ECS765P-BIG DATA PROCESSING-2021/22

COURSEWORK:ETHEREUM ANALYSIS(40%)

## **PART A. TIME ANALYSIS (20%)**

1:-JOB ID: <http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1649894236110_4902/>

In this question, a bar plot is created to represent the count of transactions that occurred each month between the start and finish of the data collection. For this ,first imported all of the necessary libraries including mrjob.

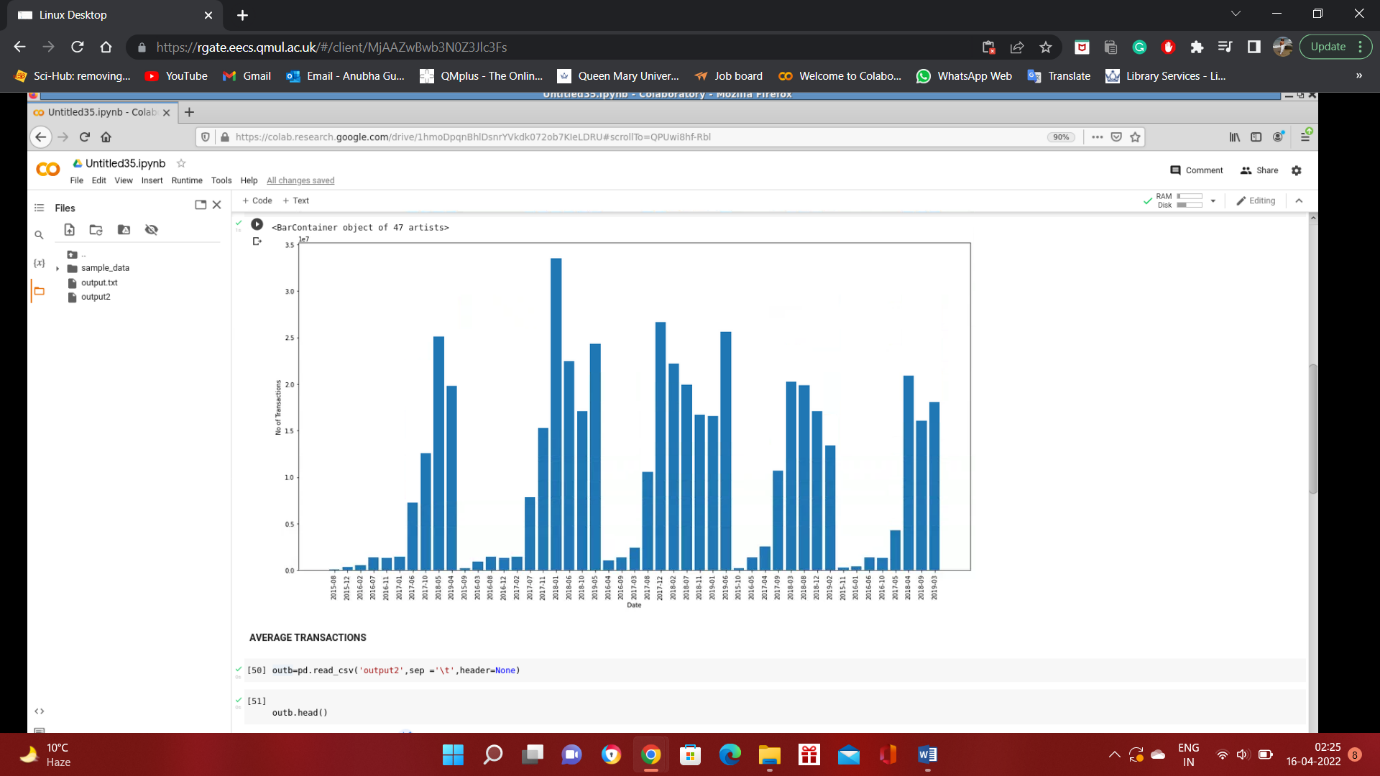
The transaction dataset is used ,from which timestamp is first extracted and then converted in a time format ie year and month. The mapper yield the year and month as key and 1 as value for each date(key).

