Task 2 (25 points) Ethereum dataset

In task 2, you will analyze the Ethereum dataset using Spark, covering transactions from August 2015 to January 2019. There are many resources available to help you understand Ethereum and blockchain technology. **<u>While it is not required but</u>** you can access an explanation of Ethereum through https://www.youtube.com/watch?v=1Hu8lzoi0Tw and learn more about blockchain through https://www.youtube.com/watch?v=SSo_EIwHSd4&t=40s Additionally, feel free to explore other resources further to enhance your understanding of Ethereum and blockchain technology.

The dataset consists of a blocks data file named **"blocks.csv",** and the transaction data file, named **"transactions.csv"** which are available under the path **//data-repository-but/ECS765/Ethereum/**. You can view them via **ccc method bucket ls**.

The description of each field in order for the **blocks.csv** is shown below (a sample CSV file was given to explore/visualise on your machine):

number: The block number
hash: Hash of the block
parent_hash: Hash of the parent of the block
nonce: Nonce that satisfies the difficulty target
sha3_uncles: Combined has of all uncles for a given parent
logs_bloom: Data structure containing event logs
transactions_root: Root hash of the transactions in the payload
state_root: Root hash of the state object
receipts_root: hash of the transaction receipts tree
miner: The address of the beneficiary to whom the mining rewards were given
difficulty: Integer of the difficulty for this block
total_difficulty: Total difficulty of the chain until this block
size: The size of this block in bytes
extra_data: Arbitrary additional data as raw bytes
gas_limit: The maximum gas allowed in this block
gas_used: The total used gas by all transactions in this block
timestamp: The timestamp for when the block was collated
transaction_count: The number of transactions in the block

base_fee_per_gas: Base fee value

The table below shows a few records from the **blocks.csv** file, displaying a subset of its fields. For some fields, the full values are omitted due to their length, but you can refer to the provided sample dataset for the complete values.

hash	miner	difficulty	size	gas_limit	gas_used	time_stamp	transaction_count
0x91…	0x5a…	1.76566E+15	9773	7995996	2042230	1513937536	62
0x1f…	0xea…	1.76738E+15	15532	8000029	4385719	1513937547	101
0x2e…	0x82…	1.76824E+15	14033	8000029	7992282	1513937564	99

The description of each field in order for the **transactions.csv** is shown below (a sample CSV file was given to explore/visualise on your machine):

hash: Hash of the block

nonce: Nonce that satisfies the difficulty target

block_hash: Hash of the block where the transaction is in

block_number: Block number where this transaction was in

transaction_index: Transactions index position in the block.

from_address: Address of the sender

to_address: Address of the receiver. null when it is a contract creation transaction

value: Value transferred in Wei (the smallest denomination of ether)

gas: Gas provided by the sender

gas_price : Gas price provided by the sender in Wei

input: Extra data for Ethereum functions

block_timestamp: Timestamp the associated block was registered at (effectively timestamp of the transaction)

max_fee_per_gas: Sum of base fee and max priority fee

max_priority_fee_per_gas: Tip for mining the transaction

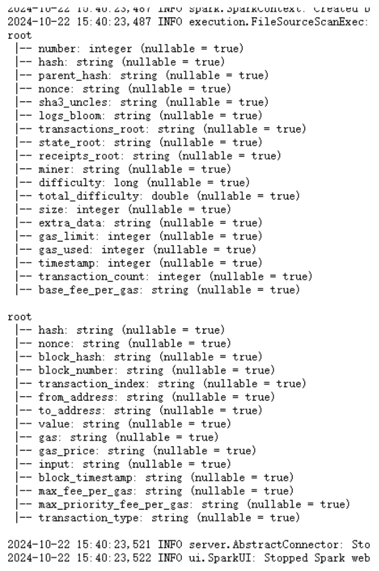
transaction_type: Value used to indicate if the transaction is related to a contract or other specialised transaction

The table below shows a few records from the **transactions.csv** file, displaying a subset of its fields. For some fields, the full values are omitted due to their length, but you can refer to the provided sample dataset for the complete values.

hash	block_hash	transaction_index	from_address	to_address	gas	gas_price	block_timestamp
0x8d…	0x99…	2	0x4b…	0xdc…	41000	50000000	1474174806
0x8b…	0xdd…	0	0x2b…	0x06…	21000	50000000	1474174820
0x8f…	0x0c…	1	0x45…	0x4e…	50000	50000000	1474174836

Please note that the example tables/pictures provided in this task are based on my results. Your results may differ depending on how you process and display the data. When a screenshot is required, in addition to the results, you must also include your running timestamp in your screenshot. A running timestamp is the recorded time (printed at the beginning of each line in the logs) during the execution of a Spark job; you can capture the running timestamp in the logs. The example tables/pictures provided in each question illustrate the expected format (including the running timestamp and your results) for your screenshot.

