CSC 555 Mining Big Data

Assignment 3

Due Wednesday 10/2

1) Consider a Hadoop job that processes an input data file of size equal to 30 disk blocks (30 different blocks, you can assume that HDFS replication factor is set to 1). The mapper in this job requires 2 minutes to read and fully process a single block of data. Reducer requires 2 seconds (not minutes) to produce an answer for a one key and there are a total of 1000 **distinct** keys (mappers generate a lot of key-value pairs, but keys only occur in 1-1000 range).

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

   b) 10 Hadoop worker nodes?

   c) 50 Hadoop worker nodes?

   d) Would the introduction of the combiner affect the runtime of this job?

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

2)

   a) Suppose you have a 7-node cluster with replication factor of 2. Describe what HDFS has to do after it determines that a node has crashed. For simplicity, you can assume that the failed node is not replaced and your cluster is reduced to 6 nodes. Do not worry about what MapReduce framework has to do, just HDFS (i.e. from the storage perspective only).

   b) If a mapper task fails due to a crash of the current node, which other node(s) is the best place to re-run the same task? Why?

3) Recall that compression ratio is defined as the ratio between "compressed data size"/"original data size".
   a) Between the string 'AAABBBAAABBBAAABBB' and string 'XYZDFBSADEGBBEFDSFDS' which one would compress better? Why?

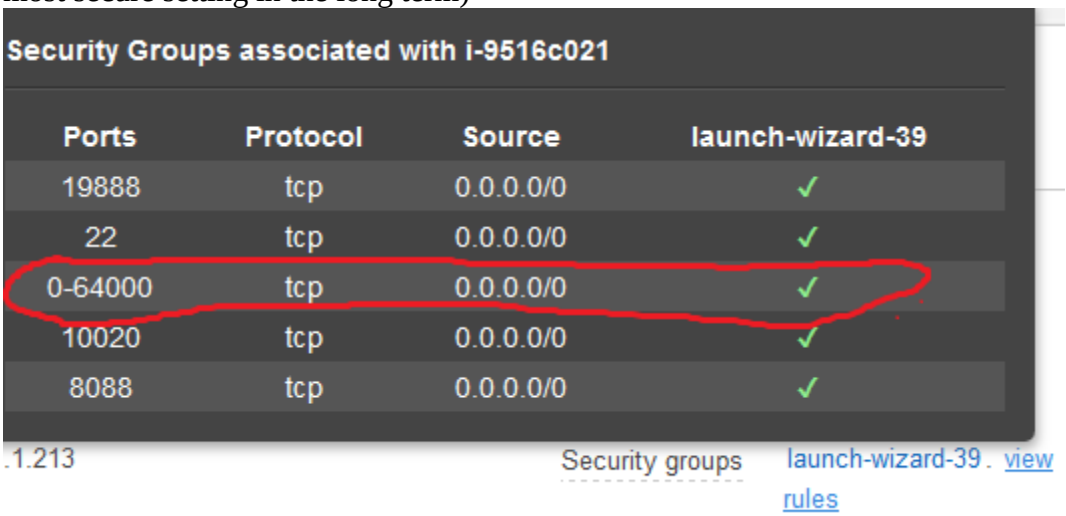
   b) Between 'ABCDABCDABCD' and 'X' which one would compress better? Why?

   c) Between a sequence of numbers "2 5 4 7 3 6 2 3 5" and "1 2 3 4 5 6 7 8 9" which one would compress better? Why?

4) Our next step is to setup a multi-node cluster and run a simple wordcount. For this assignment, we will create a 3-node cluster (with a total of 1 master + 2 worker nodes). Include your master node in the "slaves" file, to make sure all 3 nodes are working. You need to perform the following steps:
    1. Create a new node of at least medium size. It is possible, but I do not recommend trying to reconfigure existing node into a new cluster for several reasons.

        a. When creating a node I recommend changing the default 8G hard drive to at least 32G so that you do not run out of space easily.

        b. Change your security group setting to open firewall access. Rather than figure out all individual port, you can set 0-64000 range opening up all ports (not the most secure setting in the long term)



        c. Finally, right click on the Master node and choose "create more like this" to create 2 more nodes with same settings. If you configure the master first, all of the settings (drive size, security group, etc.) will be automatically copied to new instances.

