The Course Project

The course project consists of four parts. Parts 1, 2, 3 require the use of the project data zip file, while parts 4 require the use of a text file generated from part 3.

Part 1:

Develop a Mapper and Reducer application to calculate the *average* of *wind direction* (degree) for *each observation month from each year* (e.g. 195001) from NCDC records (note: 999 indicates missing value, and [01459] indicate good quality value).

Part 2:

Develop a python application that can be implemented in PySpark to calculate the *range* (the difference between max and min values) of *sky ceiling height* (meters) for *each USAF weather station ID* from NCDC records (note: 99999 indicates missing value, and [01459] indicate good quality value).

Part 3:

Develop a Mapper and Reducer application to retrieve *USAF weather station ID* and *visibility distance (meters)* from NCDC records (note: 999999 indicates missing value, and [01459] indicate good quality value) and then *write* the USAF weather station ID and visibility distance data into *a text file*.

Part 4:

Load the text file into Pig and get the range of *visibility distance* for each USAF weather station ID.

Load the text file into Hive and get the average *visibility distance* for each USAF weather station ID.

You need to turn in:

1) Part 1:

a. *if you are using JAVA to develop the Mapper and Reducer applications:* the three java files (mapper, reducer and main);

b. *if you are using Hadoop streaming jar and developing two python programs (mapper python file and reducer python file)*: the two python files (mapper and reducer);

c. *if you are using mrjob library and developing one python program with two functions:* the python file (with the mapper and reducer functions);

2) Part 2: the python program you developed;

3) Part 3: refer to Part 1; and the text file created;

4) the commands from converting java files into a Jar file to running the Jar file in Hadoop, or the commands to execute the python files in Hadoop and in Spark, all commands in Pig and Hive;

5) the step by step commands and screenshots of solutions from all the parts;

The original dataset for this project is available on Blackboard.

1. Develop a Mapper and Reducer application to calculate the *average* of *wind direction* (degree) for *each observation month from each year* (e.g. 195001) from NCDC records (note: 999 indicates missing value, and [01459] indicate good quality value).

<u>Step 1 : Utilizing Hadoop streaming jar and developing two python programs. Created the two python files : Mapper_part1.py and Reducer_part1.py to calculate the *average* of *wind direction* (degree) for *each observation month from each year*</u>

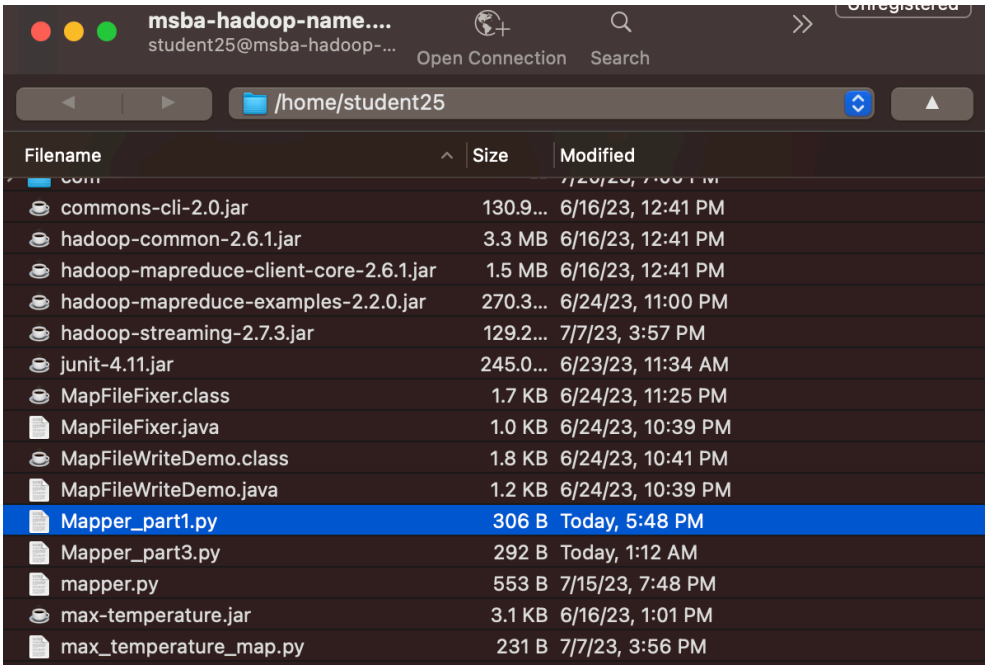
Mapper_part1.py

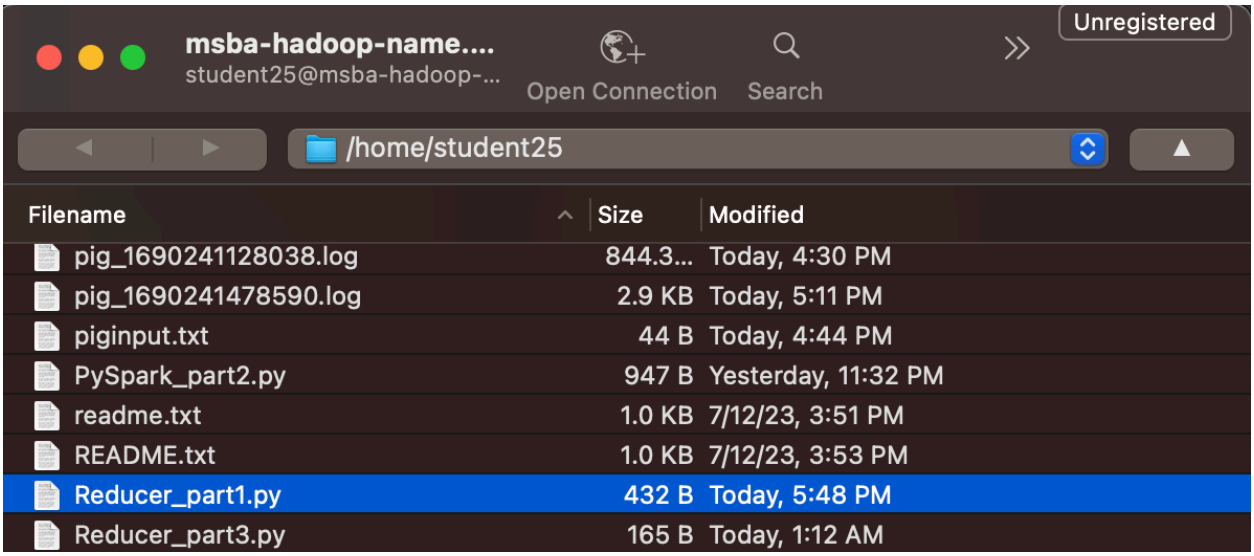
```
#!/usr/bin/env python

import re
import sys

for line in sys.stdin:
  val = line.strip()
  (observation_date, wind_direction, quality_code) = (val[15:21], val[60:63], val[63:64])
  if (wind_direction != "999" and re.match("[01459]", quality_code)):
    print("%s\t%s" % (observation_date, wind_direction))

```

Reducer_part1.py

```python
#!/usr/bin/env python

import sys
(last_key, count, sum_wind) = (None, 0, 0)
for line in sys.stdin:
  (key, val) = line.strip().split("\t")
  if last_key and last_key != key:
    print("%s\t%s" % (last_key, sum_wind/count))
    (last_key, count, sum_wind) = (key, 1, int(val))
  else:
    (last_key,count,sum_wind) = (key, count+1, sum_wind + int(val))

if last_key:
  print("%s\t%s" % (last_key, sum_wind / count))
```

Step 2 : Moved the Mapper_part1.py and Reducer_part1.py to local server using cyberduck





