2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

Dru: Studying Blockchain as a Complex Network

Radosław Michalski, Marcin Pieczka
Wrocław University of Science and Technology, Faculty of Computer Science and Management
Department of Computational Intelligence, Wrocław, Poland
Email: radoslaw.michalski@pwr.edu.pl, pieczka.marcin@gmail.com

***Abstract*—Apart from many project design decisions that have to be made when incorporating blockchain as a trusted transactions register, it is also essential to gain a deeper understanding of the ecosystem when it already entered operational state. By looking at the way how nodes perform transactions, how they cluster, or what is the dynamics of the network, many valuable insights can be derived about the condition of the whole system. This leads to improving functionality, performance, scalability, and security. We propose a way of looking at the network of transactions as at a complex network and demonstrate that the methods and algorithms provided by network science can significantly increase our knowledge about the blockchain ecosystem. In this work, we introduce an open-source platform called *Dru* that is linked to this concept and simplifies performing analyses by providing a built-in storage of transactions and an API for research. We believe that the insights presented in this work will be helpful for studying blockchain-based systems. As a result, those systems supporting variety of processes will be able to move from reactive to proactive thinking and improve their functionality, availability and security.**

***Index Terms*—blockchain, distributed ledger, network science, complex network, software platform**

I. Introduction

Blockchain solutions are entering many crucial areas of business. Started as trusted ledgers for cryptocurrencies, since few years they have been considered as transparent yet secure platforms for keeping transactions register in many fields [1]. In some cases, the idea reached only the stage of a proof of concept [2], however widespread implementations exist as well in variety of domains: finance [3], supply chain [4], or land ownership [5], to name only a few.

Meeting the project requirements for introducing distributed ledgers to business is only the beginning of making this solution long-lasting. Creating a sufficient user base or offering high levels of reliability and security can be considered as minimal prerequisites. However, as the transactions are starting to appear, it is also needed to monitor the ecosystem in many aspects: how the blockchain evolves, what are the crucial entities, whether it is about to form the expected shape, and no anomalies emerge in its structure [6], [7].

This task exceeds the typical monitoring techniques known to IT experts, such as mean time between failure, hardware load, or storage requirements of a system. Here, we focus on the internals of blockchain blocks, since they encapsulate information about the transactions between entities. The analysis of these transactions can provide helpful insights that can be turned into system improvements. To simplify the way of analysing the blockchain, in this paper we introduce a platform for making the research on blockchain straightforward.

IEEE/ACM ASONAM 2020, December 7-10, 2020
978-1-7281-1056-1/20/$31.00 © 2020 IEEE

II. Dru: a Platform for Analysing Blockchain

A. Introduction

Usually, researchers asking any blockchain-related research questions have been independently working on gathering data, and analyzing it later on. In the case of Bitcoin, apart from several online tools that typically impose limits on the use of their API, such as blockchain.info[1] or blockexplorer.com[2], the solutions space is scarce. One can also find libBitcoin-database[3] that offers in-memory storage of data that is troublesome due to blockchain size. When thinking about industry blockchain implementations, the situation is even worse, since no open source tools for performing such analyses exist. This lack of widespread available solutions motivated our work, and as a result, the *Dru* platform was built. The Dru platform presented in this work is intended to support the researchers as it provides a consistent environment for acquiring data on a blockchain that can be used for further analysis. Albeit Dru has been developed for analyzing blockchains related to cryptocurrencies, it can also be easily ported to support many other blockchains. In the next subsections, we describe the concept, requirements, and implementation of Dru.

B. Concept

Firstly, let us look at how big particular blockchains can be. For instance, currently the blockchain size of Bitcoin is 310 GB, and it grows at a rate of 180 MB per day[4]. Because of that, making large scale analyses using external services can become impossible due to performance issues and API restrictions often implemented in available tools. Of course, some other blockchains can be smaller, but knowing that every performed transaction contributes to their size, many of them will encounter the problem of volume at some point. To enable the ease of doing research in this area, the platform introduced in this work provides access to blockchain data by creating a database of blockchain blocks accessible either locally or over the network and this platform is kept in sync with the state of the blockchain.

Another problem is addressing the knowledge sharing in terms of tools and functions that researchers implement for

[1]https://blockchain.info/
[2]https://blockexplorer.com
[3]https://github.com/libbitcoin/libbitcoin-database
[4]https://blockchain.info/charts/blocks-size?timespan=all

their studies. Overlapping work wastes valuable time that could have been spent on conducting experiments. To solve this problem, an API is offered that can be easily extended with new functions by people having even little experience as developers. This approach leads not only to more reproducible research in the area of studying blockchain, but also enables researchers to focus on analyses instead of coding.
The requirements for the system are then the following:

- easy installation, leading to extending userbase
- good performance, necessary for computationally complex queries
- the integrability with R and Python, programming languages commonly used in research
- the simplicity of use, also contributing to extending userbase
- the possibility to provide the results of past queries
- facilitating the reproducibility of research

C. Implementation

The Dru platform consists of separate components, each of them having a distinct set of responsibilities. The modular design helps to maintain low coupling and increases readability, but all interplay together.

