Task 1:

How to run:

Jar is uploaded on S3.

Large file is uploaded on S3

Add the following step to the cluster:

**JAR location :**s3://nitin-assignment2/hw2-task1-0.2-SNAPSHOT-jar-with-dependencies.jar

**Main class :**None

**Arguments :**s3://metcs755/taxi-data-sorted-large.csv.bz2 s3://nitin-assignment2/task1/gpsErrorsLargejob1 s3://nitin-assignment2/task1/gpsErrorsLargejob2

**Action on failure:**Continue

Description of the mapreduce job:

The hadoop mapreduce job contains 2 jobs: 1st job calculates the number of errors rate at each hour of the day. 1st job produces multiple output file split by different hours of the day. 2nd job combines the results from all files from 1st job and produces 1 file that is sorted in desceding order of most errors.

The entire mapreduce job took 5 minutes to run.

Results (sorted in descending order):

19 179546

18 174809

20 168344

21 167194

22 160924

17 149571

14 147635

15 145712

23 142821

13 140292

12 139394

11 131993

09 130800

08 127918

16 127443

10 127126

00 116055

07 108479

01 89526

02 69277

06 67436

03 54444

04 42140

05 36306

Task 2:

How to run:

Jar is uploaded on S3.

Large file is uploaded on S3

Add the following step to the cluster:

**JAR location :**s3://nitin-assignment2/hw2-task2-0.2-SNAPSHOT-jar-with-dependencies.jar

**Main class :**None

**Arguments :**s3://metcs755/taxi-data-sorted-large.csv.bz2 s3://nitin-assignment2/task1/job1 s3://nitin-assignment2/task2/job2

**Action on failure:**Continue

Description:

The hadoop mapreduce job contains 2 jobs: 1st job calculates the error rate for each Taxi. It then outputs multiple files each containing the top 5 taxis with the most error rate for the input split that it processed. 2nd job combines the results from all files from 1st job, sorts them in desceding order of the highest error rate and produces 1 file that contains the top 5 taxis with the most error rates in all of the data given.

The entire mapreduce job took 6 minutes to run.

Results (sorted in descending order):

922B85420339480041D664134572C94C 100.0

36F5EE6F1888BBA3D85DF33B25EC3EBC 100.0

165F05ACA6203A8F38C306AD114E2C05 100.0

CF613B8747B135B52F9E36C13AF79769 100.0

0EE3FFCBDFD8B2979E87F38369A28FD9 100.0

Task 3:

How to run:

Jar is uploaded on S3.

Large file is uploaded on S3

Add the following step to the cluster:

**JAR location :**s3://nitin-assignment2/hw2-task2-0.2-SNAPSHOT-jar-with-dependencies.jar

**Main class :**None

**Arguments :**s3://metcs755/taxi-data-sorted-large.csv.bz2 s3://nitin-assignment2/task1/job1 s3://nitin-assignment2/task2/job2

**Action on failure:**Continue

Description:

The hadoop mapreduce job contains 2 jobs: 1st job calculates the money per minute for each driver. It then outputs multiple files each containing the top 5 drivers with the most error rate for the input split that it processed. 2nd job combines the results from all files from 1st job, sorts them in desceding order of most money per minute amde and produces 1 file that contains the top 5 drivers that made the highest money per minute in all of the data given.

The entire mapreduce job took 8 minutes to run.

Results (sorted in descending order):

324D6935EC71A4FDE749BF9924F3AE63 189.75

6CDD60EA0045EB7A6EC44C54D29ED402 127.0

087408522C31EEB1F982BC0EAF81D35F 126.375

49030E99DC6676FA7A0AE152CE0C68C7 114.0

E10ADC3949BA59ABBE56E057F20F883E 112.45999908447266