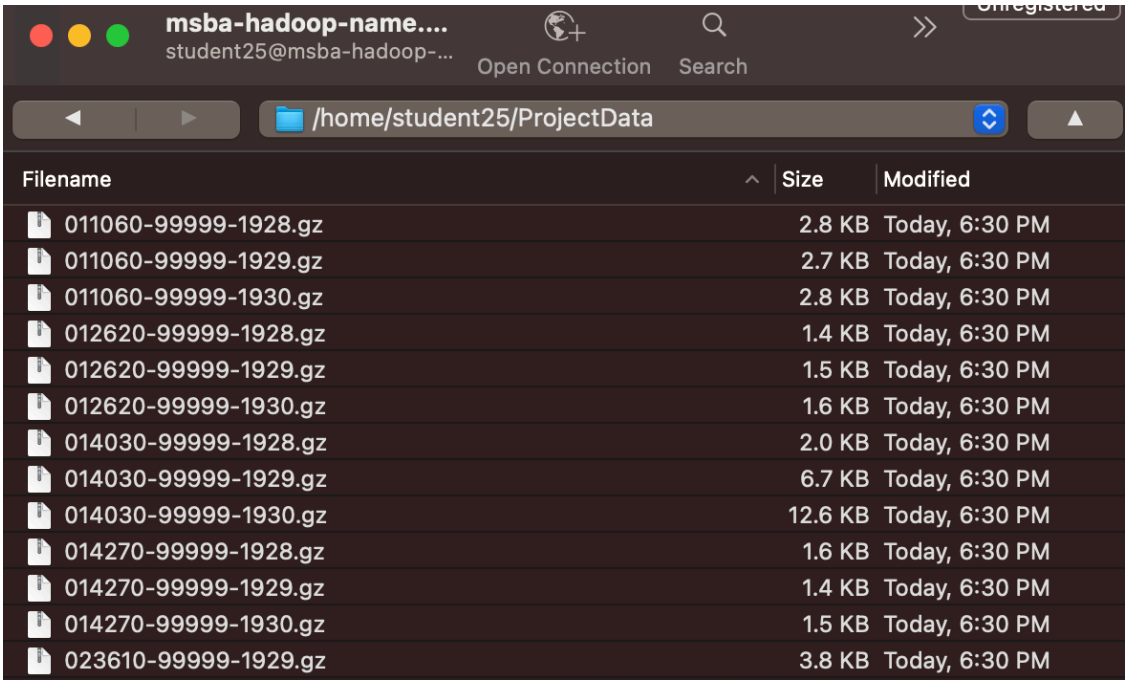
Step 3 : Changed the execution permission of the python files

chmod +x Mapper_part1.py
chmod +x Reducer_part1.py

```
[student25@msba-hadoop-name ~]$ chmod +x Mapper_part1.py
[student25@msba-hadoop-name ~]$ chmod +x Reducer_part1.py
```

Step 4: Copying the Project data:

First of all, we will copy the "ProjectData" folder which contains all the files from our desktop to the local server using cyberduck.



After this, we will create a directory in HDFS for containing all the project data. Below are the steps :

hdfs dfs -mkdir /home/25student25/BAN632Project/

hdfs dfs -copyFromLocal /home/student25/ProjectData/ /home/25student25/BAN632Project

hdfs dfs -ls /home/25student25/BAN632Project/ProjectData/

```
[student25@msba-hadoop-name ~]$ hdfs dfs -mkdir /home/25student25/BAN632Project/

[student25@msba-hadoop-name ~]$ hdfs dfs -ls /home/25student25/BAN632Project/
[student25@msba-hadoop-name ~]$ hdfs dfs -copyFromLocal /home/student25/ProjectD
ata/ /home/25student25/BAN632Project
[student25@msba-hadoop-name ~]$ hdfs dfs -ls /home/25student25/BAN632Project/Pro
jectData/
```

```
[student25@msba-hadoop-name ~]$ hdfs dfs -ls /home/25student25/BAN632Project/Pro
jectData/
Found 50 items
-rw-r--r--   5 student25 supergroup       2788 2023-07-24 19:12 /home/25student2
5/BAN632Project/ProjectData/011060-99999-1928.gz
-rw-r--r--   5 student25 supergroup       2660 2023-07-24 19:12 /home/25student2
5/BAN632Project/ProjectData/011060-99999-1929.gz
-rw-r--r--   5 student25 supergroup       2797 2023-07-24 19:12 /home/25student2
5/BAN632Project/ProjectData/011060-99999-1930.gz
-rw-r--r--   5 student25 supergroup       1428 2023-07-24 19:12 /home/25student2
5/BAN632Project/ProjectData/012620-99999-1928.gz
-rw-r--r--   5 student25 supergroup       1506 2023-07-24 19:12 /home/25student2
5/BAN632Project/ProjectData/012620-99999-1929.gz
-rw-r--r--   5 student25 supergroup       1570 2023-07-24 19:12 /home/25student2
5/BAN632Project/ProjectData/012620-99999-1930.gz
-rw-r--r--   5 student25 supergroup       1973 2023-07-24 19:12 /home/25student2
5/BAN632Project/ProjectData/014030-99999-1928.gz
-rw-r--r--   5 student25 supergroup       6658 2023-07-24 19:12 /home/25student2
5/BAN632Project/ProjectData/014030-99999-1929.gz
-rw-r--r--   5 student25 supergroup      12618 2023-07-24 19:12 /home/25student2
```

Step 4: Executing the mapper and reducer using Hadoop streaming:

hadoop jar hadoop-streaming-2.7.3.jar -file /home/student25/Mapper_part1.py -mapper /home/student25/Mapper_part1.py -file /home/student25/Reducer_part1.py -reducer /home/student25/Reducer_part1.py -input /home/25student25/BAN632Project/ProjectData/ -output /home/25student25/ProjectOutput/

```
[student25@msba-hadoop-name ~]$ hadoop jar hadoop-streaming-2.7.3.jar -file /home/student25/Mapper_part1.py -mapper /home/student2
5/Mapper_part1.py -file /home/student25/Reducer_part1.py   -reducer /home/student25/Reducer_part1.py -input /home/25student25/BAN6
32Project/ProjectData/ -output /home/25student25/ProjectOutput/
23/07/24 19:32:04 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [/home/student25/Mapper_part1.py, /home/student25/Reducer_part1.py, /tmp/hadoop-unjar1235150324716062465/] [] /tmp/
streamjob6282438524562338915.jar tmpDir=null
23/07/24 19:32:05 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0.1:8032
23/07/24 19:32:05 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0.1:8032
23/07/24 19:32:05 INFO mapred.FileInputFormat: Total input files to process : 50
23/07/24 19:32:05 INFO mapreduce.JobSubmitter: number of splits:50
23/07/24 19:32:05 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, us
e yarn.system-metrics-publisher.enabled
23/07/24 19:32:05 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1669743171306_4593
23/07/24 19:32:06 INFO impl.YarnClientImpl: Submitted application application_1669743171306_4593
23/07/24 19:32:06 INFO mapreduce.Job: The url to track the job: http://msba-hadoop-name:8088/proxy/application_1669743171306_4593/
23/07/24 19:32:06 INFO mapreduce.Job: Running job: job_1669743171306_4593
23/07/24 19:32:12 INFO mapreduce.Job: Job job_1669743171306_4593 running in uber mode : false
23/07/24 19:32:12 INFO mapreduce.Job:  map 0% reduce 0%
23/07/24 19:32:21 INFO mapreduce.Job:  map 12% reduce 0%
23/07/24 19:32:28 INFO mapreduce.Job:  map 20% reduce 0%
23/07/24 19:32:29 INFO mapreduce.Job:  map 22% reduce 0%
23/07/24 19:32:30 INFO mapreduce.Job:  map 24% reduce 0%
23/07/24 19:32:35 INFO mapreduce.Job:  map 28% reduce 0%
23/07/24 19:32:36 INFO mapreduce.Job:  map 32% reduce 0%
23/07/24 19:32:37 INFO mapreduce.Job:  map 34% reduce 0%
23/07/24 19:32:41 INFO mapreduce.Job:  map 36% reduce 0%
23/07/24 19:32:42 INFO mapreduce.Job:  map 40% reduce 0%
23/07/24 19:32:43 INFO mapreduce.Job:  map 44% reduce 0%
23/07/24 19:32:46 INFO mapreduce.Job:  map 46% reduce 0%
23/07/24 19:32:48 INFO mapreduce.Job:  map 46% reduce 15%
23/07/24 19:32:50 INFO mapreduce.Job:  map 50% reduce 15%
23/07/24 19:32:51 INFO mapreduce.Job:  map 54% reduce 15%
23/07/24 19:32:52 INFO mapreduce.Job:  map 56% reduce 15%
23/07/24 19:32:55 INFO mapreduce.Job:  map 56% reduce 19%
```

```
                Total megabyte-milliseconds taken by all reduce tasks=53875712
        Map-Reduce Framework
                Map input records=36404
                Map output records=28600
                Map output bytes=314600
                Map output materialized bytes=372100
                Input split bytes=7200
                Combine input records=0
                Combine output records=0
                Reduce input groups=110
                Reduce shuffle bytes=372100
                Reduce input records=28600
                Reduce output records=110
                Spilled Records=57200
                Shuffled Maps =50
                Failed Shuffles=0
                Merged Map outputs=50
                GC time elapsed (ms)=11598
                CPU time spent (ms)=34440
                Physical memory (bytes) snapshot=17518616576
                Virtual memory (bytes) snapshot=154121281536
                Total committed heap usage (bytes)=15569780736
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=415313
        File Output Format Counters
                Bytes Written=1210
23/07/24 19:33:24 INFO streaming.StreamJob: Output directory: /home/25student25/ProjectOutput/
[student25@msba-hadoop-name ~]$
```