Each component is hosted in a docker container. Docker was chosen for simplicity of installation of the system, and to avoid possible problems with running it on different Linux distributions. However, it can be decided whether the blockchain node should be taken from the docker container or the local one will be used. This applies to the fact that many researchers are already running their nodes. Forcing users to run the second instance would surely be a hindrance to them.

A block engine is a container responsible for gathering and preprocessing data stored by the cryptocurrency node and inserting upcoming blocks. The source of the data are the Bitcoin RPC calls, which are used by a multitude of cryptocurrencies. After creating an internal representation of blockchain, transaction input addresses are being discovered. These transactions specify the source of resources with transaction hash and index of output in a transaction. For obtaining the input address, a transaction with a specified hash must be found, and its output, specified by output index, becomes that input address. As an input address discovery is the biggest bottleneck in the whole process, a transaction caching mechanism has been implemented. The cache holds a constant number of transactions parsed recently.

MongoDB was chosen as a database engine for its dedicated APIs for Python and R, and native support for denormalized data. The collection of blocks is indexed by height and timestamp attributes, and transactions by their hash. Index for transaction hash is needed for discovering source address of the transaction. When database updater misses cache, the database is then querierd for that specific transaction.

The last container is the API for additional functionalities. When a researcher needs new functionality regarding block data, it should be almost as simple to implement it in this API as locally. A number of API endpoints providing variety of network science-related queries are already available and described in the next section. The architecture of the platform is presented in Figure 1. Dru also uses two components that are omitted in Figure 1 for clarity. Their role is message passing between the components (Rabbit MQ) and providing a queuing system (Celery). This is due to the fact that some of complex queries can be time-consuming and the queuing system allows to organize the scheduling and priorities for jobs.

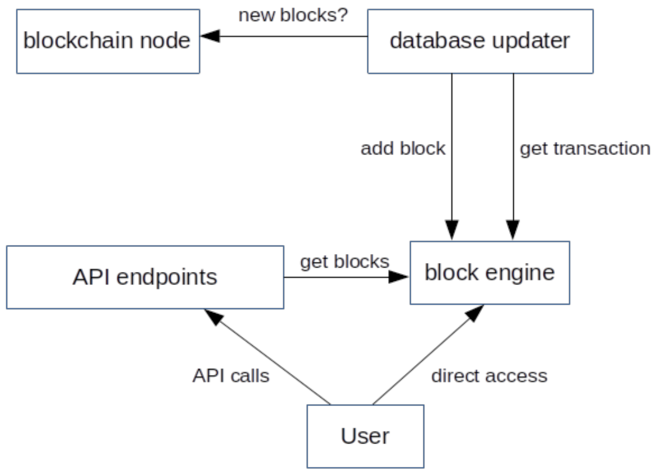

Fig. 1. The architecture of the Dru platform.

III. Using Dru for Research

This section is intended to describe the functionalities the Dru platform offers at the moment and how to use them. Moreover, the last two subsections demonstrate the use cases that have been basing on data provided by the Dru platform.

A. Block Engine

The block engine is a central repository for blocks. It is kept in sync with the blockchain node and as soon as new blocks will be obtained by the node, the block engine is populated with them. The access to blocks can be twofold: either by direct execution of no-SQL queries or by API endpoints described below. The former method is recommended for queries executed locally that return large amount of data, whilst the latter is for typical interactions with the Dru platform. API endpoints offer variety of functionalities that significantly extend typical no-SQL queries in terms of research. Moreover, only this approach is accountable, reproducible and provides the opportunity to download the results of past queries. However, the direct interface to the block engine can be used if researchers would like to use the engine for different purposes then the ones offered by API.

B. API Endpoints

The API endpoints are the most convenient way of working with Dru, especially for long-lasting queries. The platform provides multiple endpoints that can be grouped into three main categories: (i) providing blocks, (ii) network science, (iii) blockchain-specific.

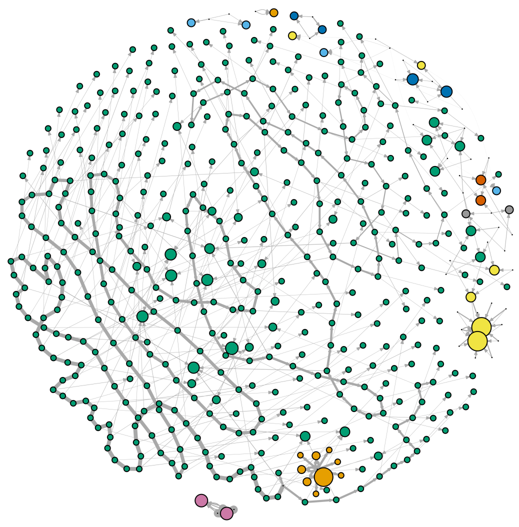

Fig. 2. The visualisation of blocks 1050-1060 of Zcash cryptocurrency. The size of a vertex is bound with its indegree, the width of edges is the amount transferred between nodes, colors indicate separate clusters. Data for this plot was gathered by using get_edges endpoint of the Dru platform.

