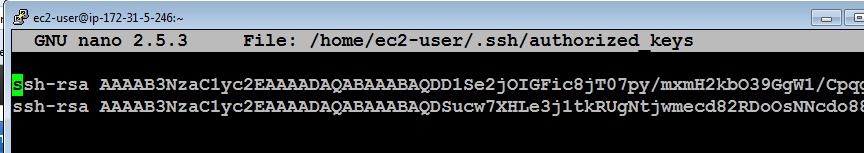
Make sure that you use private IP (private DNS is also ok) for your configuration files (such as conf/masters and conf/slaves or the other 3 config files). The advantage of the Private IP is that it does not change after your instance is stopped (if you use the Public IP, the cluster would need to be reconfigured every time it is stopped). The downside of the Private IP is that it is only meaningful within the Amazon EC2 network. So all nodes in EC2 can talk to each other using Private IP, but you cannot connect to your instance from the outside (e.g., from your laptop) because Private IP has no meaning for your laptop (since your laptop is not part of the Amazon EC2 network).

Now, we will pack up and move Hadoop to the workers. All you need to do is to generate and then copy the public key to the worker nodes to achieve passwordless access across your cluster.

1. Run ssh-keygen -t rsa (and enter empty values for the passphrase) on the master node. That will generate .ssh/id\_rsa and .ssh/id\_rsa.pub (private and public key). You now need to manually copy the .ssh/id\_rsa.pub and append it to ~/.ssh/authorized\_keys **on each worker.**

Keep in mind that this is a single-line public key and accidentally introducing a line break would cause a mismatch.

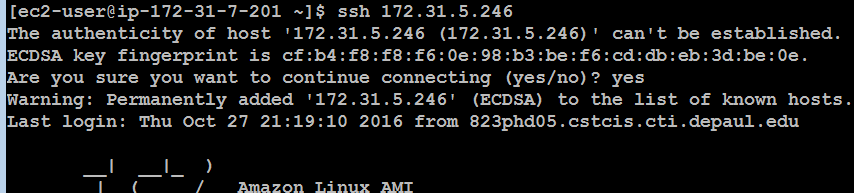
Note that the example below is NOT the master, but one of the workers (ip-172-31-5-246). The first public key is the .pem Amazon half and the 2nd public key is the master’s public key copied in as one line.



You can add the public key of the master to the master by running this command:

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

Make sure that you can ssh to all of the nodes from the master node (by running ssh 54.186.221.92, where the IP address is your worker node) from the master and ensuring that you were able to login. You can exit after successful ssh connection by typing exit (the command prompt will tell you which machine you are connected to, e.g., ec2-user@ip-172-31-37-113). Here’s me ssh-ing from master to worker.



Once you have verified that you can ssh from the master node to every cluster member including the master itself (ssh localhost), you are going to return to the master node (exit until your prompt shows the IP address of the master node) and pack the contents of the hadoop directory there. Make sure your Hadoop installation is configured correctly (because from now on, you will have 4 copies of the Hadoop directory and all changes need to be applied in 4 places).

cd (go to root home directory, i.e. /home/ec2-user/)

(pack up the entire Hadoop directory into a single file for transfer. You can optionally compress the file with gzip)

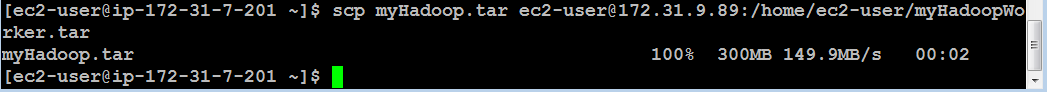
tar cvf myHadoop.tar hadoop-2.6.4

ls -al myHadoop.tar (to verify that the .tar file had been created)

Now, you need to copy the myHadoop.tar file to every non-master node in the cluster. If you had successfully setup public-private key access in the previous step, this command (for each worker node) will do that:

(copies the myHadoop.tar file from the current node to a remote node into a file called myHadoopWorker.tar. Don’t forget to replace the IP address with that your worker nodes. By the way, since you are on the Amazon EC2 network, either Public or Private IP will work just fine.)

scp myHadoop.tar ec2-user@54.187.63.189:/home/ec2-user/myHadoopWorker.tar



Once the tar file containing your Hadoop installation from master node has been copied to each worker node, you need to login to each non-master node and unpack the .tar file.

