Tao Hu

Professor: Jun Li

CS 381

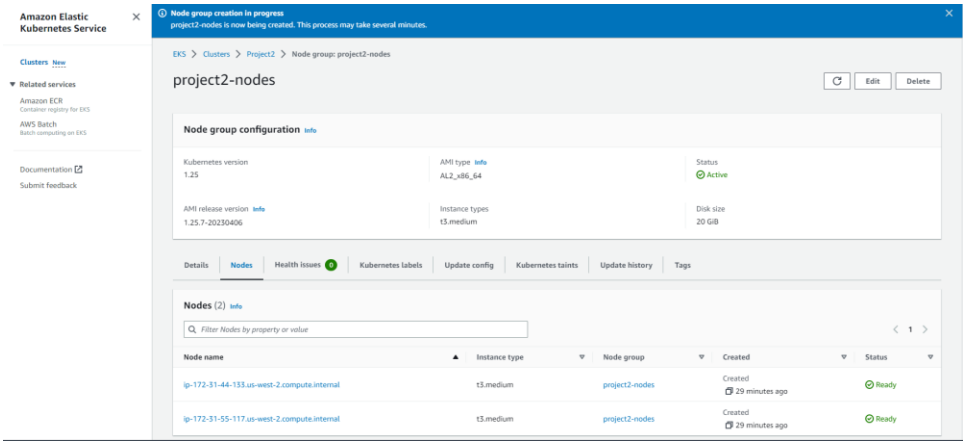
4/5/2023

Project 2

Task 1: Launch a cluster of virtual machines in a cloud environment (e.g., AWS, Azure, or GCP). You will need to have one node as the master and at least two nodes as workers (slaves).

Full steps see project 1 we did before, I also used EMR to setup cluster, deploy by AWS:

Ref:Cloud-Computing/CS381 Cloud Computing Project 1.pdf at main · Talen-520/Cloud-Computing (github.com)



Task 2: Deploy the HDFS service on the cluster.

Setup HDFS on cluster with AWS EMR is way easier than manual installation, task 5 will also include this step, here is manual guide I completed.

Here is full installation guide:

Install and Configure Apache Hadoop on Ubuntu 20.04 - Vultr.com

The trouble I met is connection refused, here is solution

linux - connect to host localhost port 22: Connection refused - Stack Overflow

Make sure run command below before ssh in local

```
sudo service ssh start
```

screenshots below is my process:

```
hadoop@DESKTOP-IBM4J8C:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Created directory '/home/hadoop/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:rxSoupuAwJ5BXf0+Q3+DCvcq9bHe3dC6UaB3j5iouSI hadoop@DESKTOP-IBM4J8C
The key's randomart image is:
+---[RSA 3072]----+
|     ..          |
|  . .  .         |
| . .    .     .  |
|o      . o   . . |
|.o    . S . o . o|
|+ o  . . O = * =.|
|oo  .   = O * = o|
| . oE .o = +. .+.|
|  =o . .*oo. .oo.|
+----[SHA256]-----+
hadoop@DESKTOP-IBM4J8C:~$ sudo cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
hadoop@DESKTOP-IBM4J8C:~$ sudo chmod 640 ~/.ssh/authorized_keys
hadoop@DESKTOP-IBM4J8C:~$ ssh localhost
ssh: connect to host localhost port 22: Connection refused
hadoop@DESKTOP-IBM4J8C:~$ ssh -vvv localhost
OpenSSH_8.2p1 Ubuntu-4ubuntu0.5, OpenSSL 1.1.1f  31 Mar 2020
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: /etc/ssh/ssh_config line 19: include /etc/ssh/ssh_config.d/*.conf matched no files
debug1: /etc/ssh/ssh_config line 21: Applying options for *
debug2: resolving "localhost" port 22
debug2: ssh_connect_direct
debug1: Connecting to localhost [127.0.0.1] port 22.
debug1: connect to address 127.0.0.1 port 22: Connection refused
ssh: connect to host localhost port 22: Connection refused
hadoop@DESKTOP-IBM4J8C:~$ service sshd restart
sshd: unrecognized service
hadoop@DESKTOP-IBM4J8C:~$ sudo service ssh start
 * Starting OpenBSD Secure Shell server sshd                                                    [ OK ]
hadoop@DESKTOP-IBM4J8C:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:oH7Zjwknn6keqOBvL/O/Gld6JibT0oLTpnsk/iottO4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
ssh_dispatch_run_fatal: Connection to 127.0.0.1 port 22: Broken pipe
```

```
hadoop@DESKTOP-IBM4J8C:~$ sudo mv hadoop-3.3.5 /usr/local/hadoop
hadoop@DESKTOP-IBM4J8C:~$ sudo mkdir /usr/local/hadoop/logs
hadoop@DESKTOP-IBM4J8C:~$ sudo chown -R hadoop:hadoop /usr/local/hadoop
hadoop@DESKTOP-IBM4J8C:~$ sudo nano ~/.bashrc
hadoop@DESKTOP-IBM4J8C:~$ sudo nano ~/.bashrc


Use "fg" to return to nano.

[1]+  Stopped                 sudo nano ~/.bashrc
hadoop@DESKTOP-IBM4J8C:~$ source ~/.bashrc
hadoop@DESKTOP-IBM4J8C:~$ which javac
/usr/bin/javac
hadoop@DESKTOP-IBM4J8C:~$ readlink -f /usr/bin/javac
/usr/lib/jvm/java-8-openjdk-amd64/bin/javac
hadoop@DESKTOP-IBM4J8C:~$ sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
hadoop@DESKTOP-IBM4J8C:~$ cd /usr/local/hadoop/lib
hadoop@DESKTOP-IBM4J8C:/usr/local/hadoop/lib$ sudo wget https://jcenter.bintray.com/javax/activation/javax.activation-api/1.2.0/javax.activation-api-1.2.0.jar
--2023-04-17 05:58:55--  https://jcenter.bintray.com/javax/activation/javax.activation-api/1.2.0/javax.activation-api-1.2.0.jar
Resolving jcenter.bintray.com (jcenter.bintray.com)... 34.95.74.180
Connecting to jcenter.bintray.com (jcenter.bintray.com)|34.95.74.180|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 56674 (55K) [application/java-archive]
Saving to: 'javax.activation-api-1.2.0.jar'

javax.activation-api-1.2.0.ja 100%[=============================================>]  55.35K  --.-KB/s    in 0.02s

2023-04-17 05:58:55 (2.84 MB/s) - 'javax.activation-api-1.2.0.jar' saved [56674/56674]

hadoop@DESKTOP-IBM4J8C:/usr/local/hadoop/lib$ hadoop version
Hadoop 3.3.5
Source code repository https://github.com/apache/hadoop.git -r 706d88266abcee09ed78fbaa0ad5f74d818ab0e9
Compiled by stevel on 2023-03-15T15:56Z
Compiled with protoc 3.7.1
From source with checksum 6bbd9afcf4838a0eb12a5f189e9bd7
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-3.3.5.jar
hadoop@DESKTOP-IBM4J8C:/usr/local/hadoop/lib$ sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml
hadoop@DESKTOP-IBM4J8C:/usr/local/hadoop/lib$ sudo mkdir -p /home/hadoop/hdfs/{namenode,datanode}
hadoop@DESKTOP-IBM4J8C:/usr/local/hadoop/lib$ sudo chown -R hadoop:hadoop /home/hadoop/hdfs
hadoop@DESKTOP-IBM4J8C:/usr/local/hadoop/lib$ sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
hadoop@DESKTOP-IBM4J8C:/usr/local/hadoop/lib$ sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml


Use "fg" to return to nano.

[2]+  Stopped                 sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
hadoop@DESKTOP-IBM4J8C:/usr/local/hadoop/lib$ sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
hadoop@DESKTOP-IBM4J8C:/usr/local/hadoop/lib$ sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
hadoop@DESKTOP-IBM4J8C:/usr/local/hadoop/lib$ sudo nano $HADOOP_HOME/etc/hadoop/yarn-site.xml
hadoop@DESKTOP-IBM4J8C:/usr/local/hadoop/lib$ sudo su - hadoop
hadoop@DESKTOP-IBM4J8C:~$ hdfs namenode -format
```

Task 3: Download the text version of Pride and Prejudice from Project Gutenberg, and save it to the HDFS cluster.

