Project 2: AWS EMR (Elastic Map Reduce) Zeehan Rahman

Step 1: Creating a EMR cluster with Spark Installed (Task 1, Task 2 and Task 4)

Before creating an EMR cluster, certain requirements needed to be met. These were:

Creating a S3 bucket

Input: aws s3 mb s3://zee786bucket

Output: make_bucket: zee786bucket

Consequently, I also uploaded the gutenberg text file to the s3 bucket using the following command:

```
aws s3 cp C:\Users\zeeha\Desktop\Documents\Courses\CSCI381-C\Projects\Project2\gutenberg.txt
s3://zee786bucket/
```

Creating a VPC: I tried creating a cluster with ec2 instance type m4.large. However, shortly after starting of the cluster, the emr cluster kept getting terminated, with the error message:

"Terminated with errorsThe VPC/subnet configuration was invalid: Subnet is required : The specified instance type m4.large can only be used in a VPC.".

Therefore, I created a VPC with a public subnet and recorded its subnet id. I used the management console as I did not find a way to achieve it via aws cli. I followed the steps from the link: https://docs.aws.amazon.com/eks/latest/userguide/creating-a-vpc.html

Finally, I used the following commands to create a cluster:

Input:

```
PS C:\Users\zeeha\Downloads> aws emr create-cluster --name HelloC3 --release-label emr-5.35.0
--applications Name=Spark --use-default-roles --ec2-attributes
KeyName=JWS,SubnetId=subnet-0812e94037abeb8ce --instance-type m4.large --instance-count 3
```

When specifying the instance count 3, it is implicitly taken as 1 master node and 2 core (worker, data) nodes

Output:

```
{
```

```
"ClusterId": "j-294LHFI7TT7K9",

"ClusterArn": "arn:aws:elasticmapreduce:us-east-1:632975414937:cluster/j-294LHFI7TT7K9"

}
```

Step 2: SSH into the master node

After recording the cluster id of the cluster, I attempted to SSH into the master node (namenode) of the cluster, using the following command:

```
aws emr ssh --cluster-id j-294LHFI7TT7K9 --key-pair-file C:\Users\zeeha\Downloads\JWS.pem
```

However, I was unsuccessful in connecting to the master and got the following error:

```
ssh -o StrictHostKeyChecking=no -o ServerAliveInterval=10 -i C:\Users\zeeha\Downloads\JWS.pem hadoop@ec2-18-215-231-244.compute-1.amazonaws.com -t

ssh: connect to host ec2-18-215-231-244.compute-1.amazonaws.com port 22: Connection timed out
```

After some research, I found out that I have to allow inbound traffic to the master node from my ip address. I used the management console as I did not find a way to achieve it via aws cli. I followed the simple steps from the link: https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-connect-ssh-prereqs.html

After that I was successfully able to login to my emr cluster. This time by mistake, I used the wrong command:

```
PS C:\Users\zeeha\Downloads> ssh -i ~/C:/Users/zeeha/Downloads/JWS.pem hadoop@ec2-18-215-231-244.compute-1.amazonaws.com
```

```
Output: Warning: Identity file C:\Users\zeeha/C:/Users/zeeha/Downloads/JWS.pem not accessible: No such file or directory.

The authenticity of host 'ec2-18-215-231-244.compute-1.amazonaws.com (18.215.231.244)' can't be established.

ECDSA key fingerprint is SHA256:d111An+rEWcciAPQ8gXXbckjAxCqtFl0MkGwf28PIf4.

Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

Warning: Permanently added 'ec2-18-215-231-244.compute-1.amazonaws.com,18.215.231.244' (ECDSA) to the list of known hosts.
```

