Author: Tim Guo

problem：

How often does the Bitcoin network see two consecutive blocks mined more than 2 hours apart from each other?

We'd like to know your answer (it doesn't have to be precise) and your approach towards this solution using probability and statistics.

How many times has the above happened so far in the history of Bitcoin?

problem approach：

See all the codes here: https://github.com/gtm1224/bitcoin_block_time_data_ETL_Analysis

Note: we ignore forking

1. get bitcoin block data:

A.

One way is to run a bitcoin core on my machine which needs time to synchronize and it wastes my disk storage. I run it on my cloud server but it is still synchronizing (needs more than 4 days). If synchronizing successfully I would use python bitcoinlib to access bitcoin node. Using python to query and store the data to a database or .csv file instead of using "bitcoin-cli getblockstats #(height)" on command line.

B.

I choose the other way which is sending get request to a third-party API. After testing many APIs, I finally decided to use https://www.blockchain.com/api. See documentation.

The way is to query each block height using python multi-threading. To mimic the producer and consumer ETL model, the function Get_Data(height) will get the block height and timestamp the block mined. Then, it stores the extracted and transformed Pandas DataFrame to a Python Queue data structure. Function Load_Data() will read data from the Queue. Get_Data and Load_Data are in different threads. Note, we cannot create too many threads and request the API at the same time, our IP will get blocked. So I used batch strategy by creating 50 requests per batch.

See bitcoin_block_data_collection_multi_thread.py

Updated version: bitcoin_block_data_collection_multi_thread_new.py (this runs slower but safer for thread)

C.

I also used a web crawler to crawl the data directly from the web: https://btc.com/btc/blocks. I used python selenium and Google chrome driver to extract data automatically. For correctness the program adjusts to show 100 lines of data per page, the total page I need to crawl is about 7200. My web crawler program ran while I was sleeping. Finally, got the data. See selenium_spyder.py

Problem and Solution:

In general, for no Proxies, C is the fastest way to get all the data. With Proxies, B would be the fastest way to get all the data.

The problem of B is IP rejection. After getting around 400000 lines of data. My IP get rejected and cause threads to run out of resource since each thread would keep asking for the data

at a certain height and not getting destroyed. Another problem is that using multi-threads will not get block height data in order. If we could get all the data, then we can apply a sort to keep our data quality. But, our IP gets rejected, after sorting, some data will still be missing. The possible solution is to use Proxies IPs. However, free Proxies are not alive all the time and connections are bad. They do not suit our fast multithreading requirements.

The problem of C is the dynamic update of the web because of newly mined blocks getting added to the web database. This will cause data missing when we go to the next page, the data shown on each page will be changed.

The solution is to first concatenate all the files and get rid of redundancy. Then, find missing block height, append it to a list. Use multithreading to request to get the missing data.

2. Data analysis:

It turns out that it is easy to calculate the average time between each incidents (two consecutive blocks mined more than 2 hours apart ) when we have the data. But, we need to find a better way describe the problem and estimate when we do not have full data.

The problem can be seen as a Poisson Process. If we see a block get mined as an event. We know that on average the time between two blocks mined is 10 minutes i.e. one event happen every 10 minutes! We know that every two events happen are independent of each other. We also know that for the main chain two events cannot happen at the same time because main chain gets rid of forks. These conditions satisfy Poisson Process.

“Common examples of Poisson processes are customers calling a help center, visitors to a website, radioactive decay in atoms, photons arriving at a space telescope, and movements in a stock price. Poisson processes are generally associated with time, but they do not have to be. In the stock case, we might know the average movements per day (events per time), but we could also have a Poisson process for the number of trees in an acre (events per area).”

$$P(k \text{ events in time period}) = e^{-\frac{\text{events}}{\text{time}} * \text{time period}} * \frac{(\frac{\text{events}}{\text{time}} * \text{time period})^k}{k!}$$

Poisson distribution for probability of k events in time period.

$$P(k \text{ events in interval}) = e^{-\lambda}\frac{\lambda^k}{k!}$$

Poisson distribution probability of k events in an interval.

For this problem, the event is a block gets mined. So we would like to know the probability that one event happen in 2 hours. In other words, after two hours we see a new block gets mined. 2 hours is 120 minutes. The average time for one event to occur is 10 minutes; we can calculate the $\lambda$=120/10 which is 12. We only want to know 1 event that happen in 120 minutes, so k is 1. We can calculate the probability that $P(k = 1) = e^{-12}\frac{12}{1} = 7.373 \times 10^{-5}$ . So we can estimate the

total number of the incidents that two consecutive blocks mined more than 2 hours apart from each other is $720000 \times 7.373 \times 10^{-5} = 53$ times which means that they happen on average 720000×10/53 =135849.0566 minutes=94 days. So, in estimation, two consecutive blocks mined more than 2 hours apart from each other should happen every 94 days on average. From my notebook, I counted the total number of this incident happened 390 times which gives an average time that incident occur at 12 days. This is not convincing to me. I realized that I should not convert unix timestamp while extracting the data, even I did data pre-processing, the raw data might get contaminated. The other reason might be while web crawler was crawling the web keep updating causing a scroll in data, this can also contaminate our raw data.

readings：
https://bitcoin.stackexchange.com/questions/77783/shortest-and-longest-block-interval-time-ever-recorded-in-bitcoin
https://developer.bitcoin.org/devguide/block_chain.html
https://www.blockchain.com/api
Youtube block info intro:
https://www.youtube.com/watch?v=ByEKBLvTNb4
Poission Process:
https://towardsdatascience.com/the-poisson-distribution-and-poisson-process-explained-4e2cb17d459#:~:text=A%20Poisson%20Process%20is%20a,time%20between%20events%20is%20memoryless).&text=All%20we%20know%20is%20the%20average%20time%20between%20failures.