BlockMaps

COMP4601: Final Project Report
Carleton University Faculty of Computer Science

Prof. Anthony White
& Course TA Rui Li
Ted Kachulis 100970278
Henry Nguyen 100968253
Final Project Report April 14th 2019

ABSTRACT

BlockMaps is a web application that provides visual and aesthetic insight into the communities which make up a given ethereum blockchain network. Using scraped data from posted network transactions, our application constructs a visual map of relationships between each unique node (user) address using Neo4J tools. The resulting display represents the relationships between nodes, and the flow of value throughout the network. These aspects, both based off of historical transactional data, help us to then infer properties about given nodes on the network. As of late, blockchain technologies are a heavy focus for industry leaders as they discover how to best utilize the benefits of these decentralized networks. Our hope is that as the adoption and prevalence of blockchain networks increase, *BlockMaps* tools will provide a method for data scientists to conduct information retrieval, visualization, and basic analysis for these networks on a large scale.

INTRODUCTION

In the fields of both modern systems and business, data reigns supreme. More specifically, large data sets regarding relationships between members in a network, and active user preferences which are derived from a given users associated data. At present, being able to capture data for the purpose of a clients contextual experience is critical to most service providers - as is being able to interpret this data both accurately and efficiently.

That's the motivation behind the development of *BlockMaps*. As companies increasingly utilize blockchain for large data, we wanted to publish a release of network analysis and visual graphing tools to be used for this aforementioned task of visual interpretation and predictive analytics. Our goal was to include functionality for both analyzing network traffic, and categorizing through community (similarity) detection - all while implementing graphical elements which represent these results into our visualization.

As described in the document index, following this introduction there are six sections to this project report. The report first covers any background information required in order for the reader to understand this project. It then goes on to note external works related to our project focus, informing the reader of projects that deal with similar technologies and issues. Third, the methodology section breaks down how our team achieved our aforementioned goals. It then goes on to provide further detail on our problem statement, our approach to solving the problem, and our implementation decisions. In the discussion section which follows, we talk about our successes, struggles, as well as note our observations and further questions. Finally the project is summarized in the conclusion, but not before the mention of future work possibilities; including additional functionality desires or specific analytical interest.

BACKGROUND INFORMATION

In order to adequately understand both the project scope as well as implementation, the reader should ensure that they have taken a look at all of the tools, concepts, and platforms which follow. Generally, the links provided are a good place to start.

BLOCKCHAIN PROPERTIES

The reader should have basic knowledge of blockchain properties; referring to the composition of a block, and how identified nodes paired with transaction data between them form a decentralized community network. These properties allow us to analyse transaction activity as we visually model a blockchain network, and infer properties about a given node. Research such as our own could lead to making predictions about future transactions on these blockchain networks.

The data moving in transactions amongst the network users can be arbitrary (depending on the intended network use case - we use ethereum) however each block contains an identifying hash, and a hash identifying its link in the chain. A useful summary of blockchain networks and technology can be found here [1].

For visual learners I suggest the following video - a helpful guide explaining the functionality, origin, and use cases of blockchain networks can be found here [2].

NEO4J: Graph Database, Visualization, and Strongly Correlated Components

To better understand the tools we leveraged in our implementation, the reader should look into the Neo4j graph platform documentation. Neo4j is an open source graph database tool (among other things) that provides an extensive implementation for a database, server, and web interface.

An overview of the tools we've used in our project for graphing and analysis of data can be found here [3]. The tools we used more specifically for graph visualization are also built using Neo4j, and can be reviewed here [4].

ETHEREUM BLOCKCHAIN, PARITY NODE (Data Source)

The information our system analyses has been modelled based on transactional data posted on the ethereum blockchain, one of the most widely used distributed application platforms. Ethereum uses an account based transactional data model, linked to form a graph. Node data can be downloaded from Parity, an implementation of an Ethereum node, learn about it here [5].

This link [6] leads to a thorough explanation of the origins and implementation details in regards to the ethereum blockchain.