Run the following command (on each worker node, not on the master) to untar the hadoop file. We are purposely using a different tar archive name (i.e., myHadoopWorker.tar), so if you get “file not found” error, that means you are running this command on the master node or have not yet successfully copied myHadoopWorker.tar file to the worker.

tar xvf myHadoopWorker.tar

Once you are done, run this on the master (nothing needs to be done on the workers to format the cluster unless you are re-formatting, in which case you’ll need to delete the dfs directory).

hadoop namenode -format

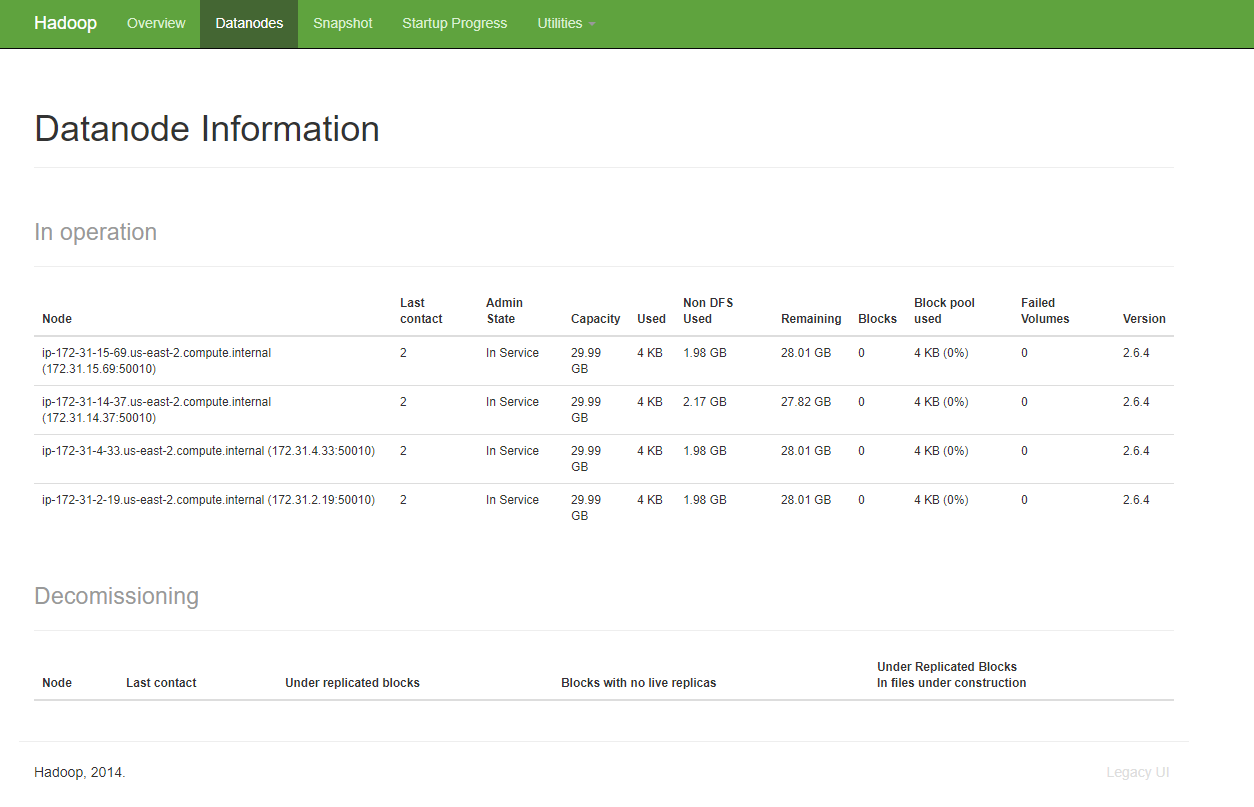
Once you have successfully completed the previous steps, you should can start and use your new cluster by going to the master node and running the start-dfs.sh and start-yarn.sh scripts (you do not need to explicitly start anything on worker nodes – the master will do that for you).

You should verify that the cluster is running by pointing your browser to the link below.

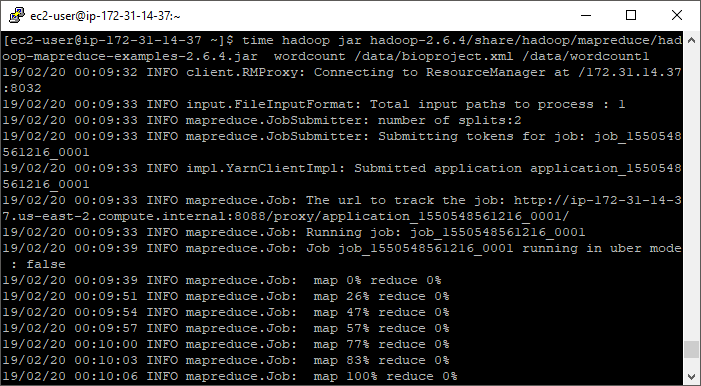
http://[insert-the-public-ip-of-master]:50070/