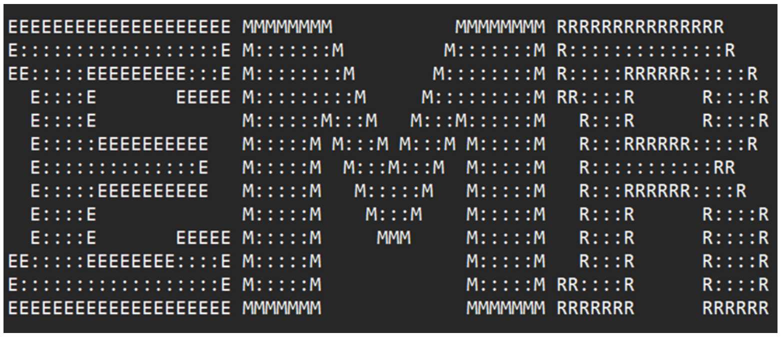
```
hadoop@ec2-18-215-231-244.compute-1.amazonaws.com: Permission denied
(publickey,gssapi-keyex,gssapi-with-mic).
```

Then, I again tried to ssh into my cluster, this time using the correct command for windows, and I was successfully able to connect to the master node

Input: PS C:\Users\zeeha\Downloads> aws emr ssh --cluster-id j-294LHFI7TT7K9 --key-pair-file C:\Users\zeeha\Downloads\JWS.pem

Output:

```
ssh -o StrictHostKeyChecking=no -o ServerAliveInterval=10 -i C:\Users\zeeha\Downloads\JWS.pem
hadoop@ec2-18-215-231-244.compute-1.amazonaws.com -t

Last login: Sat Apr 16 17:21:39 2022

__|  __|_  )

_|  (     /   Amazon Linux 2 AMI

___|\___|___|

https://aws.amazon.com/amazon-linux-2/

17 package(s) needed for security, out of 26 available

Run "sudo yum update" to apply all updates.
```



```
EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR
```

(Directly copying the above print "EMR" was not coming out perfectly aligned, therefore paste a snip)

Step 3: Load the text file in the HDFS cluster (Task 3)

After connecting to the namenode, I copied the Gutenberg text file from the s3 bucket zee786bucket to the local file system (master machine) using the following command:

(I had already loaded the text file in s3 bucket in Step 1 Line 7)

```
[hadoop@ip-192-168-84-168 ~]$ aws s3 cp  s3://zee786bucket/gutenberg.txt .
```

Output:

```
download: s3://zee786bucket/gutenberg.txt to ./gutenberg.txt
```

I verified if the file successfully transferred:

```
[hadoop@ip-192-168-84-168 ~]$ ls

gutenberg.txt
```

Consequently, I created a directory in the hdfs cluster to store the text file:

```
[hadoop@ip-192-168-84-168 ~]$ hadoop fs -mkdir /t5
```

Then, I used the following command, to finally load the text file to the hdfs cluster:

```
[hadoop@ip-192-168-84-168 ~]$ hadoop fs -put gutenberg.txt /t5
```

Made sure if the file successfully got loaded:

```
[hadoop@ip-192-168-84-168 ~]$ hadoop fs -ls /t5

Found 1 items

-rw-r--r--   1 hadoop hdfsadmingroup     798765 2022-04-16 18:01 /t5/gutenberg.txt
```

Step 4: Word Count Using Spark Shell (Task 5)

```
[hadoop@ip-192-168-84-168 ~]$ spark-shell

Setting default log level to "WARN".
```

```
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).

22/04/16 18:09:49 WARN Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to
uploading libraries under SPARK_HOME.

Spark context Web UI available at http://ip-192-168-84-168.ec2.internal:4040

Spark context available as 'sc' (master = yarn, app id = application_1650127996462_0002).

Spark session available as 'spark'.

Welcome to
```

```
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.8-amzn-1
      /_/
```

Directly copying the output makes the text illegible, therefore I paste a snip:

```
Using Scala version 2.11.12 (OpenJDK 64-Bit Server VM, Java 1.8.0_322)

Type in expressions to have them evaluated.

Type :help for more information.
```