Step 5 : Print out the content of the output

hdfs dfs -ls /home/25student25/ProjectOutput/

hdfs dfs -cat /home/25student25/ProjectOutput/part-00000

```
[student25@msba-hadoop-name ~]$ hdfs dfs -ls /home/25student25/ProjectOutput/
Found 2 items
-rw-r--r--   5 student25 supergroup          0 2023-07-24 19:33 /home/25student25/ProjectOutput/_SUCCESS
-rw-r--r--   5 student25 supergroup       1210 2023-07-24 19:33 /home/25student25/ProjectOutput/part-00000
[student25@msba-hadoop-name ~]$ hdfs dfs -cat /home/25student25/ProjectOutput/part-00000
192101  205
192102  219
192103  225
192104  201
192105  226
192106  222
192107  219
192108  200
192109  242
192110  243
192111  205
192112  211
192201  154
192202  178
192203  226
192204  176
192205  216
192206  194
192207  187
192208  199
192209  216
192210  250
192211  235
192212  206
192301  209
192302  169
192303  218
192304  208
192305  198
192306  227
```

2. Develop a python application that can be implemented in PySpark to calculate the *range* (the difference between max and min values) of *sky ceiling height* (meters) for *each USAF weather station ID* from NCDC records (note: 99999 indicates missing value, and [01459] indicate good quality value).

Step 1 : Created a python application that can be implemented in PySpark to calculate the *range* (the difference between max and min values) of *sky ceiling height* (meters) for *each USAF weather station ID*

PySpark_part2.py

```python
from pyspark import SparkContext
def main():
    #Create a sparkcontext
    sc = SparkContext(appName='RangeSkyceilingbyWeatherstation')

    # Import the input file
    input_rdd = sc.textFile('/user/hadoop/BAN632Project/ProjectData/')

    # filter out records with invalid quantity values and sky ceiling height
    filtered_rdd = input_rdd.filter(lambda line: line[70:75] != "99999" and int(line[75:76]) in [0,1,4,5,9])

    # extract the USAF weather station ID and sky ceiling height and convert them into pairs
    usaf_rdd = filtered_rdd.map(lambda line: (line[4:10], int(line[70:75])))

    # calculate the the range of sky ceiling height (meters) for each USAF weather station ID
    skyrange_rdd = usaf_rdd.groupByKey().mapValues(lambda values: max(values) - min(values))
    # save the output to a final output file
    skyrange_rdd.saveAsTextFile('/user/hadoop/outputpart2')
    sc.stop()
if __name__ == '__main__':
    main()
```

main()

Step 2 : Created Spark Cluster on AWS and established connection through key

Cluster: My cluster Waiting Cluster ready to run steps.

Summary | Application user interfaces | Monitoring | Hardware | Configurations | Events | Steps | Bootstrap actions

Summary

ID: j-39APQUK3BG261
Creation date: 2023-07-24 17:05 (UTC-10)
Elapsed time: 4 minutes
After last step completes: Cluster waits
Termination protection: Off Change
Tags: -- View All / Edit
Master public DNS: ec2-34-207-199-183.compute-1.amazonaws.com
Connect to the Master Node Using SSH

Configuration details

Release label: emr-5.36.1
Hadoop distribution: Amazon
Applications: Spark 2.4.8, Zeppelin 0.10.0
Log URI: s3://aws-logs-969509852754-us-east-1/elasticmapreduce/
EMRFS consistent view: Disabled
Custom AMI ID: --
Amazon Linux Release: 2.0.20230628.0 Learn more

Application user interfaces

Persistent user interfaces: Spark history server, YARN timeline server
On-cluster user interfaces: Not Enabled Enable an SSH Connection

Network and hardware

Availability zone: us-east-1d
Subnet ID: subnet-0639a4a20faf366fc
Master: Bootstrapping 1 m5.xlarge
Core: --
Task: --
Cluster scaling: Not enabled
Auto-termination: Terminate if idle for 1 hour

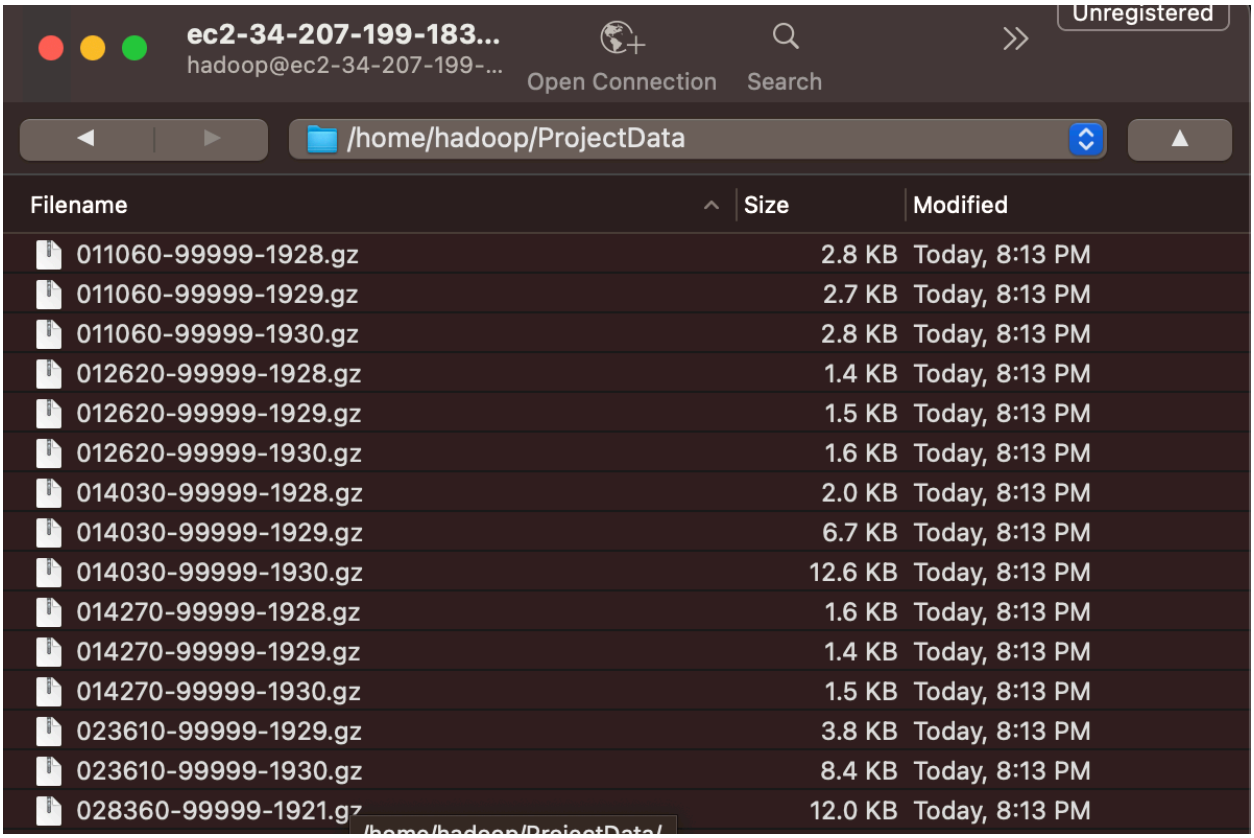
Security and access

Key name: BAN632
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole
Visible to all users: All Change

```
aasthatandon@MacBook-Air-4 ~ % cd /Users/aasthatandon/Desktop/Key
aasthatandon@MacBook-Air-4 Key % ssh -i BAN632.pem hadoop@ec2-34-207-199-183.com
pute-1.amazonaws.com
The authenticity of host 'ec2-34-207-199-183.compute-1.amazonaws.com (34.207.199
.183)' can't be established.
ED25519 key fingerprint is SHA256:7LLWbguNijhWJYnoE7ePejjNV1bRjGC9374XPx0PjhA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-207-199-183.compute-1.amazonaws.com' (ED25519
) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
20 package(s) needed for security, out of 21 available
Run "sudo yum update" to apply all updates.
```

Step 3 : Copying the Project data

Since we are using AWS EMR for this question, we will again copy the "ProjectData" folder which contains all the files from our desktop to the local server using cyberduck.