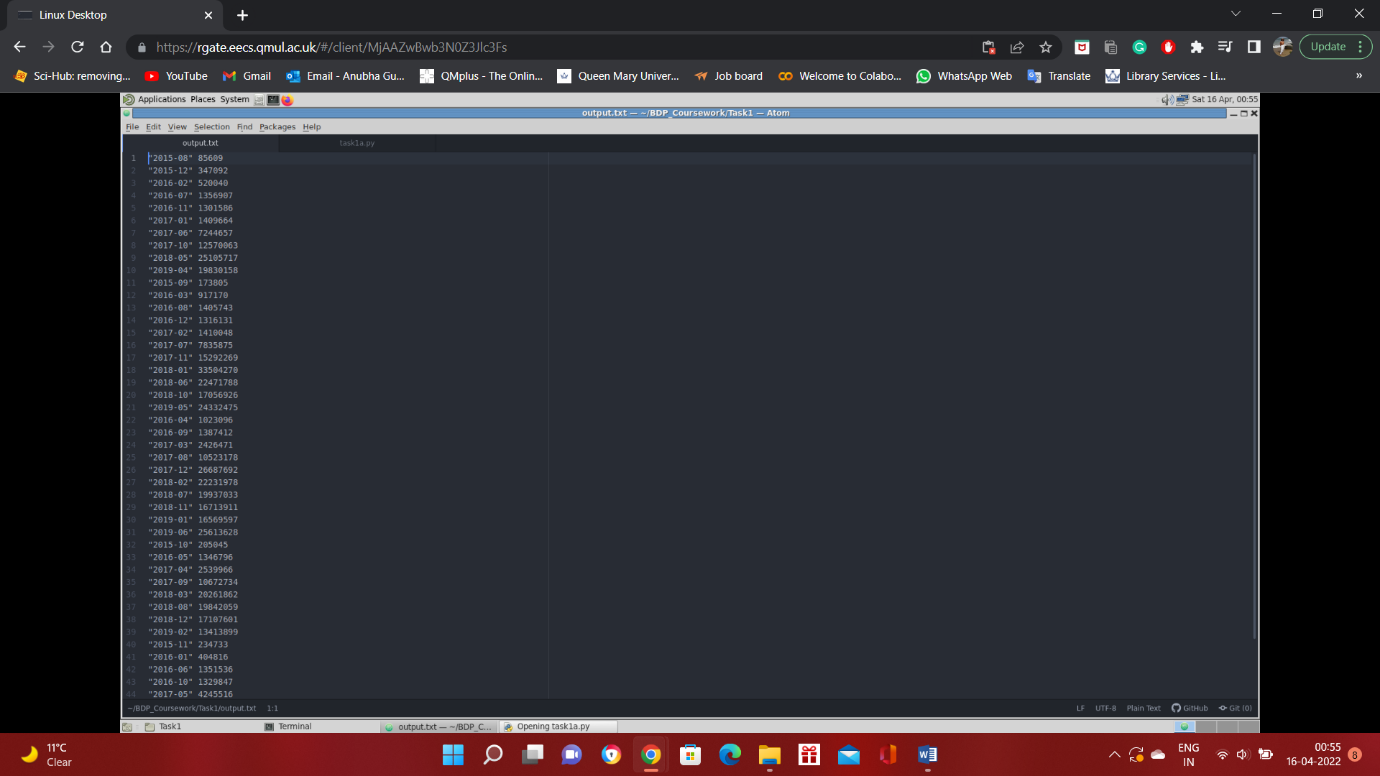
The combiner add all the values(count the transactions) for a key( date ,time).

The reducer does the same step and yield the date with no.of transaction in each date

The text file is further used to create a plot using python. All the libraries are imported and then with matplotlib ,plot is plotted in which x axis is date and y axis is no of transactions.

The plot obtained for all the transaction in each month for all the years.



Hadoop Output:-Date ,Count

2: JOB ID: -<http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1649894236110_4912/>

For this problem, average transaction in each month for all the timestamp in the transaction dataset was found.

The libraries are imported and the code in python and is executed using mrjob Hadoop.The timestamp from transaction table is used which is further converted into regular year and month format ,for transaction average,transaction value is used from the table.