1) **(2 points)** Load **blocks.csv** and **transactions.csv.** You need to include a screenshot of the resulting schema for each file in your report. For example:



```
2024-10-22 15:40:23,487 INFO spark.SparkContext: Created b
2024-10-22 15:40:23,487 INFO execution.FileSourceScanExec:
root
 |-- number: integer (nullable = true)
 |-- hash: string (nullable = true)
 |-- parent_hash: string (nullable = true)
 |-- nonce: string (nullable = true)
 |-- sha3_uncles: string (nullable = true)
 |-- logs_bloom: string (nullable = true)
 |-- transactions_root: string (nullable = true)
 |-- state_root: string (nullable = true)
 |-- receipts_root: string (nullable = true)
 |-- miner: string (nullable = true)
 |-- difficulty: long (nullable = true)
 |-- total_difficulty: double (nullable = true)
 |-- size: integer (nullable = true)
 |-- extra_data: string (nullable = true)
 |-- gas_limit: integer (nullable = true)
 |-- gas_used: integer (nullable = true)
 |-- timestamp: integer (nullable = true)
 |-- transaction_count: integer (nullable = true)
 |-- base_fee_per_gas: string (nullable = true)

root
 |-- hash: string (nullable = true)
 |-- nonce: string (nullable = true)
 |-- block_hash: string (nullable = true)
 |-- block_number: string (nullable = true)
 |-- transaction_index: string (nullable = true)
 |-- from_address: string (nullable = true)
 |-- to_address: string (nullable = true)
 |-- value: string (nullable = true)
 |-- gas: string (nullable = true)
 |-- gas_price: string (nullable = true)
 |-- input: string (nullable = true)
 |-- block_timestamp: string (nullable = true)
 |-- max_fee_per_gas: string (nullable = true)
 |-- max_priority_fee_per_gas: string (nullable = true)
 |-- transaction_type: string (nullable = true)

2024-10-22 15:40:23,521 INFO server.AbstractConnector: Sto
2024-10-22 15:40:23,522 INFO ui.SparkUI: Stopped Spark web
```

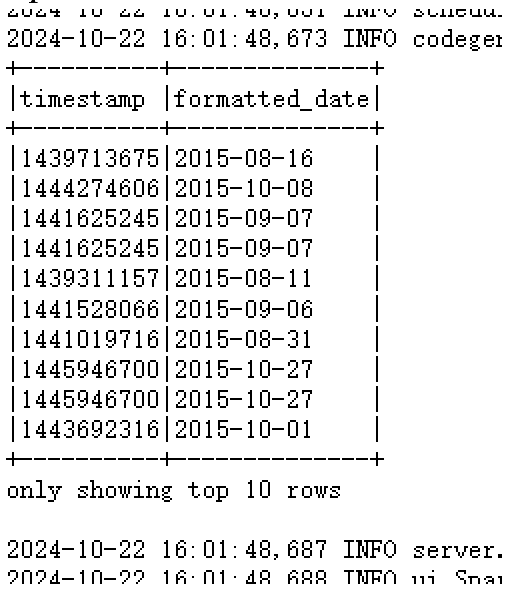
2) **(3 points)** Evaluate the top 10 miners by the total size of blocks they have mined in the blocks.csv file. You need to give a screenshot in your report. **Do not truncate the fields.** An example of the format of the results is only provided in

this question. Please ensure that you include the timestamp in your screenshot, as required.

miner	total_size
0x5a0b54d5dc17e0aadc383d2db43b0a0d3e029c4c	5
0xea674fdde714fd979de3edf0f56aa9716b898ec8	4
0x829bd824b016326a401d083b33d092293333a830	3
…….	……

Note that the above figures/values may not represent the actual result.

3) **(5 points)** The **timestamp** field of **blocks.csv** file is in UNIX format that represents the number of seconds elapsed since January 1, 1970, UTC. Therefore, you need to convert the UNIX timestamp to the **"yyyy-MM-dd"** format, excluding the specific time of day (hh-mm-ss). Create a new field to store the converted date in the "yyyy-MM-dd" format. For example, convert '1441625245' to '2015-09-07'. Select the **"timestamp"** and **"formatted_data"** fields and show 10 sample records with the **"timestamp"** and **"formatted_data"** fields. **Do not truncate the fields.** You need to include a screenshot of your results in your report.