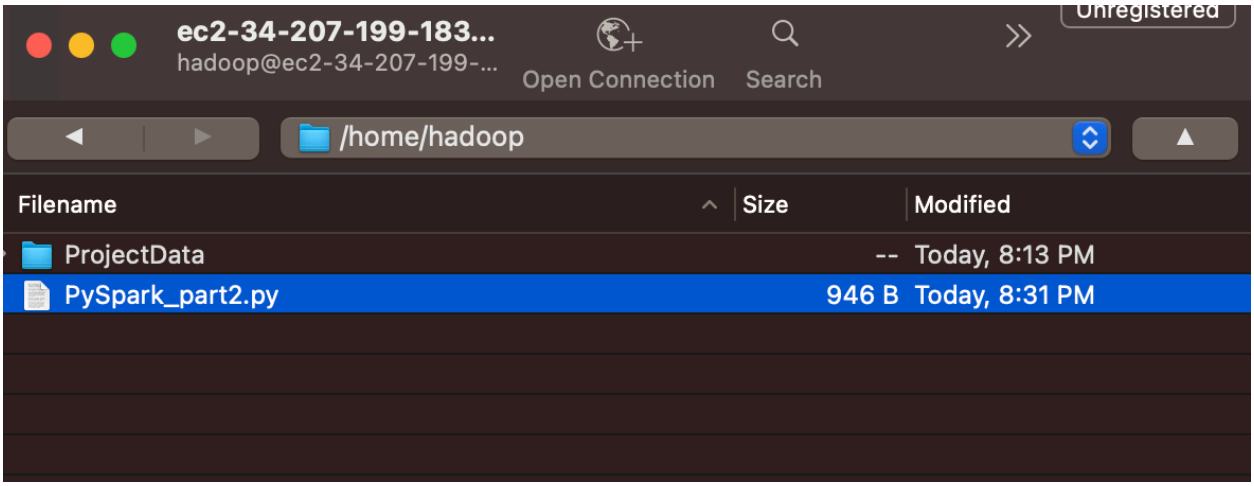


After this, we will create a directory in HDFS for containing all the project data. Below are the steps :

```
hdfs dfs -mkdir /user/hadoop/BAN632Project/
hdfs dfs -copyFromLocal /home/hadoop/ProjectData/ /user/hadoop/BAN632Project/
hdfs dfs -ls /user/hadoop/BAN632Project/ProjectData/
```

```
[hadoop@ip-172-31-46-30 ~]$ hdfs dfs -ls /user/hadoop/BAN632Project/ProjectData/
Found 50 items
-rw-r--r--   1 hadoop hdfsadmingroup       2788 2023-07-25 03:15 /user/hadoop/BA
N632Project/ProjectData/011060-99999-1928.gz
-rw-r--r--   1 hadoop hdfsadmingroup       2660 2023-07-25 03:15 /user/hadoop/BA
N632Project/ProjectData/011060-99999-1929.gz
-rw-r--r--   1 hadoop hdfsadmingroup       2797 2023-07-25 03:15 /user/hadoop/BA
N632Project/ProjectData/011060-99999-1930.gz
-rw-r--r--   1 hadoop hdfsadmingroup       1428 2023-07-25 03:15 /user/hadoop/BA
N632Project/ProjectData/012620-99999-1928.gz
-rw-r--r--   1 hadoop hdfsadmingroup       1506 2023-07-25 03:15 /user/hadoop/BA
N632Project/ProjectData/012620-99999-1929.gz
-rw-r--r--   1 hadoop hdfsadmingroup       1570 2023-07-25 03:15 /user/hadoop/BA
N632Project/ProjectData/012620-99999-1930.gz
-rw-r--r--   1 hadoop hdfsadmingroup       1973 2023-07-25 03:15 /user/hadoop/BA
N632Project/ProjectData/014030-99999-1928.gz
-rw-r--r--   1 hadoop hdfsadmingroup       6658 2023-07-25 03:15 /user/hadoop/BA
N632Project/ProjectData/014030-99999-1929.gz
-rw-r--r--   1 hadoop hdfsadmingroup      12618 2023-07-25 03:15 /user/hadoop/BA
N632Project/ProjectData/014030-99999-1930.gz
-rw-r--r--   1 hadoop hdfsadmingroup       1551 2023-07-25 03:15 /user/hadoop/BA
N632Project/ProjectData/014270-99999-1928.gz
-rw-r--r--   1 hadoop hdfsadmingroup       1395 2023-07-25 03:15 /user/hadoop/BA
N632Project/ProjectData/014270-99999-1929.gz
```

Step 4 : Copied PySpark_part2.py to Cyberduck



Step 5 : Command to execute python program in PySpark yarn

spark-submit --master yarn PySpark_part2.py

```
[hadoop@ip-172-31-46-30 ~]$ spark-submit --master yarn PySpark_part2.py
23/07/25 03:35:36 INFO SparkContext: Running Spark version 2.4.8-amzn-2
23/07/25 03:35:36 INFO SparkContext: Submitted application: RangeSkyceilingbyWea
therstation
23/07/25 03:35:36 INFO SecurityManager: Changing view acls to: hadoop
23/07/25 03:35:36 INFO SecurityManager: Changing modify acls to: hadoop
23/07/25 03:35:36 INFO SecurityManager: Changing view acls groups to:
23/07/25 03:35:36 INFO SecurityManager: Changing modify acls groups to:
23/07/25 03:35:36 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 mod
ify permissions: Set()
23/07/25 03:35:36 INFO Utils: Successfully started service 'sparkDriver' on port
 38091.
23/07/25 03:35:36 INFO SparkEnv: Registering MapOutputTracker
23/07/25 03:35:36 INFO SparkEnv: Registering BlockManagerMaster
23/07/25 03:35:36 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storag
e.DefaultTopologyMapper for getting topology information
23/07/25 03:35:36 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
23/07/25 03:35:36 INFO DiskBlockManager: Created local directory at /mnt/tmp/blo
ckmgr-a403c7ba-7afb-4787-a86e-300dd7e34789
23/07/25 03:35:36 INFO MemoryStore: MemoryStore started with capacity 912.3 MB
23/07/25 03:35:36 INFO SparkEnv: Registering OutputCommitCoordinator
```

```
23/07/25 03:36:01 INFO SparkContext: Successfully stopped SparkContext
23/07/25 03:36:01 INFO ShutdownHookManager: Shutdown hook called
23/07/25 03:36:01 INFO ShutdownHookManager: Deleting directory /mnt/tmp/spark-75
391052-1f18-4e18-ba78-c353b8abce60/pyspark-7bc91b77-ef8b-4704-a27c-e7bcd86f1cbf
23/07/25 03:36:01 INFO ShutdownHookManager: Deleting directory /mnt/tmp/spark-75
391052-1f18-4e18-ba78-c353b8abce60
23/07/25 03:36:01 INFO ShutdownHookManager: Deleting directory /mnt/tmp/spark-1b
951a72-069b-4b86-8441-b66bfd4ccabc
[hadoop@ip-172-31-46-30 ~]$
```

Step 6 : Printing the output

hdfs dfs -ls /user/hadoop/outputpart2/

```
[hadoop@ip-172-31-46-30 ~]$ hdfs dfs -ls /user/hadoop/outputpart2/
Found 51 items
-rw-r--r--   1 hadoop hdfsadmingroup          0 2023-07-25 03:36 /user/hadoop/ou
tputpart2/_SUCCESS
-rw-r--r--   1 hadoop hdfsadmingroup          0 2023-07-25 03:35 /user/hadoop/ou
tputpart2/part-00000
-rw-r--r--   1 hadoop hdfsadmingroup          0 2023-07-25 03:35 /user/hadoop/ou
tputpart2/part-00001
-rw-r--r--   1 hadoop hdfsadmingroup          0 2023-07-25 03:35 /user/hadoop/ou
tputpart2/part-00002
-rw-r--r--   1 hadoop hdfsadmingroup          0 2023-07-25 03:35 /user/hadoop/ou
tputpart2/part-00003
-rw-r--r--   1 hadoop hdfsadmingroup          0 2023-07-25 03:35 /user/hadoop/ou
tputpart2/part-00004
-rw-r--r--   1 hadoop hdfsadmingroup         18 2023-07-25 03:35 /user/hadoop/ou
tputpart2/part-00005
```

hdfs dfs -cat /user/hadoop/outputpart2/part-000**

```
[hadoop@ip-172-31-46-30 ~]$ hdfs dfs -cat /user/hadoop/outputpart2/part-000**
('023610', 21985)
('014270', 21985)
('034970', 21985)
('012620', 21985)
('033020', 21985)
('032620', 21760)
('014030', 21985)
('030910', 21985)
('011060', 21985)
('038040', 21940)
```

Part 3:

Develop a Mapper and Reducer application to retrieve *USAF weather station ID* and *visibility distance (meters)* from NCDC records (note: 999999 indicates missing value, and [01459] indicate good quality value) and then *write* the USAF weather station ID and visibility distance data into *a text file*.