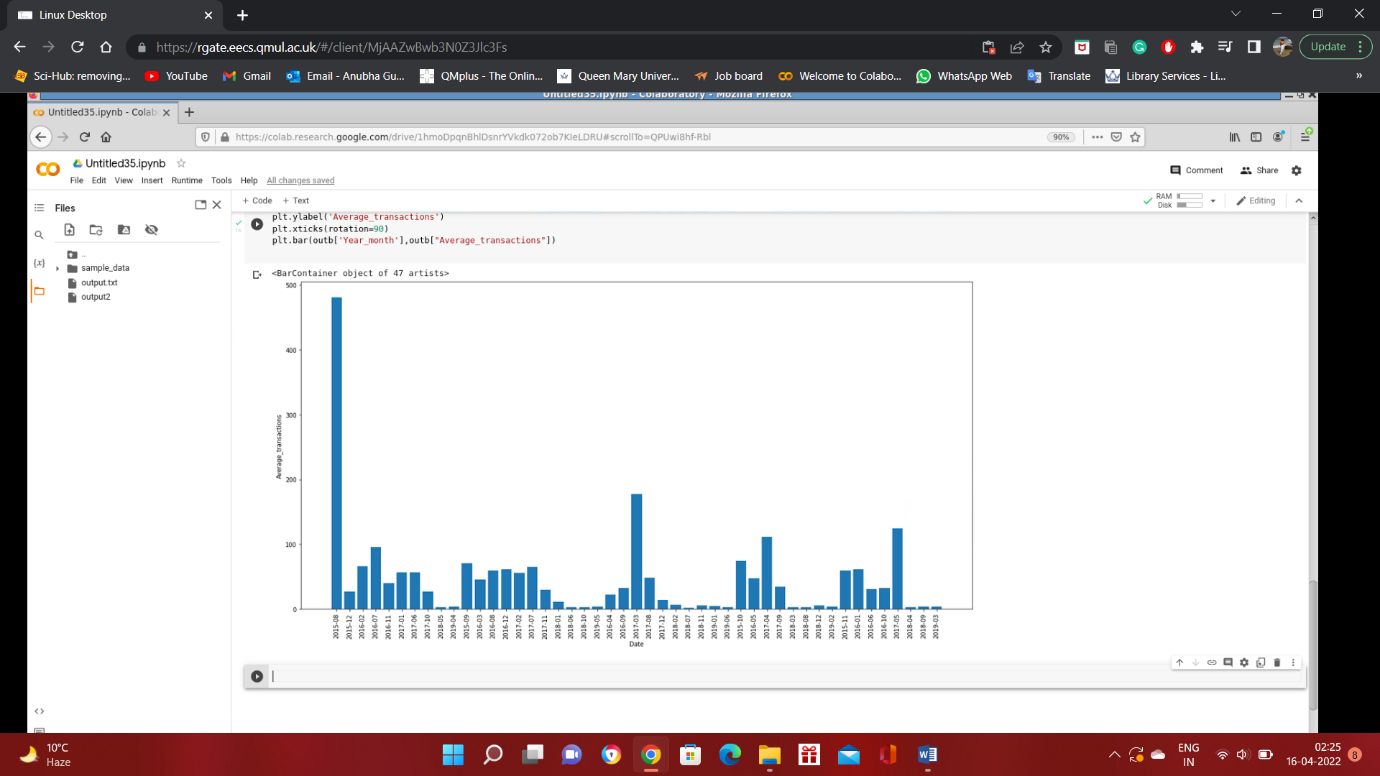
The mapper yields year and month as key and transaction value along with count as value

The combiner is used to combine the transaction for each month of the year along with the counter which can be used at the reducer to take the average of the transaction amount.

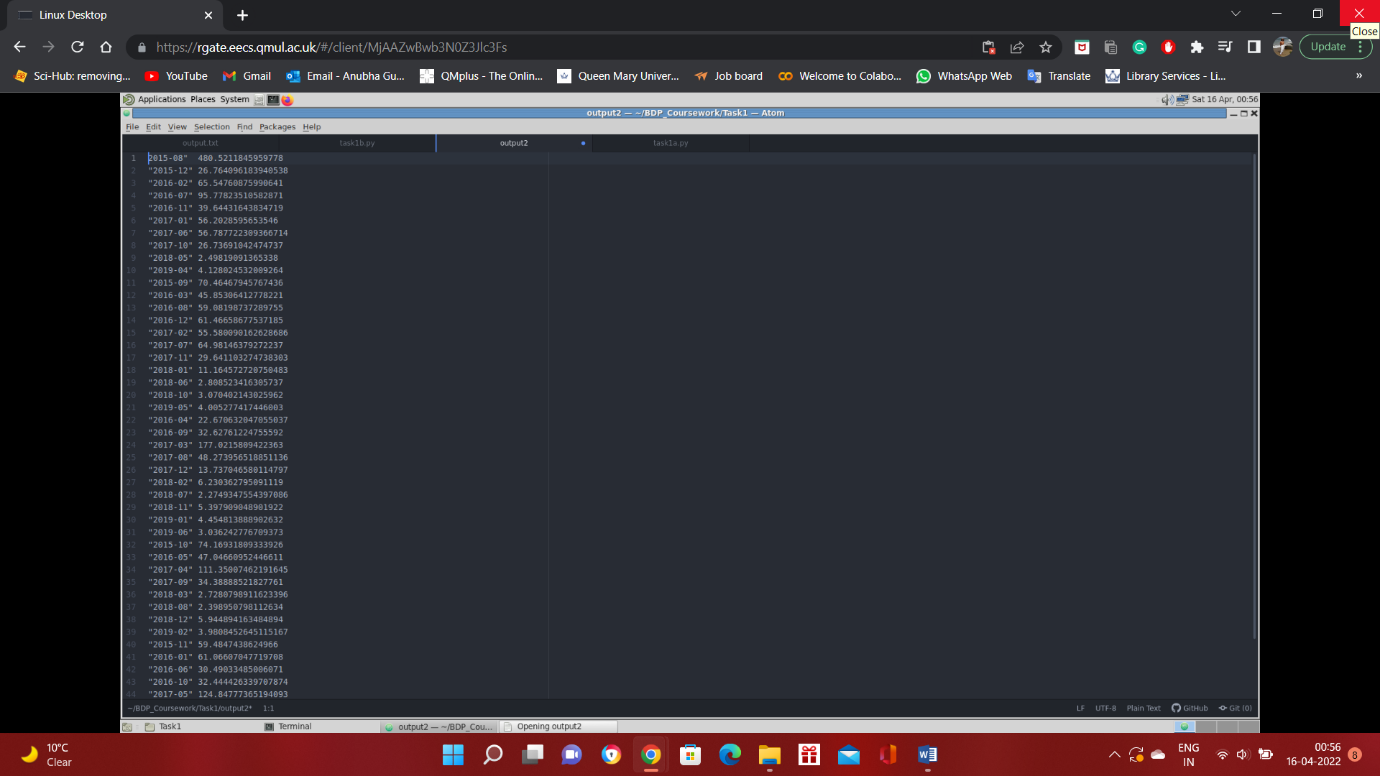
The combiner yields key and result which is year –month and count,transaction amount total for the month.