And here is the code in scala using spark shell:

```
scala> val mydata = sc.textFile("/t5/gutenberg.txt")      //Creating an RDD
mydata: org.apache.spark.rdd.RDD[String] = /t5/gutenberg.txt MapPartitionsRDD[1] at textFile at
<console>:24
scala> val counts = mydata.flatMap(line => line.toLowerCase().replace(".", " ").replace(",", "
").split(" ")).map(word => (word, 1L)).reduceByKey(_ + _)      //Transformation: One more RDD from the existing Rdd

 counts: org.apache.spark.rdd.RDD[(String, Long)] = ShuffledRDD[4] at reduceByKey at
<console>:25
scala> val sortcounts = counts.collect().sortBy(wc => -wc._2)   //Action
sortcounts: Array[(String, Long)] = Array(("",85361), (the,4495), (to,4190), (of,3708), (and,3501),
(her,2156), (a,1982), (in,1906), (was,1837), (i,1732), (she,1660), (that,1483), (not,1425), (it,1416),
(”,1372), (he,1289), (you,1264), (his,1245), (be,1240), (as,1176), (had,1162), (with,1086), (for,1046),
(but,893), (is,850), (have,837), (at,797), (mr,766), (on,716), (him,702), (by,652), (my,648), (all,611),
(they,584), (so,568), (elizabeth,566), (were,561), (which,538), (could,521), (been,511), (from,498),
(very,477), (no,464), (would,462), (this,448), (their,437), (what,433), (your,424), (will,405), (them,396),
(me,393), (such,388), (said,380), (or,367), (an,357), (when,353), (are,349), (darcy,343), (mrs,339),
(do,335), (if,323), (there,321), (much,319), (more,315), (must,314), (am,31...

scala> sortcounts.take(20).foreach(println)

(,85361)

(the,4495)

(to,4190)

(of,3708)

(and,3501)

(her,2156)

(a,1982)

(in,1906)

(was,1837)

(i,1732)

(she,1660)

(that,1483)

(not,1425)

(it,1416)

(”,1372)

(he,1289)

(you,1264)

(his,1245)

(be,1240)

(as,1176)
```

As the highest word count was not actually a word, I reran the code with 21 most used words to get the actual 20 most used words (ignoring the first one)

```
scala> sortcounts.take(21).foreach(println)

(,85361)

(the,4495)

(to,4190)

(of,3708)

(and,3501)

(her,2156)

(a,1982)

(in,1906)

(was,1837)

(i,1732)

(she,1660)

(that,1483)

(not,1425)

(it,1416)

(”,1372)

(he,1289)

(you,1264)

(his,1245)

(be,1240)

(as,1176)
```

```
(had,1162)

scala>
```

Step 5: Use Monte Carlo to estimate the value of pi (Task 6)

Lastly, I prepared a py file that uses Monte Carlo to estimate the value of pi. Then, I used the following command to load the file in s3 bucket:

```
aws s3 cp C:\Users\zeeha\Desktop\Documents\Courses\CSCI381-C\Projects\Project2\est.py s3://zee786bucket/
```

After the file was loaded into s3, I used the following command to load the file from zee786bucket to local file system (master machine)

```
[hadoop@ip-192-168-84-168 ~]$ aws s3 cp  s3://zee786bucket/est.py .

download: s3://zee786bucket/est.py to ./est.py
```

Then, ran the py file using the command: [hadoop@ip-192-168-84-168 ~]$ spark-submit est.py