RELATED WORKS

There are a limited amount of offerings for blockchain graph analysis due to its recent industry adoption, however it is becoming much more prevalent as more use cases are discovered - thus increasing the need for analysis and accessibility. As documentation for this technology becomes more accessible, platforms for this use are becoming more polished as a result. What makes our goal different than what exists today in the market is the focus on the ethereum blockchain.

CHAINALYSIS

Chainalysis is an impressive example of a recent project that focuses on the prevention and detection of fraud, laundering, and compliance issues on blockchain networks. They provide services on any scale for detecting suspicious activity in networks, or capturing user analytics for marketing purposes.

Knowing the modern prevalence of the data science field, it is logical to believe blockchain analysis will become an important niche. Chainalysis definitely serves as inspiration for our project, providing us with a vision of goals for our own system. All relevant information about Chainalysis can be found here [7].

KAGGLE: Bitcoin Blockchain Analysis (K. Deshmukh)

While we have focused on address categorizing and visualization using the ethereum blockchain as the source for our analysis, Kartik Deshmukh alternatively focused on bitcoin price variance based on his captured and observed blockchain analytics using python.

This piece of work, published by Kaggle, was entered as a competition piece by Deshmukh in 2018. Using the aforementioned data scope of the bitcoin blockchain, Deshmukh's work looks at miner selection incentive as well as value variance based on transactional data over time. A story like walkthrough of Deshmukh's project can be found here [8], with his comments on how such analysis will benefit data science.

GRAPHSENSE

Similar to Chainalysis, Graphsense is a provider in regards to blockchain analytics. The key difference between the two are what separate many projects of this nature; the blockchain platforms they are built to support.

Graphsense, the academically referenced tool for blockchain analytics, supports tools for bitcoin, bitcoin cash, litecoin, and zcash. Including a REST api for retrieving bitcoin data and more, a condensed summary of their works can be found here [9] along with their project.

APPROACH AND METHODOLOGY

Our goal was to create a solution which provided the ability to capture, visualize, and analyze blockchain data. We required a sample data source, and decided to use the ethereum blockchain for two reasons. First due to its prevalence in terms of blockchain platforms, and secondly, a lack of related works showed support for ethereum - a platform for which we show support.

Upon setting out for this goal, our tasks seemed to naturally fall into **4** predominant sections. In each section the given task will be described, and our implementation decisions explained.

TASK 1: Capture and Store Ethereum Blockchain Data (Implementation)
We required functionality capable of capturing the relational transaction data found on the ethereum blockchain. For this we used typescript and the parity node api - using RPC calls to pull the block data into Neo4j. Data was very large and had to be streamed to our analysis through a Neo4j graph database. Using this we were quickly able to interact with data, thus permitting us to make better implementation decisions early on.

TASK 2: Visualize Ethereum Blockchain Data (Implementation)
Secondly, being able to visualize the blockchain network was a priority. Storing the transactional data in Neo4j's graph database allowed us to quickly utilise the Neo4j graph platform tool library to visualize the relational data. We decided after analysis to have the size of each node be relative to its pagerank score, and the color is associated with its computed community category. The thickness of arrows between two nodes represents the quantity value of ether in the transaction.

TASK 3A: Analyze Ethereum Blockchain Data (Thinking)
The third task was harder to complete, as we had to pinpoint what types of analysis would help us provide insight for real world networks. Several options came to mind which could benefit stakeholders using blockchain platforms (i.e. future dev works / business traffic analytics).

There are many use cases for analysis which we could implement here such as scam detection, merchant detection, purchase tracking, etc - there's more to be found on future possibilities with our projects in the *future works* section. For now the following will describe our goals for analysis implementation, providing proof of concept for future analysis possibilities.

The first decided objective **[O1]** was to identify currency exchange platforms on these networks by flagging high traffic frequency found on graphed blockchain data. Our second decided objective **[O2]** is to allow relationships between id's active on the blockchain to be understood, providing insight into "communities" or "categories" of wallet id users. Implementation details and our solution for these goals are to follow in **3B**.