Download:

wget https://www.gutenberg.org/ebooks/1342.txt.utf-8

First time we need create a directory by command:

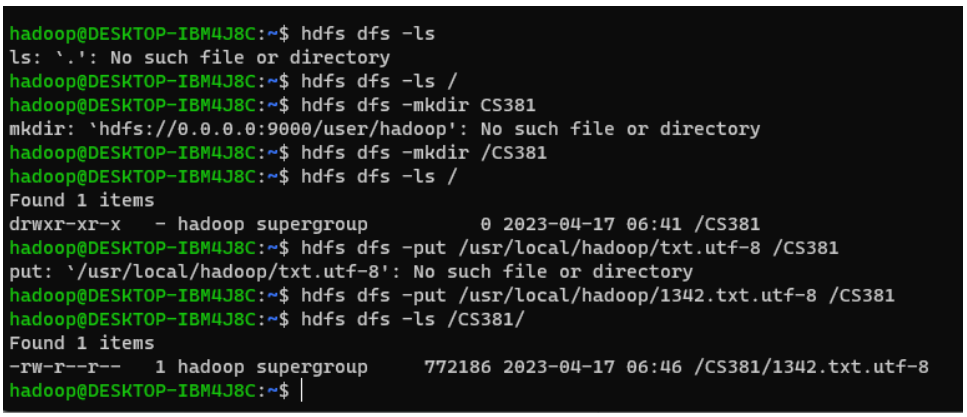
hdfs dfs –mkdir [folder name]

Then we use:

hdfs dfs –put /user/local/hadoop/1342.txt.utf-8 /CS381

```
hdfs dfs -put /path/to/local/file /path/to/hdfs/directory
```

to Copy the text file from local machine to HDFS



Task 4: Deploy the Spark service on the cluster.

Task 5: Use the file in HDFS as input, run a wordcount program in Spark to count the number of occurrences of each word. Sort the words by count, in descending order, and return a list of the (word, count) pairs for the 20 most used words.

I completed task 4 and 5 together, here is steps

Via EMR

Setup S3 bucket to store input file locations, we can save output right there as well

Ref: S3 Management Console (amazon.com)

Created cluster with Hadoop, spark and yarn, setup key pair and configure security group for master with SSH and port 22, bound my IP with it.

Then hit following command (for Linux/MacOS):

ssh -i C:\Users\Owner\Downloads\test1.pem hadoop@ec2-35-153-83-126.compute-1.amazonaws.com

Once done create and run python program with code [Github]:

Command to create file:

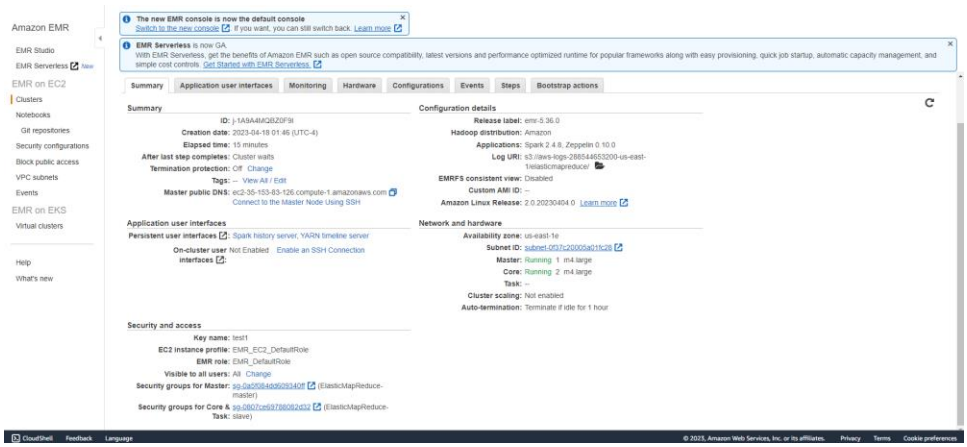
via main.py

Then copy code into file, hit esc key then type :wq then hit enter exit

Run program with:

spark-submit main.py

EMR UI:



Output:

Cost 17 sec to get result below

the 4509

to 4275

of 3897

and 3443

a 2021

in 1923

her 1905

was 1817

I 1764

that 1458

not 1432

she 1341

be 1227

his 1196

as 1165

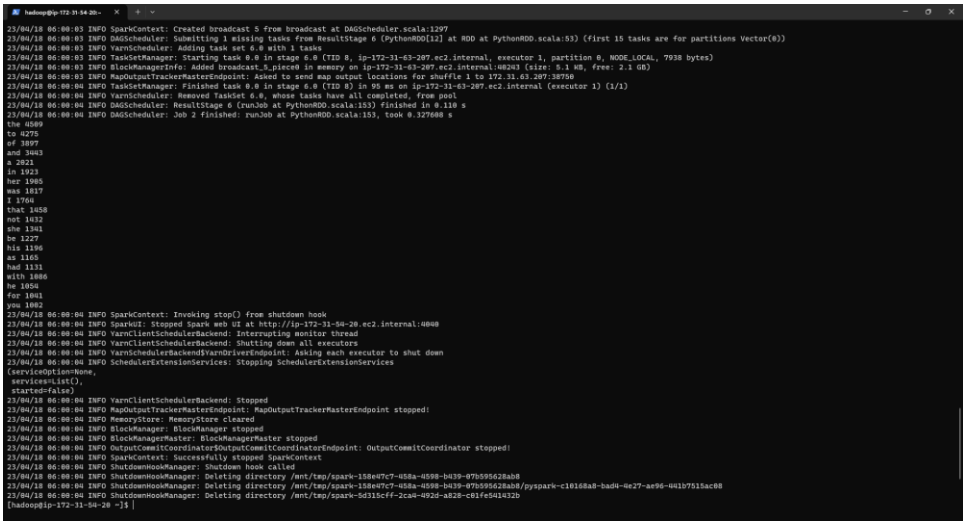
had 1131

with 1086

he 1054

for 1041

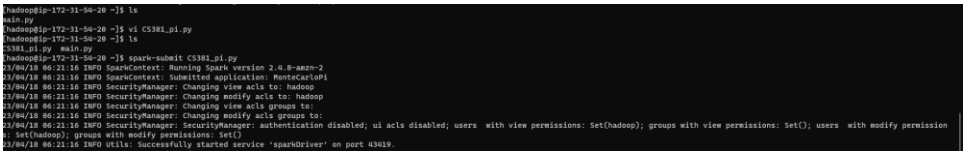
you 1002



Task 6: Write a Spark program that uses Monte Carlo methods to estimate the value of $π$.