Reducer will perform the summation of the transaction for each month along with count (to count the total transaction) and at last evaluate the average and store the result in ‘average’ and yield key-year ,month and value-average

The output is a txt file contains the date and average transaction for each date(year and month)

The plot is successful with the use of txt output file .

Hadoop output:- Date ,Average



## **PART B. TOP TEN MOST POPULAR SERVICES (25%)**

JOB ID:- <http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1649894236110_5911/>

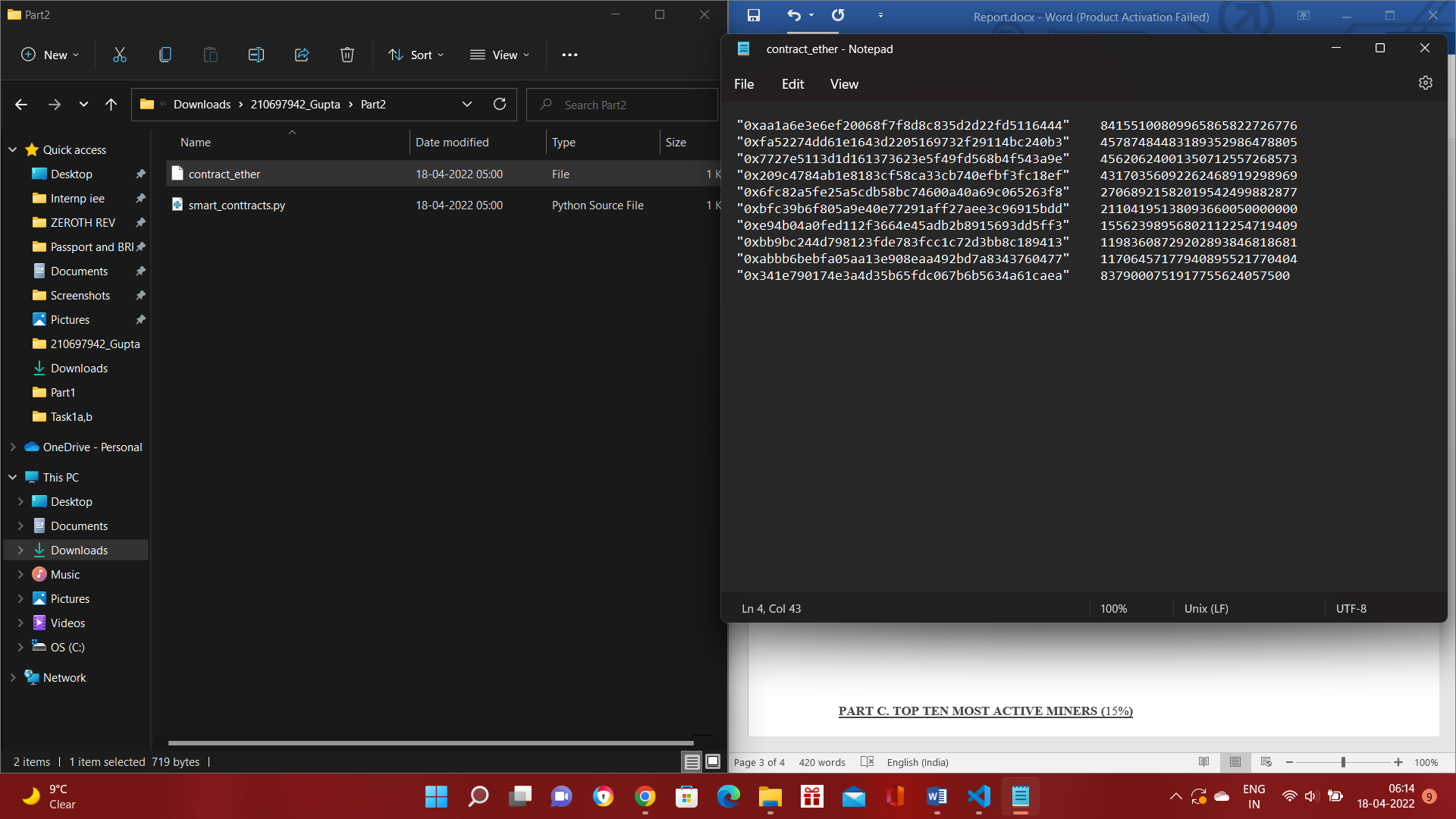
To solve the problem ,the dataset used are Transaction and contracts. MRStep library is used in order to run multiple jobs.