    2. 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.

a. Configure core-site.xml, adding the PrivateIP (Private NOT public) of the master.

```
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>
<name>fs.defaultFS</name>
<value>hdfs://172.31.7.201/</value>
</property>

</configuration>
[ec2-user@ip-172-31-7-201 ~]$ cat hadoop-2.6.4/etc/hadoop/core-site.xml
```

b. Configure hdfs-site and set replication factor to 2.

```
<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>
<name>dfs.replication</name>
<value>2</value>
</property>

</configuration>
[ec2-user@ip-172-31-9-105 ~]$
```

c. 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

```
<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>

</configuration>
[ec2-user@ip-172-31-9-105 ~]$ cat hadoop-2.6.4/etc/hadoop/mapred-site.xml
```

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

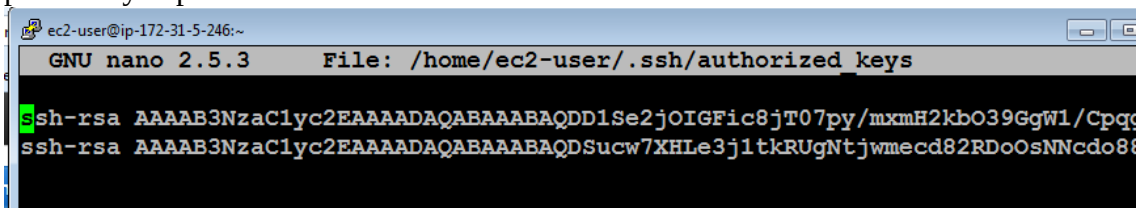
```
<!-- Site specific YARN configuration properties -->

<property>
<name>yarn.resourcemanager.hostname</name>
<value>172.31.7.201</value>
</property>

<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>

</configuration>
[ec2-user@ip-172-31-7-201 ~]$ cat hadoop-2.6.4/etc/hadoop/yarn-site.xml
```

Finally, edit the slaves file and list your 3 nodes (master and 2 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
172.31.11.50

Make sure that you use private IP, not the public IPs 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. Not on the master.**
   Keep in mind that this is a single-line public key and accidentally introducing a line break would break it. For example:
   Note that this 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.

```
[ec2-user@ip-172-31-7-201 ~]$ ssh 172.31.5.246
The authenticity of host '172.31.5.246 (172.31.5.246)' can't be established.
ECDSA key fingerprint is cf:b4:f8:f8:f6:0e:98:b3:be:f6:cd:db:eb:3d:be:0e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.31.5.246' (ECDSA) to the list of known hosts.
Last login: Thu Oct 27 21:19:10 2016 from 823phd05.cstcis.cti.depaul.edu

       __|  __|_  )
         |  (     /   Amazon Linux AMI
```

Once you are 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 up the contents of the hadoop directory there. Make sure your Hadoop installation is configured correctly (because from now on, you will have 3 copies of the Hadoop directory and all changes may need to be applied in 3 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 <u>each</u> 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

```
[ec2-user@ip-172-31-7-201 ~]$ scp myHadoop.tar ec2-user@172.31.9.89:/home/ec2-user/myHadoopWo
rker.tar
myHadoop.tar                                  100%  300MB 149.9MB/s   00:02
[ec2-user@ip-172-31-7-201 ~]$
```

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 successfully copied myHadoopWorker.tar file.

tar xvf myHadoopWorker.tar

Once you are done, run this on the master:
hadoop namenode -format

Once you have successfully completed the previous steps, you should be able to 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 3 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.

5) Finally, we will practice using HBase. Note that HBase runs on top of HDFS, bypassing MapReduce (so only NameNode and DataNode need to be running).

cd
(Download HBase)
wget http://rasinsrv07.cstcis.cti.depaul.edu/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', '222-333-445'
put 'employees', 'ID3', 'private:address', '123 Fake St.'
put 'employees', 'ID1', 'private:address', '243 N. Wabash Av.'

scan 'employees'

Now that we have filled in a few values, add at least 2 columns with at least 3 new values total to the "public" column family (e.g., position, officeNumber). Verify that the table has been filled in properly with scan command and submit a screenshot.

6) This section is entirely optional and gives you an opportunity to make-up Project Phase1 part-2 and part-3 if you were not able to get them to run. If you have completed your Project Phase 1 submission, you do not need to do anything here.

   a) Run the Pig queries as stated in Project Phase 1. In addition to that, please also run Query 1.3

   b) Run the hadoop streaming exercise from Project Phase 1. You do not need to do anything extra, but you will need to use SSBM Scale4 instead of Scale1 (24M rows instead of 6M rows):
   Located here: http://rasinsrv07.cstcis.cti.depaul.edu/CSC553/data/
   (there is also a Scale14 version, but you do not need to use it)

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