TASK 3B: Analyze Ethereum Blockchain Data (Implementation)
We implemented two main sets of functionality to satisfy objectives **[O1]** and **[O2]**. To satisfy the defined objective **[O1]**, we decided on using pagerank. This turned out to be quite an efficient method of identifying high traffic nodes in the graph, as we were able to leverage even more Neo4j functionality with built in relational scoring functionality.

To give understanding to node relationships on the network, thus satisfying objective **[O2]**, we use Neo4j's interactive graph database dashboard and visualization tools. Not only is this accomplished with visualization, but by running community detection on our graph database. We chose to use Neo4j's implementation to detect strongly connected components found here [10], which helps us see patterns like interconnected exchange nodes on the network - helping to detect cases of arbitrage. The details of implementation are further touched upon in the discussion.

TASK 3C: Interpret Analysis Observations (Thinking)
While the implementation of pagerank and the visualization of ethereum network id relations provides insight on its own, a goal of ours remains providing helpful analysis to the end user. To ensure this, we had to look at the resulting data from our analysis and piece together what metrics allow for helpful inference about a given network or user.

Studying the results from objective **[O1]**, we noticed that using pagerank to score popularity of a given id permits us to make certain assumptions about an id; high transactional frequency indicates a currency exchange, low transactional frequency indicates inactive user, etc. The given address can then be categorized based on network traffic - this was confirmed through observing our top pagerank results, which is further explained in the *discussion*.

Using our community detection and visual representation after the completion of our aforementioned objective **[O2]**, we are able to track id properties and therefore categorize them accordingly. Not only is this currently beneficial, as we can detect exchanges and mining pools, but useful in the future too when marketing becomes prevalent on distributed applications. If ecommerce operated more frequently on blockchain platforms, these observations could help lead to a contextual experience based on inferred user preferences. More on these observations can be found in the *discussion* and *future work* sections of the report.

TASK 4: Provide End User Functionality (Implementation)
To satisfy this goal, we continue to use typescript and available apis. The end user experience leaves a lot of room for improvement, but we made sure to visually represent transaction heuristics with intuitive interpretation in mind. Users can perform actions like analyze a node id; visualize ethereum block segment; analyze traffic.

DISCUSSION

Overall we were able to achieve most of our goals, and the results of our analysis were those desired/expected. Our implementation consisted of two key desires. First being able to construct an *augmented* visual representation of network data for the ethereum blockchain, and secondly to infer address *categories* through analysis of transaction heuristics.

By running our pagerank implementation on all of the nodes in the network, we are able to score them by popularity. Since each node is a wallet address and the edges between them are each a transaction, we can infer that the most linked to nodes will be most likely to be currency exchanges (or mining pools,etc) since they are a central point of interest.

To see how our addition of pagerank allowed us to then augment our blockchain network visualization, see *Figure 1* here [1]. During observations we were able to validate our strategy by checking the pagerank results, noting that our top result reflects the address of one of the worlds largest cryptocurrency exchanges: Binance. Other top hits from our results included things like token contracts, mining pools, etc - all labeled as high traffic points of interest in our analysis.

The addition of community detection within our project was meant to help classify things, for example; which users transact often with a particular service; flag arbitrage; assess address categories. We chose to leverage Neo4j's strongly connected components tools in this case to categorize each node, and to monitor the bidirectional connections between them - specifically between highly scored nodes to detect arbitrage or other mass movements of value. Each node is then associated with a different color in our visualization, representing its determined category. An example of the blockchain network visualization augmented with community detection can be seen in *Figure 2* here [2].

After these two goals had been met, we were able to recreate our expected results to repeatedly identify node properties properly. Once we had an augmented network graph containing communities and address types, we wanted to add functionality for an individual node inspection. Using the augmented graph (with community colors, traffic ranking, and visual transaction data), a graph like the one shown in *Figure 3* here [3] can be computed for any given node address on the ethereum network.

Our results showed us that it is possible to retrieve reliable outcomes with our blockchain modeling methods, and have validated our thoughts on the implementation strategies chosen. By categorizing nodes based on their attributes, we can track, advertise, or give contextual benefits through different platforms based on the address of a node in the network.