Step 1 : Utilizing Hadoop streaming jar and developing two python programs. Created the two python files : Mapper_part3.py and Reducer_part3.py

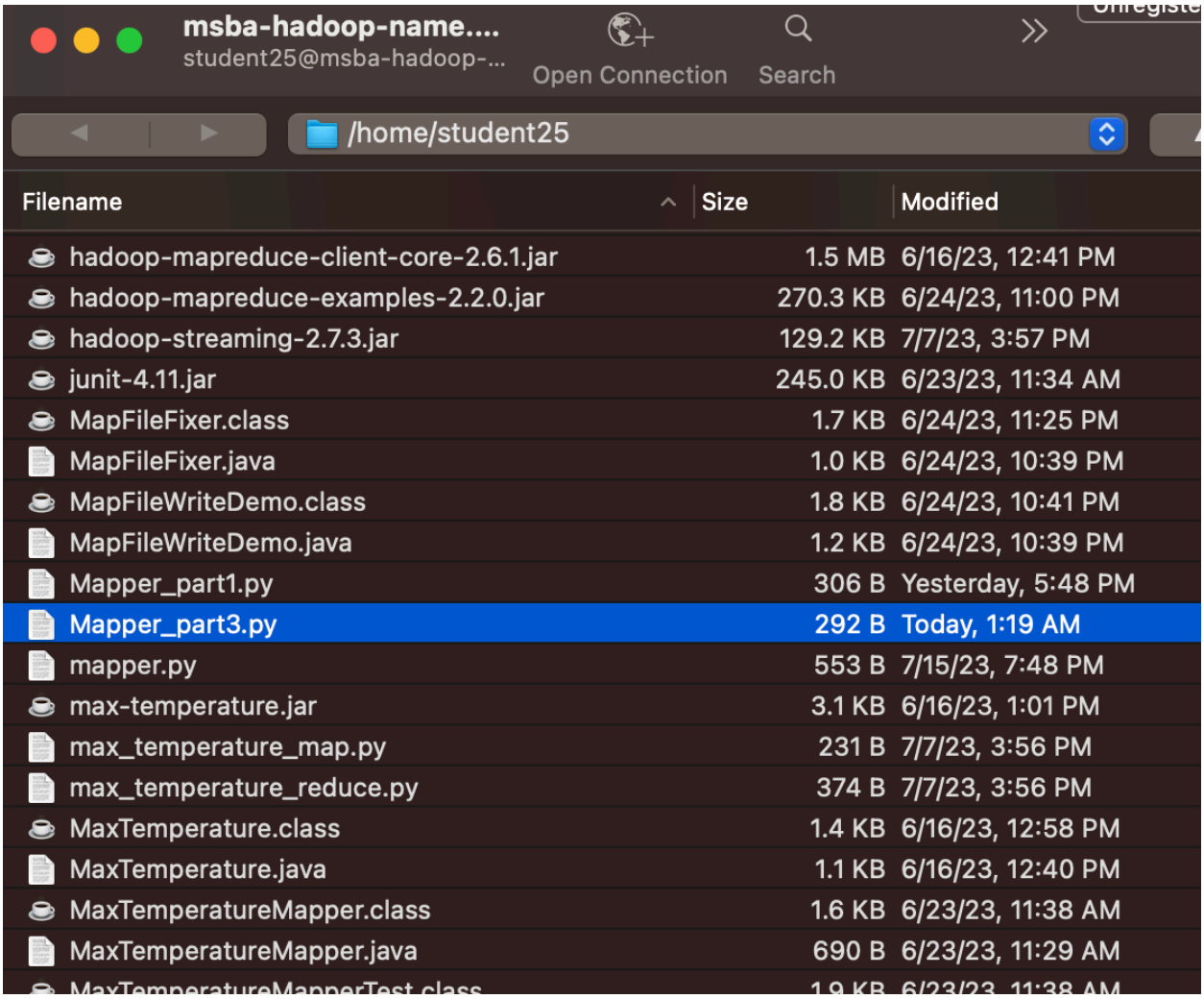
Mapper_part3.py

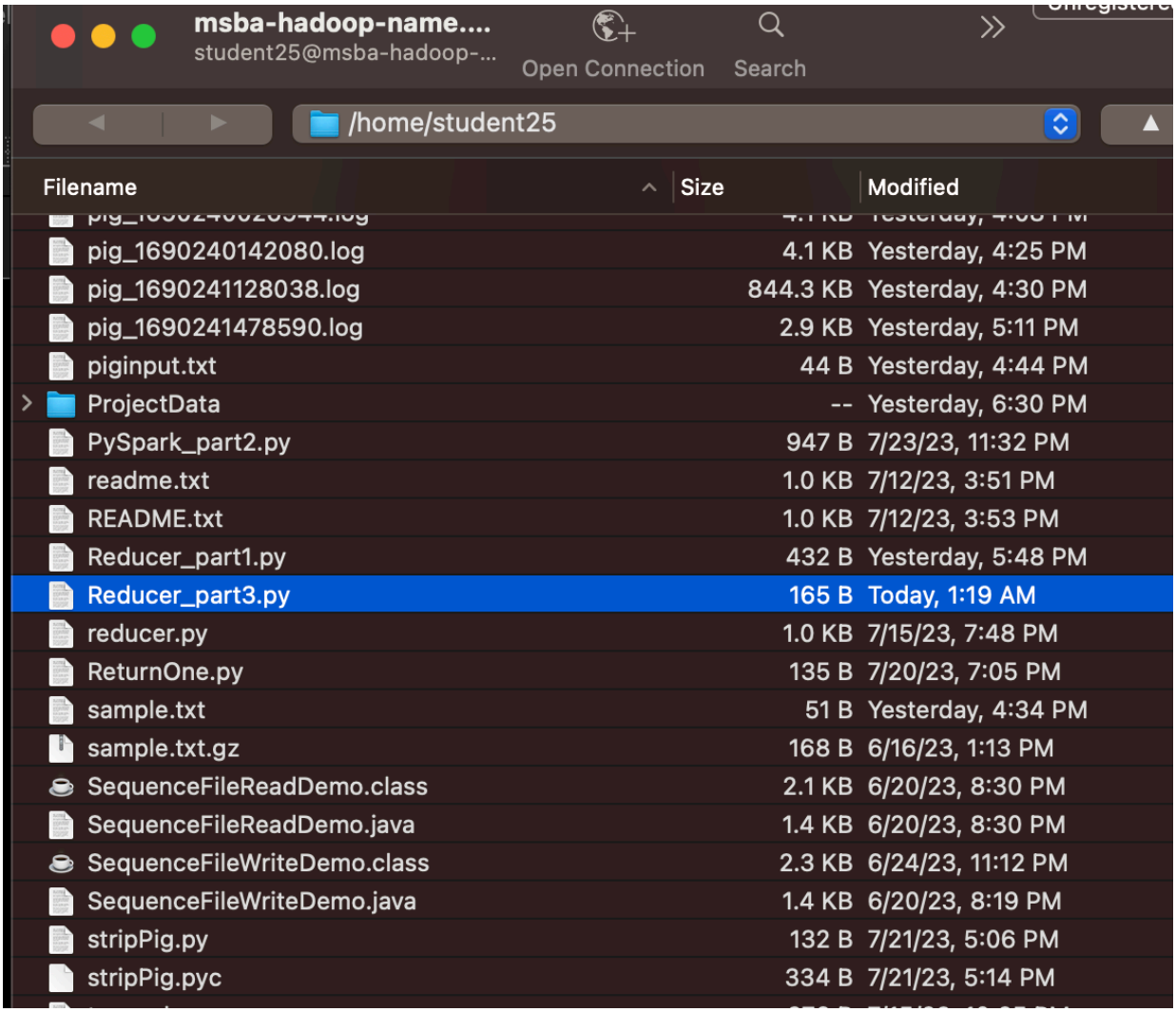
```
#!/usr/bin/env python

import re
import sys

for line in sys.stdin:
  val = line.strip()
  (usaf_id, visibility_dist, quality_code) = (val[4:10], val[78:84], val[84:85])
  if (visibility_dist != "999999" and re.match("[01459]", quality_code)):
    print("%s\t%s" % (usaf_id, visibility_dist))