The output I got includes the log content of the program. This makes the actual result of the output (estimated value of pi) difficult to spot. Therefore, I included a bunch of end lines (/n) to make the actual output easier to locate (page 16)

```
22/04/16 22:18:27 INFO SparkContext: Running Spark version 2.4.8-amzn-1

22/04/16 22:18:27 INFO SparkContext: Submitted application: CalculatePi

22/04/16 22:18:27 INFO SecurityManager: Changing view acls to: hadoop

22/04/16 22:18:27 INFO SecurityManager: Changing modify acls to: hadoop

22/04/16 22:18:27 INFO SecurityManager: Changing view acls groups to:

22/04/16 22:18:27 INFO SecurityManager: Changing modify acls groups to:

22/04/16 22:18:27 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(hadoop); groups with view permissions: Set(); users  with modify permissions: Set(hadoop); groups with modify permissions: Set()

22/04/16 22:18:28 INFO Utils: Successfully started service 'sparkDriver' on port 37981.
```

```
22/04/16 22:18:28 INFO SparkEnv: Registering MapOutputTracker

22/04/16 22:18:28 INFO SparkEnv: Registering BlockManagerMaster

22/04/16 22:18:28 INFO BlockManagerMasterEndpoint: Using
org.apache.spark.storage.DefaultTopologyMapper for getting topology information

22/04/16 22:18:28 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up

22/04/16 22:18:28 INFO DiskBlockManager: Created local directory at
/mnt/tmp/blockmgr-573a10df-ea0f-49d3-93a5-be6823530a83

22/04/16 22:18:28 INFO MemoryStore: MemoryStore started with capacity 912.3 MB

22/04/16 22:18:28 INFO SparkEnv: Registering OutputCommitCoordinator

22/04/16 22:18:28 INFO Utils: Successfully started service 'SparkUI' on port 4040.

22/04/16 22:18:29 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at
http://ip-192-168-84-168.ec2.internal:4040

22/04/16 22:18:29 INFO Utils: Using 50 preallocated executors (minExecutors: 0). Set
spark.dynamicAllocation.preallocateExecutors to `false` disable executor preallocation.

22/04/16 22:18:30 INFO RMProxy: Connecting to ResourceManager at
ip-192-168-84-168.ec2.internal/192.168.84.168:8032

22/04/16 22:18:30 INFO Client: Requesting a new application from cluster with 2 NodeManagers

22/04/16 22:18:31 INFO Configuration: resource-types.xml not found

22/04/16 22:18:31 INFO ResourceUtils: Unable to find 'resource-types.xml'.

22/04/16 22:18:31 INFO ResourceUtils: Adding resource type - name = memory-mb, units = Mi, type =
COUNTABLE

22/04/16 22:18:31 INFO ResourceUtils: Adding resource type - name = vcores, units = , type =
COUNTABLE

22/04/16 22:18:31 INFO Client: Verifying our application has not requested more than the maximum
memory capability of the cluster (6144 MB per container)

22/04/16 22:18:31 INFO Client: Will allocate AM container, with 896 MB memory including 384 MB
overhead
```

```
22/04/16 22:18:31 INFO Client: Setting up container launch context for our AM

22/04/16 22:18:31 INFO Client: Setting up the launch environment for our AM container

22/04/16 22:18:31 INFO Client: Preparing resources for our AM container

22/04/16 22:18:31 WARN Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HOME.

22/04/16 22:18:35 INFO Client: Uploading resource file:/mnt/tmp/spark-6069dfb5-59c1-4148-9b2c-b95d2322e6d7/__spark_libs__3789218786151101032.zip -> hdfs://ip-192-168-84-168.ec2.internal:8020/user/hadoop/.sparkStaging/application_1650127996462_0010/__spark_libs__3789218786151101032.zip

22/04/16 22:18:38 INFO Client: Uploading resource file:/usr/lib/spark/python/lib/pyspark.zip -> hdfs://ip-192-168-84-168.ec2.internal:8020/user/hadoop/.sparkStaging/application_1650127996462_0010/pyspark.zip

22/04/16 22:18:38 INFO Client: Uploading resource file:/usr/lib/spark/python/lib/py4j-0.10.7-src.zip -> hdfs://ip-192-168-84-168.ec2.internal:8020/user/hadoop/.sparkStaging/application_1650127996462_0010/py4j-0.10.7-src.zip

22/04/16 22:18:39 INFO Client: Uploading resource file:/mnt/tmp/spark-6069dfb5-59c1-4148-9b2c-b95d2322e6d7/__spark_conf__8199921185668943979.zip -> hdfs://ip-192-168-84-168.ec2.internal:8020/user/hadoop/.sparkStaging/application_1650127996462_0010/__spark_conf__.zip

22/04/16 22:18:39 INFO SecurityManager: Changing view acls to: hadoop

22/04/16 22:18:39 INFO SecurityManager: Changing modify acls to: hadoop

22/04/16 22:18:39 INFO SecurityManager: Changing view acls groups to:

22/04/16 22:18:39 INFO SecurityManager: Changing modify acls groups to:

22/04/16 22:18:39 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(hadoop); groups with view permissions: Set(); users  with modify permissions: Set(hadoop); groups with modify permissions: Set()

22/04/16 22:18:41 INFO Client: Submitting application application_1650127996462_0010 to ResourceManager

22/04/16 22:18:41 INFO YarnClientImpl: Submitted application application_1650127996462_0010
```

```
22/04/16 22:18:41 INFO SchedulerExtensionServices: Starting Yarn extension services with app
application_1650127996462_0010 and attemptId None

22/04/16 22:18:42 INFO Client: Application report for application_1650127996462_0010 (state:
ACCEPTED)

22/04/16 22:18:42 INFO Client:

client token: N/A

diagnostics: AM container is launched, waiting for AM container to Register with RM

ApplicationMaster host: N/A

ApplicationMaster RPC port: -1

queue: default

start time: 1650147521212

final status: UNDEFINED

tracking URL: http://ip-192-168-84-168.ec2.internal:20888/proxy/application_1650127996462_0010/

user: hadoop

22/04/16 22:18:43 INFO Client: Application report for application_1650127996462_0010 (state:
ACCEPTED)

22/04/16 22:18:44 INFO Client: Application report for application_1650127996462_0010 (state:
ACCEPTED)

22/04/16 22:18:45 INFO Client: Application report for application_1650127996462_0010 (state:
ACCEPTED)

22/04/16 22:18:46 INFO Client: Application report for application_1650127996462_0010 (state:
ACCEPTED)

22/04/16 22:18:47 INFO Client: Application report for application_1650127996462_0010 (state:
ACCEPTED)

22/04/16 22:18:48 INFO Client: Application report for application_1650127996462_0010 (state:
RUNNING)

22/04/16 22:18:48 INFO Client:
```

```
client token: N/A

diagnostics: N/A

ApplicationMaster host: 192.168.91.31

ApplicationMaster RPC port: -1

queue: default

start time: 1650147521212

final status: UNDEFINED

tracking URL: http://ip-192-168-84-168.ec2.internal:20888/proxy/application_1650127996462_0010/

user: hadoop

22/04/16 22:18:48 INFO YarnClientSchedulerBackend: Application application_1650127996462_0010 has
started running.

22/04/16 22:18:48 INFO Utils: Successfully started service
'org.apache.spark.network.netty.NettyBlockTransferService' on port 35143.

22/04/16 22:18:48 INFO NettyBlockTransferService: Server created on
ip-192-168-84-168.ec2.internal:35143

22/04/16 22:18:48 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for
block replication policy

22/04/16 22:18:48 INFO YarnClientSchedulerBackend: Add WebUI Filter.
org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter, Map(PROXY_HOSTS ->
ip-192-168-84-168.ec2.internal, PROXY_URI_BASES ->
http://ip-192-168-84-168.ec2.internal:20888/proxy/application_1650127996462_0010),
/proxy/application_1650127996462_0010

22/04/16 22:18:48 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver,
ip-192-168-84-168.ec2.internal, 35143, None)

22/04/16 22:18:48 INFO BlockManagerMasterEndpoint: Registering block manager
ip-192-168-84-168.ec2.internal:35143 with 912.3 MB RAM, BlockManagerId(driver,
ip-192-168-84-168.ec2.internal, 35143, None)
```

```
22/04/16 22:18:48 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver,
ip-192-168-84-168.ec2.internal, 35143, None)

22/04/16 22:18:48 INFO BlockManager: external shuffle service port = 7337

22/04/16 22:18:48 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver,
ip-192-168-84-168.ec2.internal, 35143, None)

22/04/16 22:18:48 INFO JettyUtils: Adding filter
org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /metrics/json.

22/04/16 22:18:48 INFO SingleEventLogFileWriter: Logging events to
hdfs:/var/log/spark/apps/application_1650127996462_0010.inprogress

22/04/16 22:18:48 INFO YarnSchedulerBackend$YarnSchedulerEndpoint: ApplicationMaster registered as
NettyRpcEndpointRef(spark-client://YarnAM)

22/04/16 22:18:48 INFO Utils: Using 50 preallocated executors (minExecutors: 0). Set
spark.dynamicAllocation.preallocateExecutors to `false` disable executor preallocation.

22/04/16 22:18:49 INFO YarnClientSchedulerBackend: SchedulerBackend is ready for scheduling beginning
after reached minRegisteredResourcesRatio: 0.0

22/04/16 22:18:49 INFO SharedState: loading hive config file: file:/etc/spark/conf.dist/hive-site.xml