First condition is set if the length of the input is 7 ,The receiver address is extracted along with the value from the transaction table. If the input is 5 , the Contract table ,address is extracted from the contracts dataset.

For the transaction ,all the values are added together for the address and atlas for the first reducer a condition is set,upon satisfying the condition ,the reducer yields only that transaction address which is contracts (which is smart contract) and the total value is extracted for the particular address.

Second job will take the input .Mapper 2, yield the value as address and total value ,it will be the input for the reducer and the reducer sort the values that’s yielded and obtain 10 contracts with their values.

Hadoop output:- Address, Ether



## **PART C. TOP TEN MOST ACTIVE MINERS (15%)**

JOB ID: <http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1649894236110_6060/>

To find the top miners ,the dataset used is blocks. The blocks dataset was aggregated to observe the miners involvement.

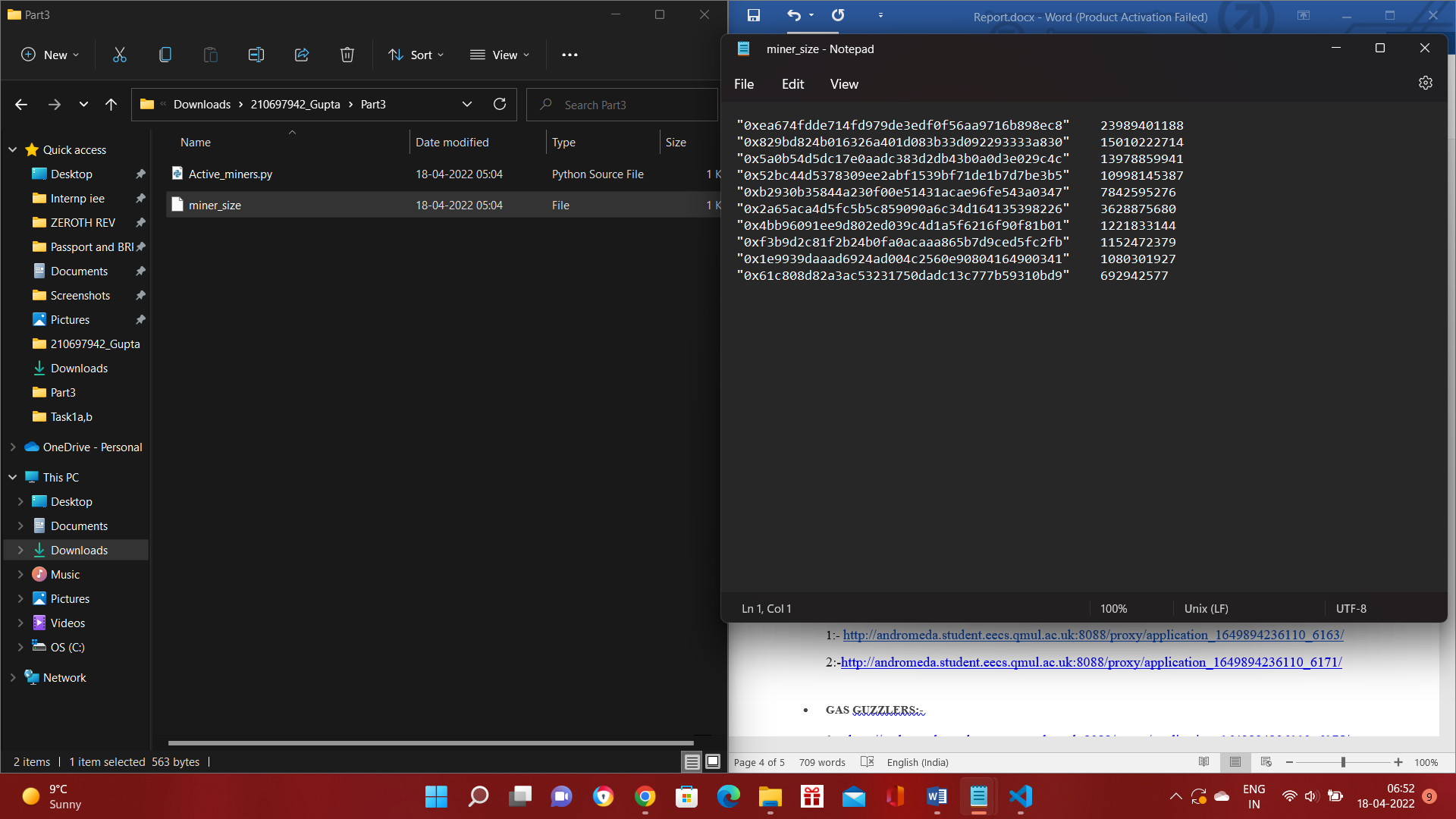
For this problem, multiple jobs were executed using MR Step .The miner address column is used along with the size column from the block dataset. The code is processed using MR Job Hadoop.

