

CSC111 Project Proposal: Reconstructing the Ethereum Network Using Graph Data Structures in Python

Tobey Brizuela, Daniel Lazaro, Matthew Parvaneh, Michael Umeh

Tuesday, March 16, 2021

Problem Description and Research Question

Peer-to-peer computer networks are networks whose infrastructure relies on distributed computing, and are generally completely decentralized. Peer-to-peer networks starkly contrast with most conventional computer networks, which tend to have dedicated servers that handle requests from clients and subsequently serve webpages or other kinds of digital content back to them. Instead, the peer-to-peer network model treats all computers that make up the systems as equal, each being a "peer" in the greater system (*What Is a Peer-to-Peer Network?*, 2020). Each peer in the network will be connected to some of the other peers on the network and work together in order to accomplish a certain task; historically, peer-to-peer networks were most often used for file-sharing (e.g. Napster.com) (Peterson, 2020). Evidently, this network structure is a prime candidate for representation in the form of the graph data structure; every computer or peer in the network can be represented as a vertex, while each connection between peers can be represented by an edge.

While peer-to-peer networks were most commonly associated with file sharing in the past, they have begun to see usage in the realm of cryptocurrencies, which have become increasingly popular in recent years. Popular cryptocurrency platforms such as Bitcoin and Ethereum rely on this same technology behind peer-to-peer networks in addition to blockchain technology in order to make and record transactions, which is a peer-to-peer network itself that can also be modelled with graphs (Hu, 2018). Ethereum in particular, created by Russian-Canadian programmer Vitalik Buterin, has public ledgers and makes all of its transactions public (Wackerow et al., 2021), thus, the data can be easily collected, formatted and manipulated in Python, in order to gain insights into the way transactions occur in the system.

Ultimately, all of this lends itself to an interesting research question: **how can we replicate the structure of the Ethereum peer-to-peer network using graphs in Python, and then visualize the resulting structure?** After doing this, it will also be prudent to determine the correlation between the balance of Ether (the name of the cryptocurrency itself) at an address/wallet and the number of transactions to and from that address.

Computational Plan

Processing Data

As explained in the introduction, the Ethereum network consists of thousands of nodes (computers with unique addresses) that are connected to each other by transactions. Therefore, we can model the network using a directed graph, where each vertex represents an account on the Ethereum network, and each edge represents a transaction between two accounts. We will be writing code in Python that automatically takes the data from our datasets (formatted as csv files) and creates a graph. The vertexes of the graph will be initialized using the unique address and Ether balance of each account, which can be found in our balances dataset. Then, edges can be added one by one from our transaction dataset, which contains the from address and to address of each transaction on the network. Both of these datasets can be retrieved from Google's [BigQuery public database](#), which provides comprehensive data on a variety of blockchain-based decentralized networks, including [Ethereum](#). **Sample data** (in csv format) can be downloaded from [UTSend](#) (ID: rVZ8UexXgHxUsaoU, Passcode: oWJSgoh848PHNSmS).

To make analysis and visualization easier, we will be using the [NetworkX](#) Python library to build our graph. This library allows us to assign arbitrary attributes (like hashes, balances, transaction amounts, etc.) to both the nodes

and the edges of the graph, which makes it perfect for our use case. It also allows for directed graphs, which are necessary to determine the sender and receiver in each transaction. Finally, NetworkX also integrates nicely with the [plotly](#) library, which will allow us to visualize our graph in 2D space (more on this later).

Computing Data

Once we have created our graph, we can perform calculations on the data and attempt to answer our main question: “what is the correlation between Ether balance at an address and the number of transactions to/from that address?” To do this, we will write code that adds up all the transactions that an account has participated in by looking at its neighbours. Then, we can compare this with the account balance that we get from the balances dataset. Once this is done, we can also answer some related questions. For example, “are accounts with higher balances more likely to make transactions with other accounts with high balances, or vice-versa?”.

Visualizing Data

After performing these calculations, we hope to visualize our Ethereum graph using the plotly library, as mentioned previously. One feature of plotly is that it allows us to [adjust the size and colour each node according to certain values](#). For example, we can size each node according to the number of edges it is a part of. We may also be able to colour/size nodes according to the balance of the account that each node corresponds to. Visualizing the data this way will allow us to see patterns in the graph, such as clusters of high balance accounts (if such clusters exist).

References

Datasets – Marketplace. (n.d.). Google Cloud Platform. Retrieved March 16, 2021, from <https://console.cloud.google.com/marketplace/browse?filter=solution-type:dataset>

Ethereum Cryptocurrency. (2020, February 7). Google Cloud Platform. <https://console.cloud.google.com/marketplace/product/ethereum/crypto-ethereum-blockchain>

Hu, L. (2018, October 30). Understanding Ethereum’s P2P Network. *Shyft Network - Medium*. <https://medium.com/shyft-network-media/understanding-ethereums-p2p-network-86eeaa3345>

Intro to Ethereum. (2021, March 10). Ethereum.Org. <https://ethereum.org>

Network Graphs. (n.d.). Plotly. Retrieved March 16, 2021, from <https://plotly.com/python/network-graphs/>

NetworkX documentation. (n.d.). NetworkX. Retrieved March 16, 2021, from <https://networkx.org/>

Peterson, M. (2020, August 12). The History of P2P Networks- and Why They Remain So Important Now. *Apaylo*. <https://apaylo.com/2020/08/12/the-history-of-p2p-networks-and-why-they-remain-so-important-now/>

What Is a Peer-to-Peer Network? (n.d.). Indeed Career Guide. Retrieved March 16, 2021, from <https://www.indeed.com/career-advice/career-development/what-is-a-peer-to-peer-network>