22/04/16 22:18:49 INFO SharedState: Setting hive.metastore.warehouse.dir ('null') to the value of
spark.sql.warehouse.dir ('hdfs:///user/spark/warehouse').

22/04/16 22:18:49 INFO SharedState: Warehouse path is 'hdfs:///user/spark/warehouse'.

22/04/16 22:18:49 INFO JettyUtils: Adding filter
org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /SQL.

22/04/16 22:18:49 INFO JettyUtils: Adding filter
org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /SQL/json.

22/04/16 22:18:49 INFO JettyUtils: Adding filter
org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /SQL/execution.

22/04/16 22:18:49 INFO JettyUtils: Adding filter
org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /SQL/execution/json.

22/04/16 22:18:49 INFO JettyUtils: Adding filter
org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /static/sql.
```

```
22/04/16 22:18:49 INFO StateStoreCoordinatorRef: Registered StateStoreCoordinator endpoint

22/04/16 22:18:50 INFO SparkContext: Starting job: reduce at /home/hadoop/est.py:24

22/04/16 22:18:50 INFO DAGScheduler: Got job 0 (reduce at /home/hadoop/est.py:24) with 2 output
partitions

22/04/16 22:18:50 INFO DAGScheduler: Final stage: ResultStage 0 (reduce at /home/hadoop/est.py:24)

22/04/16 22:18:50 INFO DAGScheduler: Parents of final stage: List()

22/04/16 22:18:50 INFO DAGScheduler: Missing parents: List()

22/04/16 22:18:50 INFO DAGScheduler: Submitting ResultStage 0 (PythonRDD[1] at reduce at
/home/hadoop/est.py:24), which has no missing parents

22/04/16 22:18:50 INFO MemoryStore: Block broadcast_0 stored as values in memory (estimated size 6.3
KB, free 912.3 MB)

22/04/16 22:18:50 INFO MemoryStore: Block broadcast_0_piece0 stored as bytes in memory (estimated size
4.3 KB, free 912.3 MB)

22/04/16 22:18:50 INFO BlockManagerInfo: Added broadcast_0_piece0 in memory on
ip-192-168-84-168.ec2.internal:35143 (size: 4.3 KB, free: 912.3 MB)

22/04/16 22:18:50 INFO SparkContext: Created broadcast 0 from broadcast at DAGScheduler.scala:1297

22/04/16 22:18:50 INFO DAGScheduler: Submitting 2 missing tasks from ResultStage 0 (PythonRDD[1] at
reduce at /home/hadoop/est.py:24) (first 15 tasks are for partitions Vector(0, 1))

22/04/16 22:18:50 INFO YarnScheduler: Adding task set 0.0 with 2 tasks

22/04/16 22:18:54 INFO YarnSchedulerBackend$YarnDriverEndpoint: Registered executor
NettyRpcEndpointRef(spark-client://Executor) (192.168.117.115:34644) with ID 2

22/04/16 22:18:54 INFO ExecutorAllocationManager: New executor 2 has registered (new total is 1)

22/04/16 22:18:54 INFO TaskSetManager: Starting task 0.0 in stage 0.0 (TID 0,
ip-192-168-117-115.ec2.internal, executor 2, partition 0, PROCESS_LOCAL, 8004 bytes)

22/04/16 22:18:54 INFO TaskSetManager: Starting task 1.0 in stage 0.0 (TID 1,
ip-192-168-117-115.ec2.internal, executor 2, partition 1, PROCESS_LOCAL, 8004 bytes)
```

```
22/04/16 22:18:54 INFO BlockManagerMasterEndpoint: Registering block manager
ip-192-168-117-115.ec2.internal:34117 with 2.1 GB RAM, BlockManagerId(2,
ip-192-168-117-115.ec2.internal, 34117, None)

22/04/16 22:18:55 INFO YarnSchedulerBackend$YarnDriverEndpoint: Registered executor
NettyRpcEndpointRef(spark-client://Executor) (192.168.91.31:55368) with ID 1

22/04/16 22:18:55 INFO ExecutorAllocationManager: New executor 1 has registered (new total is 2)

22/04/16 22:18:55 INFO BlockManagerInfo: Added broadcast_0_piece0 in memory on
ip-192-168-117-115.ec2.internal:34117 (size: 4.3 KB, free: 2.1 GB)

22/04/16 22:18:55 INFO BlockManagerMasterEndpoint: Registering block manager
ip-192-168-91-31.ec2.internal:36771 with 2.1 GB RAM, BlockManagerId(1, ip-192-168-91-31.ec2.internal,
36771, None)

22/04/16 22:19:08 INFO TaskSetManager: Finished task 1.0 in stage 0.0 (TID 1) in 13849 ms on
ip-192-168-117-115.ec2.internal (executor 2) (1/2)

22/04/16 22:19:08 INFO PythonAccumulatorV2: Connected to AccumulatorServer at host: 127.0.0.1 port:
37719

22/04/16 22:19:08 INFO TaskSetManager: Finished task 0.0 in stage 0.0 (TID 0) in 13948 ms on
ip-192-168-117-115.ec2.internal (executor 2) (2/2)

22/04/16 22:19:08 INFO DAGScheduler: ResultStage 0 (reduce at /home/hadoop/est.py:24) finished in
18.150 s

22/04/16 22:19:08 INFO YarnScheduler: Removed TaskSet 0.0, whose tasks have all completed, from pool

22/04/16 22:19:08 INFO DAGScheduler: Job 0 finished: reduce at /home/hadoop/est.py:24, took 18.265990
s
```