The first mapper yield the address and the size for each miner. The reducer for the first job takes the key and value ,ie. Miner address and the size and sum all the size for each miner.

The Second mapper takes the output of first reducer and yield the value by paring th input’s key and value.

The second reducer takes the values and performs the sort method inorder to yield the top 10 miners along with their aggregated size.

Hadoop Output:- Miners , Size



## **PART D. DATA EXPLORATION (40+%)**

### **SCAM ANALYSIS:- Popular Scams**

### JOB ID <http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1649894236110_6128/>

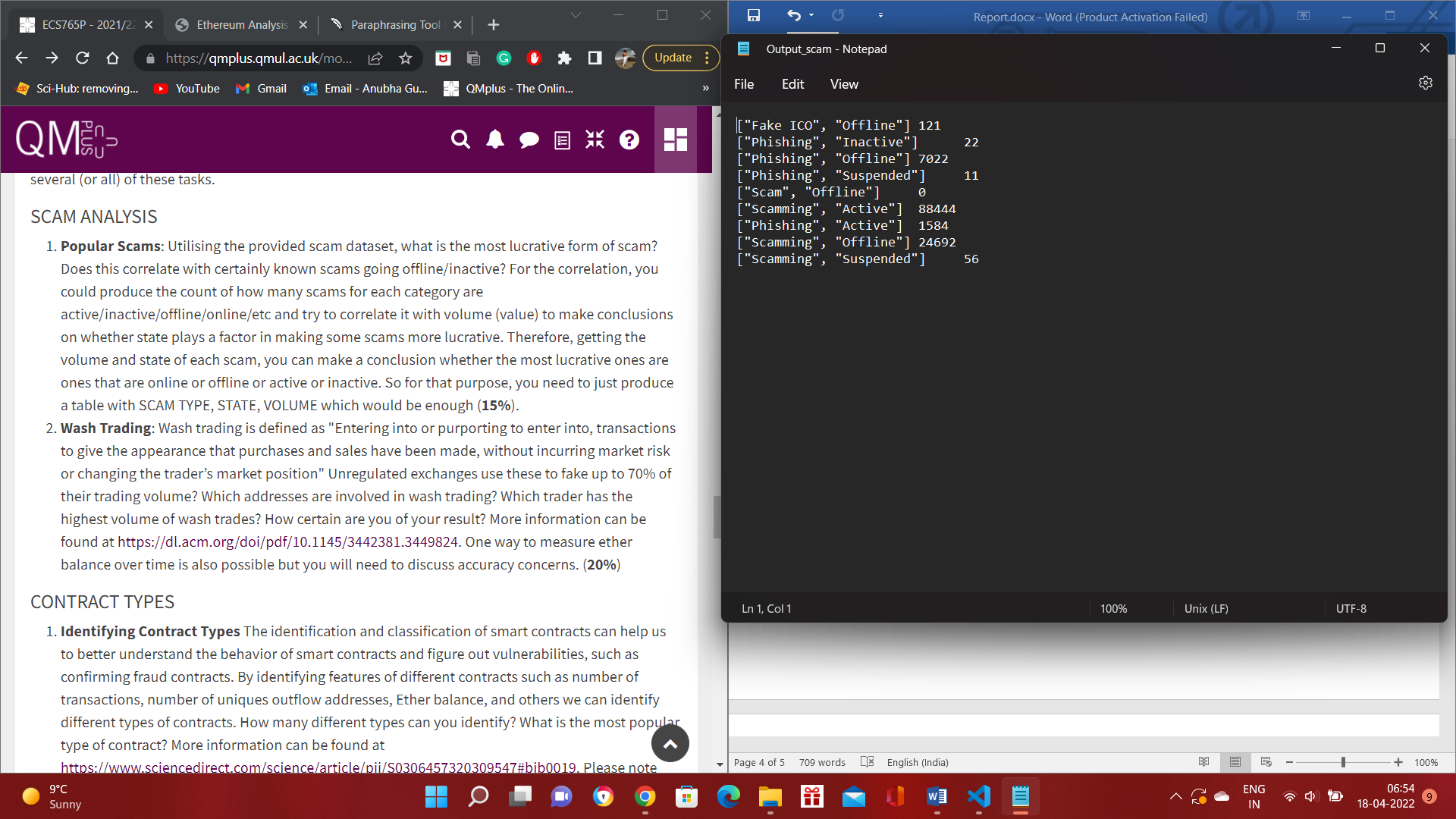
The problem is by utilising the scam dataset finding the most lucrative form of scam and finding the correlation with status of the scam by providing the total volume for each scams with different status.For this ,the two datsets are used,transaction.csv and scams.json.The question is solved using mrjob Hadoop.