```

Reducer_part3.py

```
#!/usr/bin/env python

import sys

for line in sys.stdin:
  usaf_id, visibility_dist = line.strip().split("\t")
  print("%s\t%s" % (usaf_id, visibility_dist))

```

Step 2 : Moved the Mapper_part3.py and Reducer_part3.py to local server using cyberduck


msba-hadoop-name....
student25@msba-hadoop-...
Open Connection
Search
/home/student25
Filename
Size
Modified
hadoop-mapreduce-client-core-2.6.1.jar 1.5 MB 6/16/23, 12:41 PM
hadoop-mapreduce-examples-2.2.0.jar 270.3 KB 6/24/23, 11:00 PM
hadoop-streaming-2.7.3.jar 129.2 KB 7/7/23, 3:57 PM
junit-4.11.jar 245.0 KB 6/23/23, 11:34 AM
MapFileFixer.class 1.7 KB 6/24/23, 11:25 PM
MapFileFixer.java 1.0 KB 6/24/23, 10:39 PM
MapFileWriteDemo.class 1.8 KB 6/24/23, 10:41 PM
MapFileWriteDemo.java 1.2 KB 6/24/23, 10:39 PM
Mapper_part1.py 306 B Yesterday, 5:48 PM
Mapper_part3.py 292 B Today, 1:19 AM
mapper.py 553 B 7/15/23, 7:48 PM
max-temperature.jar 3.1 KB 6/16/23, 1:01 PM
max_temperature_map.py 231 B 7/7/23, 3:56 PM
max_temperature_reduce.py 374 B 7/7/23, 3:56 PM
MaxTemperature.class 1.4 KB 6/16/23, 12:58 PM
MaxTemperature.java 1.1 KB 6/16/23, 12:40 PM
MaxTemperatureMapper.class 1.6 KB 6/23/23, 11:38 AM
MaxTemperatureMapper.java 690 B 6/23/23, 11:29 AM



Step 3 : Changed the execution permission of the python files

chmod +x Mapper_part3.py
chmod +x Reducer_part3.py

```
Last login: Mon Jul 24 17:51:24 2023 from 24.56.142.70
[student25@msba-hadoop-name ~]$ chmod +x Mapper_part3.py
[student25@msba-hadoop-name ~]$ chmod +x Reducer_part3.py
```

Step 4: Executing the mapper and reducer using Hadoop streaming:

We already copied the Project Data in part 1, hence we have data in
/home/25student25/BAN632Project/ProjectData/

hadoop jar hadoop-streaming-2.7.3.jar -file /home/student25/Mapper_part3.py -mapper /home/student25/Mapper_part3.py -file /home/student25/Reducer_part3.py -reducer /home/student25/Reducer_part3.py -input /home/25student25/BAN632Project/ProjectData/ -output /home/25student25/Projectoutputpart3/

```
[student25@msba-hadoop-name ~]$ hadoop jar hadoop-streaming-2.7.3.jar -file /hom
e/student25/Mapper_part3.py -mapper /home/student25/Mapper_part3.py -file /home/
student25/Reducer_part3.py   -reducer /home/student25/Reducer_part3.py -input /h
ome/25student25/BAN632Project/ProjectData/ -output /home/25student25/Projectoutp
utpart3/
23/07/25 01:25:07 WARN streaming.StreamJob: -file option is deprecated, please u
se generic option -files instead.
packageJobJar: [/home/student25/Mapper_part3.py, /home/student25/Reducer_part3.p
y, /tmp/hadoop-unjar3388726722311866116/] [] /tmp/streamjob5554862460807738960.j
ar tmpDir=null
23/07/25 01:25:08 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0
.1:8032
23/07/25 01:25:08 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0
.1:8032
23/07/25 01:25:08 INFO mapred.FileInputFormat: Total input files to process : 50
23/07/25 01:25:08 INFO mapreduce.JobSubmitter: number of splits:50
23/07/25 01:25:08 INFO Configuration.deprecation: yarn.resourcemanager.system-me
trics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publishe
r.enabled
23/07/25 01:25:09 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_16
69743171306_4600
23/07/25 01:25:09 INFO impl.YarnClientImpl: Submitted application application_16
69743171306_4600
23/07/25 01:25:09 INFO mapreduce.Job: The url to track the job: http://msba-hado
op-name:8088/proxy/application_1669743171306_4600/
```

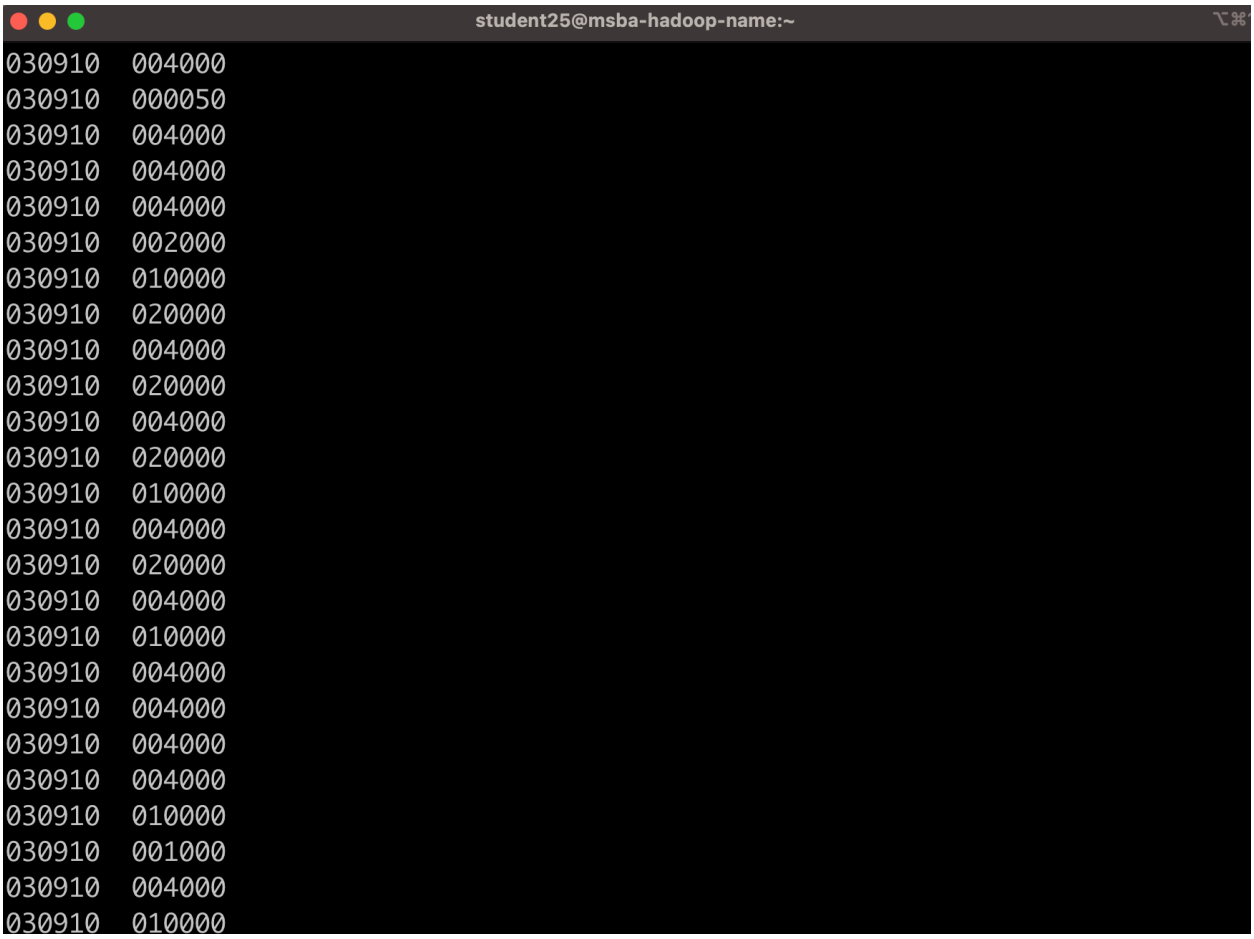
```
                Reduce input records=36305
                Reduce output records=36305
                Spilled Records=72610
                Shuffled Maps =50
                Failed Shuffles=0
                Merged Map outputs=50
                GC time elapsed (ms)=12072
                CPU time spent (ms)=34880
                Physical memory (bytes) snapshot=17378025472
                Virtual memory (bytes) snapshot=154169008128
                Total committed heap usage (bytes)=16395534336
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=415313
        File Output Format Counters
                Bytes Written=508270
23/07/25 01:26:28 INFO streaming.StreamJob: Output directory: /home/25student25/
Projectoutputpart3/
[student25@msba-hadoop-name ~]$
```