FUTURE WORK

Both in terms of our own project and the other works regarding blockchain analysis going forward, there are many more opportunities for future work. In regards to *BlockMaps* we could have done a lot more, given more time to allow focus on this project. For example, we wanted to implement services for things like detecting exchange scams, mentioned earlier. Something we could do to implement this functionality now would be to utilise value tracking on the network, meaning we could trace all outgoing transactions in nodes from a given node source. From there we could implement filters to identify unique transaction patterns indicating malicious movement in the network, and flag the source node address owner. Another thing we weren't able to complete in this release of *BlockMaps* was a, ideal user interface. Though the visualization is what we imagined in the front end, in the future we could implement a dashboard displaying analytics in a list format as well. Additional features for the end user could include input areas for node specification, and the ability to export a produced graph visualization.

In terms of future works for the analysis of blockchain networks, there is surely more growth to come. Groups such as the ones mentioned in *related works* as well as others are continuing to add support for more blockchain platform variants. As businesses move towards decentralized online models, the analytics on these networks will become more prevalent, presenting funding for more analytical research opportunities. In the future, expect to see implementations for predictive modeling on blockchain networks in order to provide contextual experiences, facilitate the end user experience, as well as provide marketing analytics for business. A specific application of predictive analytics directly related to our project could be to predict price variances on cryptocurrency markets, such as presented by Kaggle in related works.

CONCLUSION

Motivated by the growth of blockchain prevalence in the data sector, we set out to create a tool for ethereum blockchain visualization and analysis. Our mission: to help those interested gain insight into the relational networks derived from block data and associated address id's. By leveraging the Neo4j graph platform, we went on to implement both helpful analysis and an intuitive visualization offering for the ethereum blockchain. Our analysis allowed us to infer several properties about a node such as type of user, transactional habits, as well as network popularity. Secondly it allowed us to augment the default graph visualization to offer informative graphic elements based on node relations. In the future, our improved work as well as industry solutions will benefit the development of data solutions, and business analytics. Pairing secure blockchain data with highly adaptive tools from Neo4j provide a great platform for hosting large datasets. As decentralized applications grow on blockchain platforms, projects like ours will be a great place to start for projects regarding ethereum network visualization and analysis.

REFERENCES

1. Blockgeeks Group. Blockgeeks website. *What Is Blockchain Technology?* 21 Mar. 2019, https://blockgeeks.com/guides/what-is-blockchain-technology Accessed 9 Apr. 2019.

2. Blockgeeks Group published via Youtube. *What Is Blockchain Technology? Easy to Understand Video*. 5 Nov. 2018, https://youtu.be/27nS3p2i_3g Accessed 9 Apr. 2019.

3. Neo4J Group. Neo4j Website. *Neo4j Graph Database Platform*, 2019, https://neo4j.com/developer/graph-platform Accessed 8 Apr. 2019

4. Neo4J Group. Neo4j Website. *Neo4j Tools for Graph Visualization*, 2019, https://neo4j.com/developer/tools-graph-visualization Accessed 8 Apr. 2019

5. Parity. Parity Website. Parity: *Blockchain Infrastructure for The Decentralised Web,* 2019, https://www.parity.io/ Accessed 8 Apr. 2019

6. Blockgeeks Group. Blockgeeks website. *What is Ethereum? The Most Comprehensive Guide Ever!*. 2017, https://blockgeeks.com/guides/ethereum/ Accessed 9 Apr. 2019.

7. Chainalysis. Chainalysis website. *Chainalysis Informative Landing Page*. 2019, https://www.chainalysis.com/ Accessed 8 Apr. 2019.

8. Kartik Deshmukh. Kaggle website. *Bitcoin Blockchain Analysis*. 2017, https://www.kaggle.com/kardesh/bitcoin-blockchain-analysis Accessed 10 Apr. 2019.

9. Austrian Institute of Technology. Graphsense website. *Graphsense Landing Page*. 2016-19, http://graphsense.info/ Accessed 10 Apr. 2019.

10. Neo4J Group. Neo4j Website. *Neo4j Tools for Graph Visualization*, 2019, https://neo4j.com/docs/graph-algorithms/current/algorithms/strongly-connected-components/ Accessed 9 Apr. 2019.

FIGURES