The code process in such a way. The mapper takes the input and for each input checks the length of it and according to the length further processing is done.If the length is 7 ie transaction dataset and transaction address is collected as key and for scam ,as the file is in json format ,it uses json.load to load the input and from scam dataset category,address and status is yield.

The first reducer will that the input and perform some conditional loops to get the address from the transaction dataset which is affected by the scams ie whose address is present in the scam dataset.Once the check is processed ,it yield category of the scam ,the status and the count as key and value simultaneously

The second mapper takes the input and yield the key and value as it is ,at last the final reducer takes the key and value and sun the total count for each category and the status of the scam.

The Hadoop output:- Category , Status ,Value



Scamming is the most lucrative type of scam ,which is active and has affected 88444 addresses.

Based on the output ,it can be assume that the lucrative form of scam is correlated to its status as the scamming which is still active already affected a lot of users.The most lucrative ones are active ,so the scam status play a important role in analysing the threat.

### **MISCELLANEOUS ANALYSIS**

**FORK THE CHAIN**

1:- <http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1649894236110_6163/>

For finding the forks ,first the dataset used is blocks.As if the two miner is present in a block number ,fork is present.By using this analysis ,fork date is found out by first using the block dataset ,accessing the miner ,block number and date value for the input.The mapper yields the block number as a key with the miner and date as a value.

The reducer will perform the conditional loop for the key and value ,received as an input.

If the block number has 2 or more than 2 miner ,get the date,using this logic the date for fork is found.