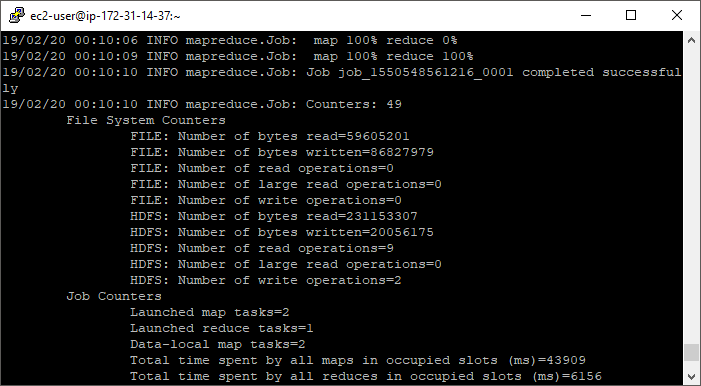
Make sure that the cluster is operational (you can see the 4 nodes under Datanodes tab).

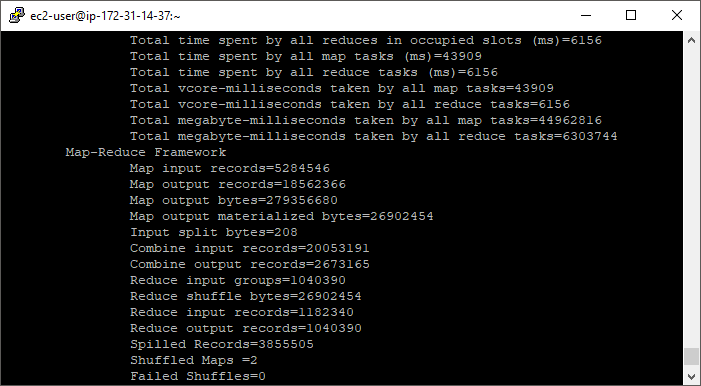
Submit a screenshot of your cluster status view.

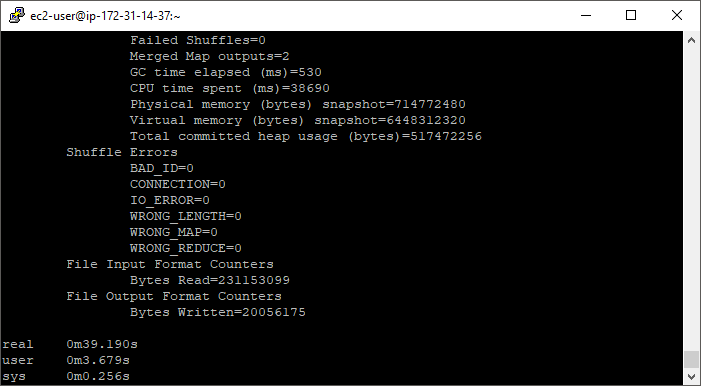


Repeat the steps for wordcount using bioproject.xml from Assignment 1 and submit screenshots of running it.









Submit a short paragraph with a discussion about how the results compare (faster? slower? How much faster/slower? Due to what?)

The single-node Hadoop instance from assignment 1 ran the word count job in 1 minute 10.229 seconds. The four-node instance ran the job 39.190 seconds which is roughly twice as fast. The speed increase is due to the additional nodes, however, I would have expected a more than doubling of speed, based just on the number of nodes. It’s likely that network speed and block distribution have impacted speed.

Running the following command shows that the file was only split into two blocks and where they are located.

hdfs fsck /data/bioproject.xml -files -blocks -locations

Connecting to namenode via http://ip-172-31-14-37.us-east-2.compute.internal:50070

FSCK started by ec2-user (auth:SIMPLE) from /172.31.14.37 for path /data/bioproject.xml at Wed Feb 20 00:30:53 UTC 2019

/data/bioproject.xml 231149003 bytes, 2 block(s): OK

0. BP-806142183-172.31.14.37-1550548147742:blk\_1073741825\_1001 len=134217728 repl=2 [172.31.14.37:50010, 172.31.4.33:50010]

1. BP-806142183-172.31.14.37-1550548147742:blk\_1073741826\_1002 len=96931275 repl=2 [172.31.14.37:50010, 172.31.4.33:50010]

Status: HEALTHY

Total size: 231149003 B

Total dirs: 0

Total files: 1

Total symlinks: 0

**Total blocks (validated): 2 (avg. block size 115574501 B)**

Minimally replicated blocks: 2 (100.0 %)

Over-replicated blocks: 0 (0.0 %)

Under-replicated blocks: 0 (0.0 %)

Mis-replicated blocks: 0 (0.0 %)

Default replication factor: 2

Average block replication: 2.0

Corrupt blocks: 0

Missing replicas: 0 (0.0 %)