Step 5 : Print out the content of the output:

hdfs dfs -ls /home/25student25/Projectoutputpart3/

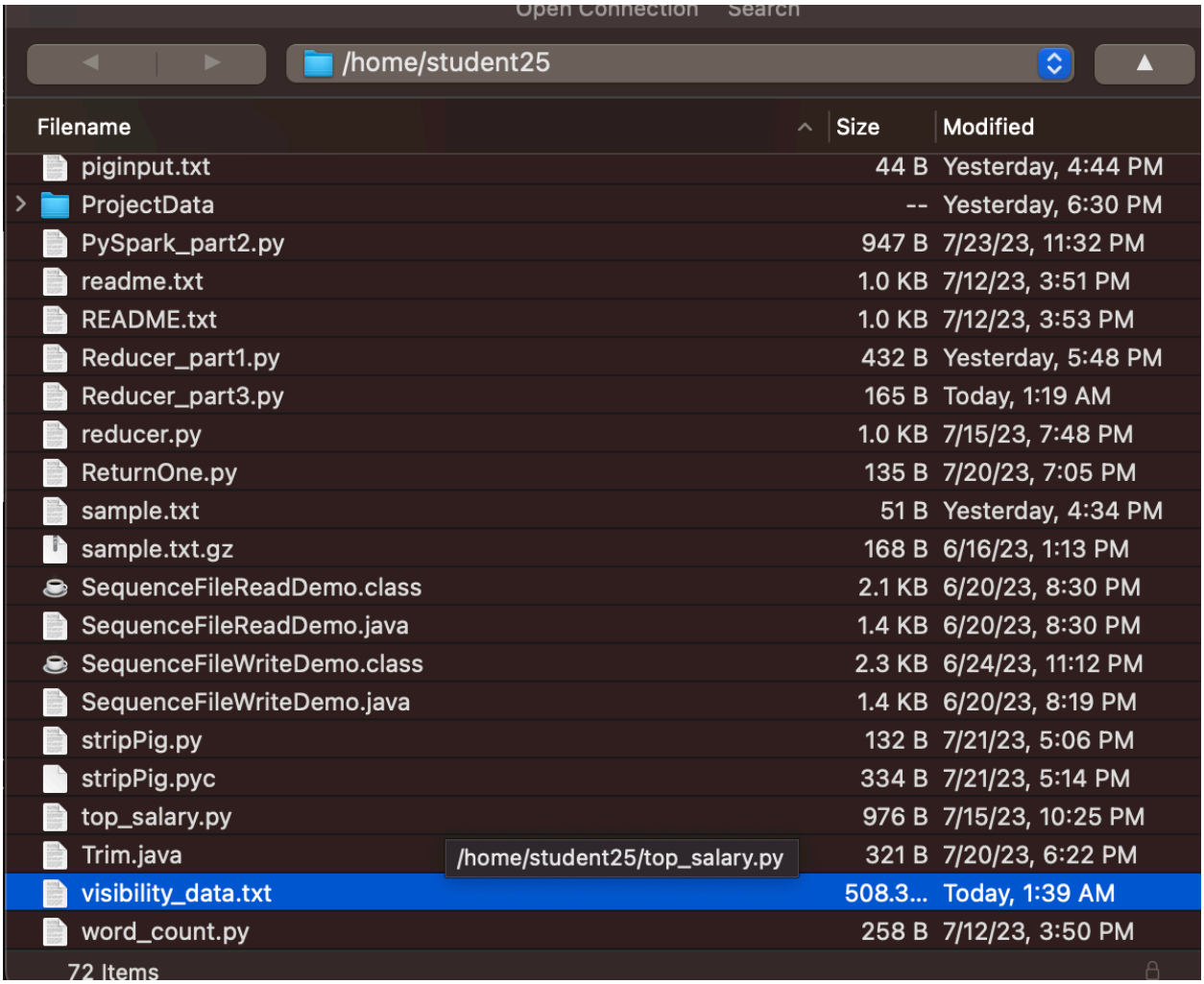
hdfs dfs -cat /home/25student25/Projectoutputpart3/part-00000

```
[student25@msba-hadoop-name ~]$ hdfs dfs -ls /home/25student25/Projectoutputpart
3/
Found 2 items
-rw-r--r--   5 student25 supergroup          0 2023-07-25 01:26 /home/25student2
5/Projectoutputpart3/_SUCCESS
-rw-r--r--   5 student25 supergroup     508270 2023-07-25 01:26 /home/25student2
5/Projectoutputpart3/part-00000
[student25@msba-hadoop-name ~]$ hdfs dfs -cat /home/25student25/Projectoutputpar
t3/part-00000
```

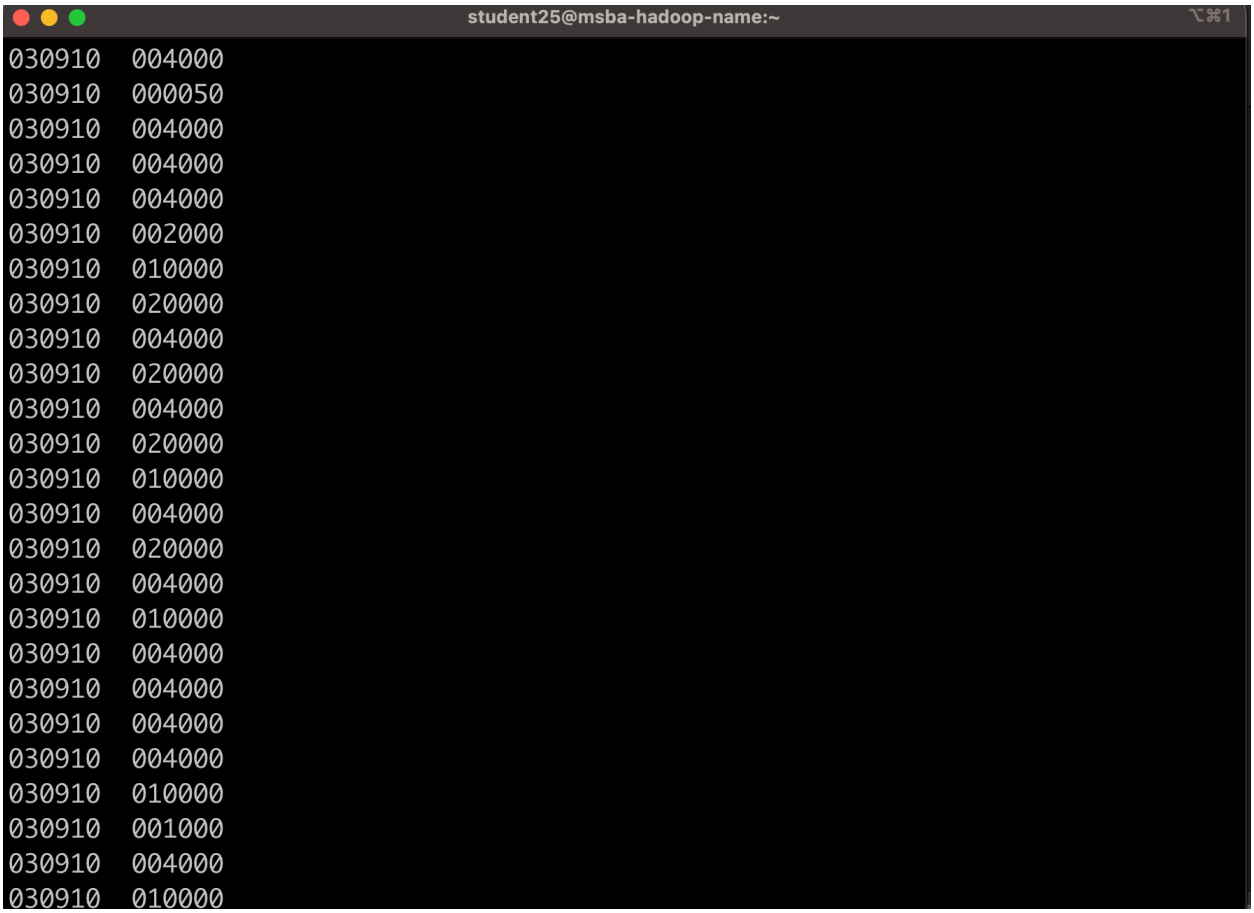


Step 6 : Command to copy output data file from HDFS to local and save into text format

hdfs dfs -copyToLocal /home/25student25/Projectoutputpart3/part-00000 /home/student25/visibility_data.txt



cat /home/student25/visibility_data.txt



Part 4:

Load the text file into Pig and get the range of *visibility distance* for each USAF weather station ID.

