Using Databases for Blockchain Analysis

Objective: NLT than July 26th (36 days) perform analysis on Bitcoin’s Block chain by integrating database. Complete Research paper in IEEE format. Deliver oral presentation on our findings.

Abstract: Bitcoin represents to most well established cryptocurrencies. Fundamental to this technology is cryptography and distributed ledgers called blocks. People refer to it as distributed because no centralized agent manages a blockchain ledger on its own. The ledger is distributed across computers connected to the internet, called nodes. Each node keeps a record of the entire blockchain and receives updates on transactions when they occur. Increase in popularity of Bitcoin created a greater demand on the network. A problem of scalability emerged. The Bitcoin communities current approach to solve this problem is the Lightning Network. A second layer transaction network that runs on top of the blockchain. Similar to how visa, venmo run on top of the FedWire system. Nodes can establish a channel on the lightning network to send payments very fast and with minimal transaction fees. Settlement of the transaction occurs later. Nodes can also send payments through channels that their connections are connected too. As the network grows more efficiency is gained. Our interest is to evaluate the min/max and optimal fee to maximize profitability of the node.

Approach: Break the problem down into small steps.

1. Proof of concept
2. Establish a node and synch it with the bitcoin network.
3. Transfer the data from the node synch to a database. This will provide a proof of concept for how to move data from the node to perform the more complicated analysis on the lightning network
4. Database integration exploring best use cases and practices for preparing the data for analysis.
5. Do structed Data Science utilizing the skills we’ve learned thus far.
   1. learn which packages might exist to help managing the type of data
6. Second Layer Analysis
7. Establish Bitcoin wallet in order to establish payment channels and inject liquidity into our node.
8. Transfer data from the node about the lightning network transactions
9. Validate Database integration with lightning network data
10. Do Data Science

Building a Node:

There exists on the market many out of the box solutions to establish a Bitcoin node on a home computer network. These solutions cost around $400 dollars and up. Another solution is to purchase your own hardware and then integrate a company’s proprietary software. Entry price for this solution is $250. The cost-conscious solution to running your own Bitcoin Node is to employ cloud computing to rent what you need and use open-source software to run a node costing $18 dollars a month. This is the approach we’ve taken. Currently we are exploring two-options. BTCPay and MyNode. MyNode is not techniquely free though by asking the company for support in our research they’ve offered free software which we can run in the cloud.

BTCPay

What is BTCPay: It is an open-source, self-hosted payment processor for Bitcoin and other cryptocurrencies. BTCPay Server is not a company. To use BTCPay Server, we deployed our own instance in the cloud. Most users are interested in the payment processing feature of BTCPay to integrate into their webpage or online store.

Graphical user interface

Description automatically generatedeach time at check our BTCPay Server provides the customer with an invoice.

LunaNode: Graphical user interface, text, website

Description automatically generated

The developers of BTCPay server teamed up with a web hosting company called Luna Node to provide a user friendly solution to deploying BTCPay.

To launch BTCPay server on LunaNode we followed the following steps:

1. Created an Account on LunaNode:
2. Create New API
   1. Graphical user interface, application

      Description automatically generated
   2. I stored the public and private keys safely.
   3. Then set up the instance
      1. Graphical user interface, application

         Description automatically generated
   4. To lunch the BTCPay server on luna Node I went to
      1. Graphical user interface, text, application, email

         Description automatically generated
      2. I let Lunda node host a domain for me. btcpay983737.lndyn.com

Graphical user interface, application, Teams

Description automatically generated

* + 1. Provisioned the VM using the BTCpayserver-docker github repository

Graphical user interface, application

Description automatically generated

* + 1. The node took 3 days to synch with the blockchain network.

Graphical user interface, website

Description automatically generated

Accessing Data on the Node:

Our BTCPay server stores data in different docker containers. To access these containers we SSH into the server. Once in side we establish root permissions with:

sudo su –

cd btcpayserver-docker

docker exec -ti $(docker ps -a -q -f "name=postgres\_1") psql -U postgres

# list databases

\l

# connect to database

\c btcpayservermainnet

# list tables

\dt+

# list users

SELECT "Id", "Email" FROM "AspNetUsers";

# end session

\q

Next steps:

* Currently the group is working on exporting the data into mongo DB or NEO4J
* Then the group will do Data Science
  + This will conclude our proof of concept
* Establish a Bitcoin wallet
* Create channels to other nodes
* Collect data on the LN traffic
* Export the data to the database
* Do data science on the data.

Index

BTCPay: <https://btcpayserver.org/>

BTCPay: Documentation: <https://docs.btcpayserver.org/>

BTCPay Set up using LunaNode: <https://medium.com/@BtcpayServer/launch-btcpay-server-via-web-interface-and-deploy-full-bitcoin-node-lnd-in-less-than-a-minute-dc8bc6f06a3>

Luanch BTCPay using Luna Node: <https://launchbtcpay.lunanode.com/>

Step by step video: <https://youtu.be/NjslXYvp8bk>

BTCPayServer-Docker “how to connect to the database” : <https://github.com/btcpayserver/btcpayserver-docker>