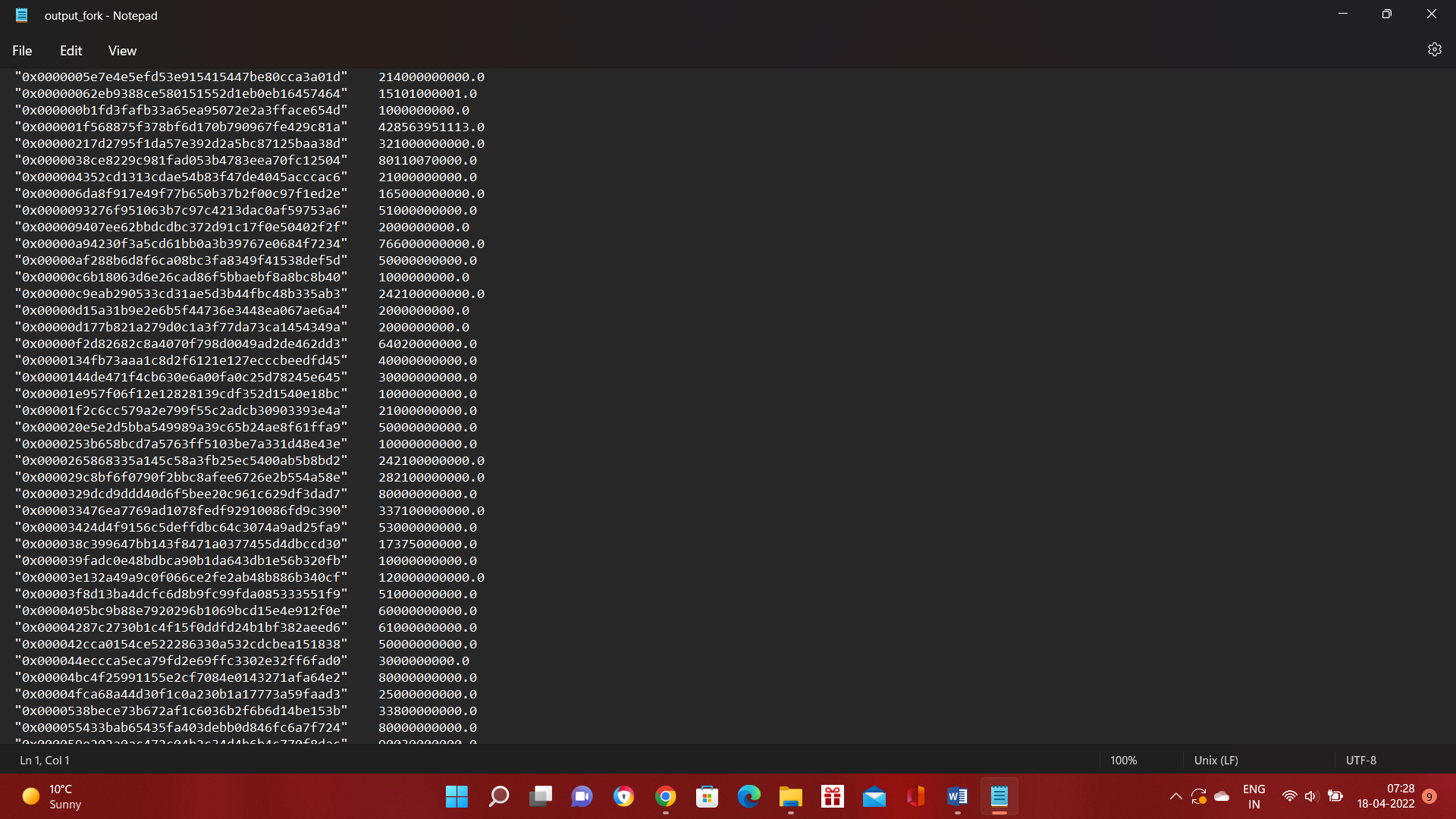
Output for the job:-

2017-12

2:-<http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1649894236110_6171/>

Using the date found out by the first step and further performing can be done.To excess the details of address benefited from the fork.For this date,address,price column value is yielded and reducer performs the sorting onto the input values and sort it from high to low (price) for each address.

Output of the job:-



Particular address:- 0x0000000000b3f879cb30fe243b4dfee438691c04" 29466184559989.0

Profited most from the fork

:-Based on the output ,it is seen that after the fork ie 2017-12 ,there has been decrease in the price.

**GAS GUZZLERS:-**

<http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1649894236110_6175/4>

For analysing the change in the gas price over time ,transaction dataset is used.The gas price and the date are yield from the mapper.

The combiner takes the input key and value ,ie date and price, and for each date ,sum the price and count for average gas price over the months. The yielded output goes to the reducer ,where again the prices and the count are summed for the date and then the average of the price for the date is yielded as the value and the date as key.

The output ,gas\_price.txt is further used to plot and visualize the pattern of the gas price with the date.

The plot shows that the average price of gas has decreased from 2017 to 2019 and each year there a spike can be seen in the prices in the few months of the year.

2:-<http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1649894236110_6176/>

For analysing the change in the gas over time ,transaction dataset is used.The gas and the date are yield from the mapper.

The combiner takes the input key and value ,ie date and price, and for each date ,sum the price and count for average gas over the months. The yielded output goes to the reducer ,where again the prices and the count are summed for the date and then the average of the gas for the date is yielded as the value and the date as key.

The output ,gas.txt is further used to plot and visualize the pattern of the gas with the date to analyse the change in gas for the contracts.

Output:-

For gas,it is observed that it has high demand before 2017-12 but after the fork ,it became steady .

**COMPARATIVE EVALUATION**

In this question, used the spark code to complete the partB.

As ,map reduce task was performed in PartB, for this task using the Transactions and Contracts dataset .

To begin, import the pyspark module. Then, much like in the map reduce code, write a function to verify the fields: if the field is not equal to 7, we return false for the transaction dataset, and if the value of the fields is not equal to 5, we return false for the contracts dataset.

Then, using the aforesaid functions, load the dataset and examine the fields. Then used the address as a key and the value as a value.At the reduce step, with the key being the same but all the values for the keys aggregating. Mapping the address as the key for contracts. Then a join operation is done between this address and the aggregate values obtained from the preceding reducer's calculation on the transaction dataset.

Result:-

Both produces same output.

The following advantages of utilising Spark over MapReduce were discovered after implementing the application in both MapReduce and Spark:

Less code - The Spark application is about 15 lines long, compared to the MRJob framework on Hadoop, which is around 50 lines long.

Faster execution — The tasks were run five times each on Hadoop and Spark.The primary difference between the Hadoop and Spark tasks is that the Hadoop job was really two MapReduce processes combined into one, and hence Spark was quicker because it didn't write the intermediate results to HDFS.