Load the text file into Hive and get the average *visibility distance* for each USAF weather station ID.

Step 1 : Connect to Pig

pig -x local

Step 2: Execute the commands

--Loading the 'visibility_data.txt' into pig using the below command.

records = LOAD 'visibility_data.txt'

AS (usaf_id:chararray, visibility_dist:int);

```
DESCRIBE records;

DUMP records;

grouped_records = GROUP records BY usaf_id;

DUMP grouped_records;

DESCRIBE grouped_records;


max_min_vis = FOREACH grouped_records {
   max_dist= MAX(records.visibility_dist);
   min_dist = MIN(records.visibility_dist);
   GENERATE group AS usaf_id, max_dist- min_dist AS visibility_range;
}


DUMP max_min_vis;
```

```
Input(s):
Successfully read 36305 records from: "file:///home/student25/visibility_data.txt"

Output(s):
Successfully stored 15 records in: "file:/tmp/temp107592874/tmp-1991639514"

Counters:
Total records written : 15
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local1019366017_0003


2023-07-25 19:32:38,026 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=Job
Tracker, sessionId= - already initialized
2023-07-25 19:32:38,027 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=Job
Tracker, sessionId= - already initialized
2023-07-25 19:32:38,028 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=Job
Tracker, sessionId= - already initialized
2023-07-25 19:32:38,030 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2023-07-25 19:32:38,031 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Inste
ad, use dfs.bytes-per-checksum
2023-07-25 19:32:38,031 [main] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2023-07-25 19:32:38,041 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2023-07-25 19:32:38,041 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process
: 1
(011060,49800)
(012620,50000)
(014030,50000)
(014270,49500)
```

```
job_local1019366017_0003


2023-07-25 19:32:38,026 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=Job
Tracker, sessionId= - already initialized
2023-07-25 19:32:38,027 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=Job
Tracker, sessionId= - already initialized
2023-07-25 19:32:38,028 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=Job
Tracker, sessionId= - already initialized
2023-07-25 19:32:38,030 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2023-07-25 19:32:38,031 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Inste
ad, use dfs.bytes-per-checksum
2023-07-25 19:32:38,031 [main] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2023-07-25 19:32:38,041 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2023-07-25 19:32:38,041 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process
: 1
(011060,49800)
(012620,50000)
(014030,50000)
(014270,49500)
(023610,50000)
(028360,0)
(028970,0)
(029110,0)
(029350,0)
(029700,0)
(030910,49950)
(032620,20000)
(033020,50000)
(034970,50000)
(038040,49950)
```

Step 3 Exit Pig

quit

```
grunt> quit
2023-07-25 02:00:03,953 [main] INFO  org.apache.pig.Main - Pig script completed
in 6 minutes, 4 seconds and 291 milliseconds (364291 ms)
[student25@msba-hadoop-name ~]$
```

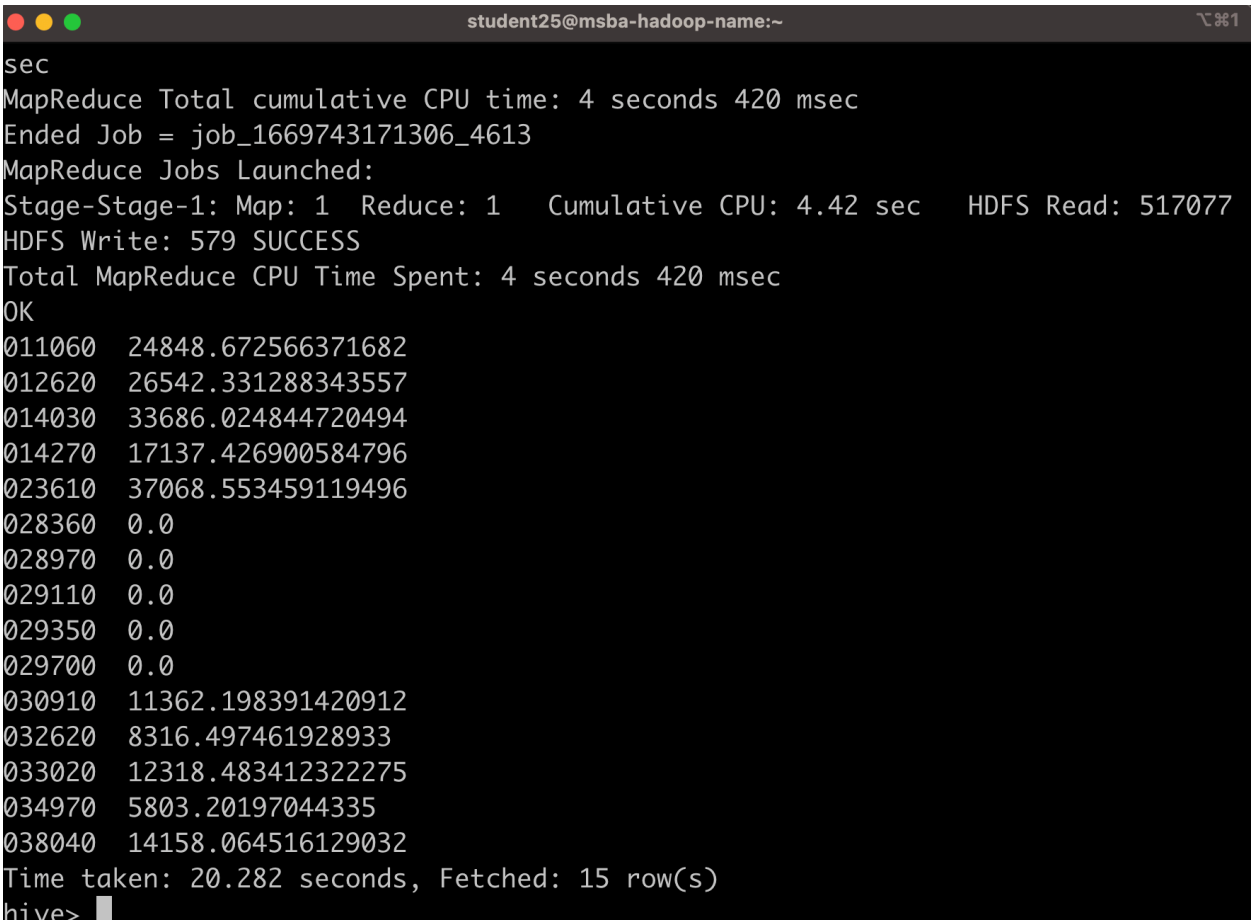
Step 4 Connect to Hive

hive

```
[student25@msba-hadoop-name ~]$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive-2.3.2/lib/log4j-slf4j-impl-2.6
.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.9.0/share/hadoop/common/li
b/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/usr/local/hive-2.3.2/lib/hi
ve-common-2.3.2.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versio
ns. Consider using a different execution engine (i.e. spark, tez) or using Hive
1.X releases.
hive>
```

Step 5 : Execute the commands on hive

```
DROP TABLE IF EXISTS avg_visibility25;

CREATE TABLE  avg_visibility25 (usaf_id STRING,visibility_dist INT)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t' ;


LOAD DATA LOCAL INPATH 'visibility_data.txt'
OVERWRITE INTO TABLE avg_visibility25;

SELECT usaf_id, AVG(visibility_dist)

FROM avg_visibility25

GROUP BY usaf_id;
```


student25@msba-hadoop-name:~
sec
MapReduce Total cumulative CPU time: 4 seconds 420 msec
Ended Job = job_1669743171306_4613
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1   Cumulative CPU: 4.42 sec   HDFS Read: 517077
HDFS Write: 579 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 420 msec
OK
011060 24848.672566371682
012620 26542.331288343557
014030 33686.024844720494
014270 17137.426900584796
023610 37068.553459119496
028360 0.0
028970 0.0
029110 0.0
029350 0.0
029700 0.0
030910 11362.198391420912
032620 8316.497461928933
033020 12318.483412322275
034970 5803.20197044335
038040 14158.064516129032
Time taken: 20.282 seconds, Fetched: 15 row(s)
hive>