Do the same steps from Task 5 solution

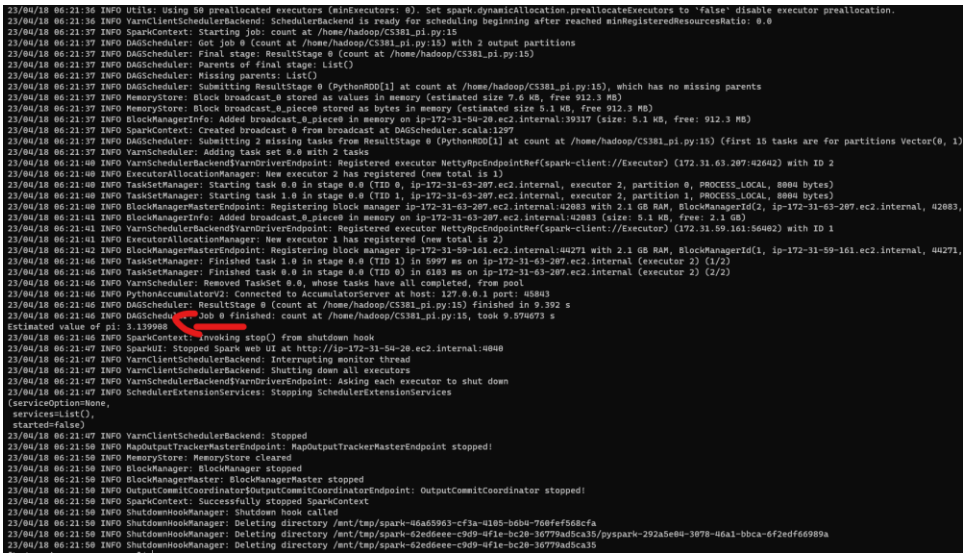
Code [Github]



Output:

This process took me 34 sec to get result

Estimated value of pi: 3.139908



```
23/04/18 06:21:36 INFO Utils: Using 50 preallocated executors (minExecutors: 0). Set spark.dynamicAllocation.preallocateExecutors to 'false' disable executor preallocation.
23/04/18 06:21:36 INFO YarnClientSchedulerBackend: SchedulerBackend is ready for scheduling beginning after reached minRegisteredResourcesRatio: 0.0
23/04/18 06:21:37 INFO SparkContext: Starting job: count at /home/hadoop/CS381_pi.py:15
23/04/18 06:21:37 INFO DAGScheduler: Got job 0 (count at /home/hadoop/CS381_pi.py:15) with 2 output partitions
23/04/18 06:21:37 INFO DAGScheduler: Final stage: ResultStage 0 (count at /home/hadoop/CS381_pi.py:15)
23/04/18 06:21:37 INFO DAGScheduler: Parents of final stage: List()
23/04/18 06:21:37 INFO DAGScheduler: Missing parents: List()
23/04/18 06:21:37 INFO DAGScheduler: Submitting ResultStage 0 (PythonRDD[1] at count at /home/hadoop/CS381_pi.py:15), which has no missing parents
23/04/18 06:21:37 INFO MemoryStore: Block broadcast_0 stored as values in memory (estimated size 7.6 KB, free 912.3 MB)
23/04/18 06:21:37 INFO MemoryStore: Block broadcast_0_piece0 stored as bytes in memory (estimated size 5.1 KB, free 912.3 MB)
23/04/18 06:21:37 INFO BlockManagerInfo: Added broadcast_0_piece0 in memory on ip-172-31-54-20.ec2.internal:39317 (size: 5.1 KB, free: 912.3 MB)
23/04/18 06:21:37 INFO SparkContext: Created broadcast 0 from broadcast at DAGScheduler.scala:1297
23/04/18 06:21:37 INFO DAGScheduler: Submitting 2 missing tasks from ResultStage 0 (PythonRDD[1] at count at /home/hadoop/CS381_pi.py:15) (first 15 tasks are for partitions Vector(0, 1)
23/04/18 06:21:37 INFO YarnScheduler: Adding task set 0.0 with 2 tasks
23/04/18 06:21:40 INFO YarnSchedulerBackend$YarnDriverEndpoint: Registered executor NettyRpcEndpointRef(spark-client://Executor) (172.31.63.207:42642) with ID 2
23/04/18 06:21:40 INFO ExecutorAllocationManager: New executor 2 has registered (new total is 1)
23/04/18 06:21:40 INFO TaskSetManager: Starting task 0.0 in stage 0.0 (TID 0, ip-172-31-63-207.ec2.internal, executor 2, partition 0, PROCESS_LOCAL, 8004 bytes)
23/04/18 06:21:40 INFO TaskSetManager: Starting task 1.0 in stage 0.0 (TID 1, ip-172-31-63-207.ec2.internal, executor 2, partition 1, PROCESS_LOCAL, 8004 bytes)
23/04/18 06:21:40 INFO BlockManagerMasterEndpoint: Registering block manager ip-172-31-63-207.ec2.internal:42083 with 2.1 GB RAM, BlockManagerId(2, ip-172-31-63-207.ec2.internal, 42083,
23/04/18 06:21:41 INFO BlockManagerInfo: Added broadcast_0_piece0 in memory on ip-172-31-63-207.ec2.internal:42083 (size: 5.1 KB, free: 2.1 GB)
23/04/18 06:21:41 INFO YarnSchedulerBackend$YarnDriverEndpoint: Registered executor NettyRpcEndpointRef(spark-client://Executor) (172.31.59.161:56402) with ID 1
23/04/18 06:21:41 INFO ExecutorAllocationManager: New executor 1 has registered (new total is 2)
23/04/18 06:21:42 INFO BlockManagerMasterEndpoint: Registering block manager ip-172-31-59-161.ec2.internal:44271 with 2.1 GB RAM, BlockManagerId(1, ip-172-31-59-161.ec2.internal, 44271,
23/04/18 06:21:46 INFO TaskSetManager: Finished task 1.0 in stage 0.0 (TID 1) in 5997 ms on ip-172-31-63-207.ec2.internal (executor 2) (1/2)
23/04/18 06:21:46 INFO TaskSetManager: Finished task 0.0 in stage 0.0 (TID 0) in 6103 ms on ip-172-31-63-207.ec2.internal (executor 2) (2/2)
23/04/18 06:21:46 INFO YarnScheduler: Removed TaskSet 0.0, whose tasks have all completed, from pool
23/04/18 06:21:46 INFO PythonAccumulatorV2: Connected to AccumulatorServer at host: 127.0.0.1 port: 45843
23/04/18 06:21:46 INFO DAGScheduler: ResultStage 0 (count at /home/hadoop/CS381_pi.py:15) finished in 9.392 s
23/04/18 06:21:46 INFO DAGScheduler: Job 0 finished: count at /home/hadoop/CS381_pi.py:15, took 9.574673 s
Estimated value of pi: 3.139908
23/04/18 06:21:46 INFO SparkContext: Invoking stop() from shutdown hook
23/04/18 06:21:47 INFO SparkUI: Stopped Spark web UI at http://ip-172-31-54-20.ec2.internal:4040
23/04/18 06:21:47 INFO YarnClientSchedulerBackend: Interrupting monitor thread
23/04/18 06:21:47 INFO YarnClientSchedulerBackend: Shutting down all executors
23/04/18 06:21:47 INFO YarnSchedulerBackend$YarnDriverEndpoint: Asking each executor to shut down
23/04/18 06:21:47 INFO SchedulerExtensionServices: Stopping SchedulerExtensionServices
(serviceOption=None,
 services=List(),
 started=false)
23/04/18 06:21:47 INFO YarnClientSchedulerBackend: Stopped
23/04/18 06:21:50 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
23/04/18 06:21:50 INFO MemoryStore: MemoryStore cleared
23/04/18 06:21:50 INFO BlockManager: BlockManager stopped
23/04/18 06:21:50 INFO BlockManagerMaster: BlockManagerMaster stopped
23/04/18 06:21:50 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
23/04/18 06:21:50 INFO SparkContext: Successfully stopped SparkContext
23/04/18 06:21:50 INFO ShutdownHookManager: Shutdown hook called
23/04/18 06:21:50 INFO ShutdownHookManager: Deleting directory /mnt/tmp/spark-46a65963-cf3a-4105-b6b4-760fef568cfa
23/04/18 06:21:50 INFO ShutdownHookManager: Deleting directory /mnt/tmp/spark-62ed6eee-c9d9-4f1e-bc20-36779ad5ca35/pyspark-292a5e84-3078-46a1-bbca-6f2edf66989a
23/04/18 06:21:50 INFO ShutdownHookManager: Deleting directory /mnt/tmp/spark-62ed6eee-c9d9-4f1e-bc20-36779ad5ca35
```