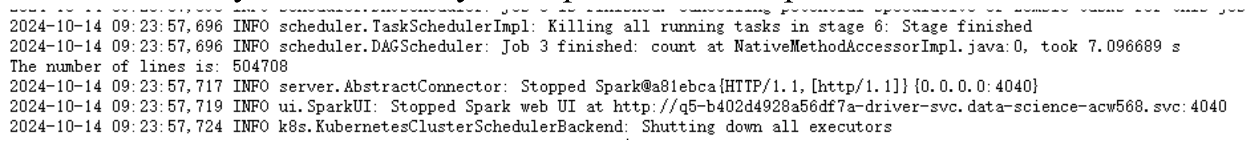
```
2024-10-22 16:01:48,673 INFO codeger
+----------+--------------+
|timestamp |formatted_date|
+----------+--------------+
|1439713675|2015-08-16    |
|1444274606|2015-10-08    |
|1441625245|2015-09-07    |
|1441625245|2015-09-07    |
|1439311157|2015-08-11    |
|1441528066|2015-09-06    |
|1441019716|2015-08-31    |
|1445946700|2015-10-27    |
|1445946700|2015-10-27    |
|1443692316|2015-10-01    |
+----------+--------------+
only showing top 10 rows

2024-10-22 16:01:48,687 INFO server.
```

Note that the above figures/values may not represent the actual result.

4) **(3 points)** Use **inner join** to merge the two files by joining the **"block_hash"** field from **"transactions.csv"** with the **"hash"** field from **"blocks.csv"**. After

merging, print the number of lines in the merged dataset. You will need to include a screenshot of your results in your report. For example:



```
2024-10-14 09:23:57,696 INFO scheduler.TaskSchedulerImpl: Killing all running tasks in stage 6: Stage finished
2024-10-14 09:23:57,696 INFO scheduler.DAGScheduler: Job 3 finished: count at NativeMethodAccessorImpl.java:0, took 7.096689 s
The number of lines is: 504708
2024-10-14 09:23:57,717 INFO server.AbstractConnector: Stopped Spark@a81ebca{HTTP/1.1, [http/1.1]}{0.0.0.0:4040}
2024-10-14 09:23:57,719 INFO ui.SparkUI: Stopped Spark web UI at http://q5-b402d4928a56df7a-driver-svc.data-science-acw568.svc:4040
2024-10-14 09:23:57,724 INFO k8s.KubernetesClusterSchedulerBackend: Shutting down all executors
```

Note that the above figures/values may not represent the actual result.

5) (**6 points**) Extract the entries in **September 2015** from the joined table to find the number of blocks produced and the number of unique senders (**from_address field**) for each day in the month. You need to produce the date field and the two required fields, then sort by date. You need to include a screenshot of your results in your report. **Do not truncate the fields.** An example of the format of the results is only provided in this question. Please ensure that you include the timestamp in your screenshot, as required.

formatted_date	**block_count**	**unique_senders_count_number**
2015-09-01	**5321**	**987**
2015-09-02	**2344**	**564**
2015-09-03	**3425**	**5435**
2015-09-04	**4324**	**547**
……	**……**	**……**

Note that the above figures/values may not represent the actual result. Your screenshot needs to include the running timestamp.

Then, you need to extract the results into a file and plot from the results **two histogram plots** with **'formatted_date'** on the x-axis and **"block_count"** and "**unique_senders_count_number**" on the y-axis, respectively. For example, assume that the **"block_count"** and "**unique_senders_count_number**" in **"2015-09-01" are 5321 and 987 separately;** in one histogram **"2015-09-01"** will be on the x-axis, indicating **5321** above the bar of **"2015-09-01"**. in another histogram **"2015-09-01"** will be on the x-axis, indicating **987** above the bar of **"2015-09-01"**. You will need to include two histograms in your report.

6) **(6 points)** Extract the entries in **October 2015** from the joined table. Calculate the total transaction fee (calculated as **gas * gas_price**) where transactions index

position in the block is **0** (**transaction_index field**) for each day in **October 2015**. select the date field and the two resulting fields, then sort the results by date. You need to produce the date field and the two required fields, then sort by date. You need to include a screenshot of your results in your report. **Do not truncate the fields.** An example of the format of the results is only provided in this question. Please ensure that you include the timestamp in your screenshot, as required.

formatted_date	**total_transaction_fee**
2015-10-01	**2.43245454545E18**
2015-10-02	**1.43247699090E18**
2015-10-03	**3.78234989062 E18**
2015-10-04	**1.2853847909173277E19**
……	**……**

Note that the above figures/values may not represent the actual result. Your screenshot needs to include the running timestamp.

Then, you need to extract the results into a file and plot from the results **histogram plot** with **'formatted_date'** on the x-axis and **"block_count"** and "**total_transaction_fee**" on the y-axis. For example, assume if the "**total_transaction_fee**" in **"2015-10-01"** is **2.43245454545E18**, **"2015-10-01"** will be on the x-axis, indicating **2.43245454545E18** above the bar of **"2015-10-01"** separately**.** You will need to include the histogram in your report.