Pi is roughly 3.141295

```
22/04/16 22:19:08 INFO SparkUI: Stopped Spark web UI at http://ip-192-168-84-168.ec2.internal:4040

22/04/16 22:19:08 INFO YarnClientSchedulerBackend: Interrupting monitor thread

22/04/16 22:19:08 INFO YarnClientSchedulerBackend: Shutting down all executors

22/04/16 22:19:08 INFO YarnSchedulerBackend$YarnDriverEndpoint: Asking each executor to shut down

22/04/16 22:19:08 INFO SchedulerExtensionServices: Stopping SchedulerExtensionServices

(serviceOption=None,

services=List(),

started=false)

22/04/16 22:19:08 INFO YarnClientSchedulerBackend: Stopped

22/04/16 22:19:08 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!

22/04/16 22:19:08 INFO MemoryStore: MemoryStore cleared

22/04/16 22:19:08 INFO BlockManager: BlockManager stopped

22/04/16 22:19:08 INFO BlockManagerMaster: BlockManagerMaster stopped

22/04/16 22:19:08 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint:
OutputCommitCoordinator stopped!

22/04/16 22:19:08 INFO SparkContext: Successfully stopped SparkContext

22/04/16 22:19:09 INFO ShutdownHookManager: Shutdown hook called

22/04/16 22:19:09 INFO ShutdownHookManager: Deleting directory
/mnt/tmp/spark-6069dfb5-59c1-4148-9b2c-b95d2322e6d7/pyspark-cda6bf0d-2d39-4442-82c1-f62348c0099
b
```

```
22/04/16 22:19:09 INFO ShutdownHookManager: Deleting directory
/mnt/tmp/spark-6069dfb5-59c1-4148-9b2c-b95d2322e6d7

22/04/16 22:19:09 INFO ShutdownHookManager: Deleting directory
/mnt/tmp/spark-b6de89db-db87-41db-b898-bb63ba4a3ff2

[hadoop@ip-192-168-84-168 ~]$
```