Number of data-nodes: 4

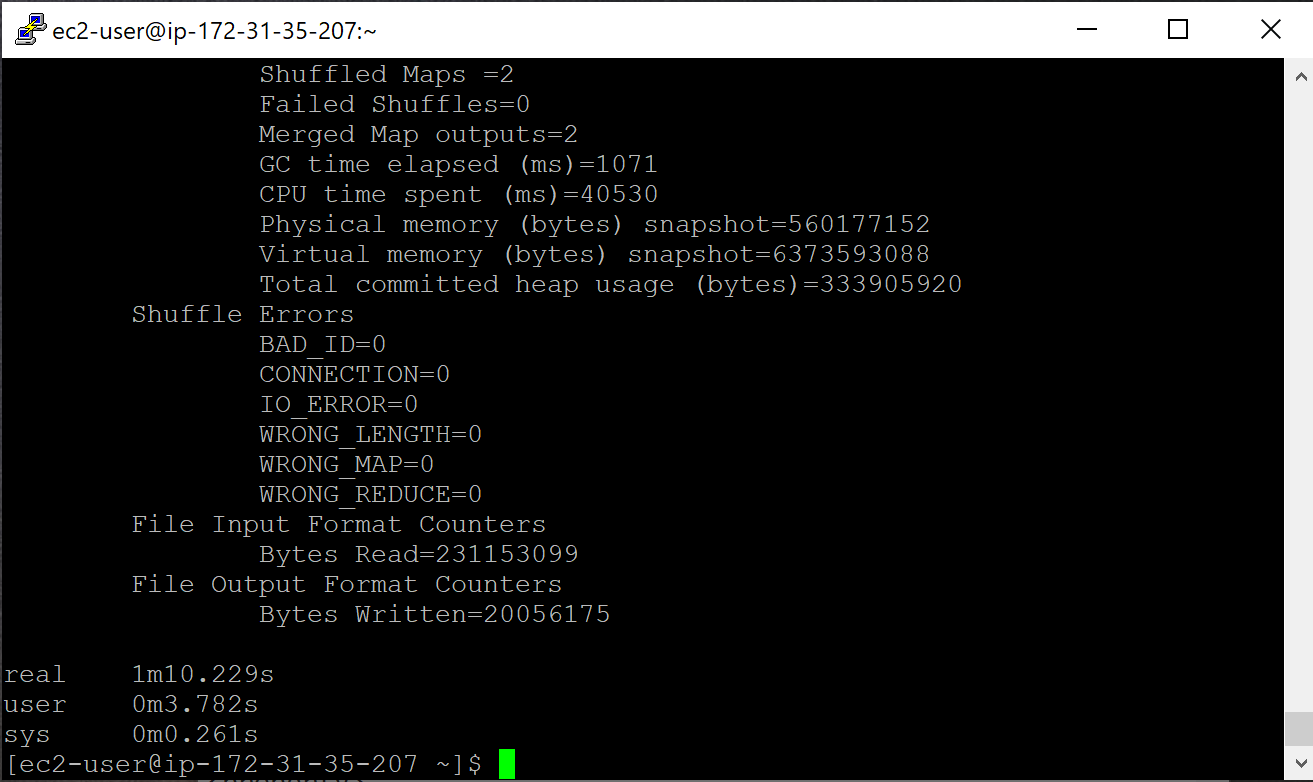
Number of racks: 1

FSCK ended at Wed Feb 20 00:30:53 UTC 2019 in 1 milliseconds

The filesystem under path '/data/bioproject.xml' is HEALTHY

This would explain the half-time only gain in speed.

Here are the times from assignment 1:



# Part 2: Hive

Run the following three (1.2, 1.3 and 2.1) queries in Hive and record the time they take to execute: <http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/SSBM1/SSBM_queries.sql>

Perform the following transform operation using SELECT TRANSFORM on the customer table by creating a new table:

For the c\_address column, shorten it to 8 characters (i.e., if the value is longer, remove extra characters, but otherwise keep it as-is). For c\_city, add a space and a # to indicate the digit at the end (e.g., UNITED KI2 => UNITED KI #2, or INDONESIA4 => INDONESIA #4). Make sure to modify the columns of the target table accordingly (since you are introducing longer columns).

# Part 3: Pig

Convert and load the data into Pig, implementing only queries 0.1, 0.2, 0.3. Do not implement all queries.

Check disk storage space in HDFS, if your disk usage is over 90% Pig may hang without an error or a warning.

One easy way to time Pig is as follows: put your sequence of pig commands into a text file and then run, from command line in pig directory (e.g., [ec2-user@ip-172-31-6-39 pig-0.15.0]$), bin/pig -f pig\_script.pig (which will inform you how long the pig script took to run).

# Part 4: Hadoop Streaming

Implement queries **0.2 and 0.3** using Hadoop streaming with python.

NOTE: You may implement this part in Java if you prefer.

Submit a single document containing your written answers. Be sure that this document contains your name and “CSC 555 Project Phase 1” at the top.