The first category is an API-based way of obtaining the blocks from block engine. It is especially convenient when a user would like to parse the blocks in other ways then the ones provided by API endpoints. Dru offers the ability to provide blocks as they are stored in the block engine, including all fields or a reduced version of blocks striping the majority of data, but containing the essential information about height, transaction number, transaction partners and value.

The second group of endpoints can be considered as the core of the Dru platform. Their role is to consider the blockchain as a graph where transactions are linking the addresses. By using this graph it is possible to perform its analysis by means of network science. The most basic endpoint in this group allows to get the edges that exists in a blocks range provided as input (get_edges). This data can lead to performing analyses locally, if no endpoints for preforming specific operations are found in Dru. Another application of this endpoint can be visualisation, as depicted in Figure 2.

Other endpoints in this group allow to submit a variety of network science queries. These include, but are not limited to: computing the degrees of nodes, multiple centrality measures, such as closeness or betweenness, clustering coefficient (transitivity). Moreover, a number of endpoints for network-level measures is available, providing information about diameter, average shortest path or density of the network. In all these endpoints the block range is required to be provided as an input. This group is also accompanied by connectivity-related queries that for instance allow to find out whether there exists a path in a network based on blocks from a given range that connects two addresses in question. As most of the endpoints in this group can work with directed or undirected graphs, this is also a typical parameter to be provided when running these queries. Depending on its value, the graph the measures will be computed in either directed or undirected variants.

Lastly, the group for blockchain-specific queries allows to extract information specific for particular blockchains. For instance, for cryptocurrencies, this could be the amount of assets transferred between addresses (get_transactions_value). For one of the use cases presented below, we developed cryprocurrency-specific endpoint about the number of transactions providing different levels of privacy. Due to their nature, these endpoints most likely will be incompatible with different blockchain types, but the developers are always looking for the opportunity to make the endpoint as universal as possible.

C. *Working with Dru*

The installation and initial testing steps of the Dru platform have been described in the getting started guide[5]. These steps require installing Docker components and performing basic configuration tasks. It is suggested to install Dru on a separate always-on workstation so that the blocks will be always in sync with the state of the network and to provide a single instance of the platform among more researchers.

After installing and starting Dru, it downloads the blocks from a chosen blockchain node using RPC. Depending whether the node already was in sync with the blockchain or not, in the latter case, the process can be long-lasting. However, when the node already acquires some blocks, the block engine will populate these into the database, so the node does not need to in full sync for Dru to work with partial data.

When a user wants to access the blocks directly, he is encouraged to use MongoDB client to access the database and run the queries. In the case of the API endpoints, a specific endpoint needs to be called with required parameters. For instance, when using our demo Zcash Dru instance (see Section IV), if one is interested in acquiring first ten blocks of the Zcash blockchain, the query would be the following: http://dru.bergplace.org/api/get_blocks/0/9. After submitting the query, the platform will return the result URL where the status of a job and, when finished, final results will be available. We differentiate four types of statuses: queued (waiting in queue to be processed), running (processing), ready (results available) and error (an error encountered when processing the query). If the job will finish and have a *ready* status, this link will provide results in JSON format.

For long-lasting queries, Dru has a functionality to inform the user about status change of his query by sending an email. To use it, Dru configuration has to be provided with SMTP server parameters. Next, the user has to visit the main website of the platform and register its email. After confirming the address, each query can be accompanied by this address and when the results will be ready, a notification will be sent.

From the administrator's perspective, Dru offers two interfaces provided by the components used by the platform: MongoDB and Celery. These interfaces allow to control the state of the database and queues, and to look for anomalies. Celery can be also configured in such a way that some queries have higher priority than others.

[5]https://dru.readthedocs.io/en/master/docs/getting-started.html

D. Dru's Security

As Dru is a platform for monitoring and doing research on blockchain, it cannot perform any transactions in blockchain itself. However, it is highly encouraged to not to link the Dru platform to a blockchain node that is capable of performing transactions (e.g. has some assets). Dru also offers a number of restrictions that can be implemented, e.g. limiting the users to specific IP addresses or ranges to be able to use Dru or strict binding the blockchain node to Dru. Another security-related decision would be whether the Dru instance will be available over the Internet or not. It is not recommended to make it publicly accessible, so VPNs or similar solutions are recommended. However, we successfully host a Dru instance for more than a year so far without any problems related to stability or security. Depending on the requirements on auditing, Dru's components allow to fulfill different organizations' needs regarding logging: from limited to verbose one. Dru was audited by external security expert to identify and remove any potential security risks. The open-sourced nature of the project allows also the community to constantly audit the code.

E. Case Studies

The Dru platform is used for research on blockchain and two examples of research conducted by using the platform are presented below. These examples come from the domain of cryptocurrencies, but the application of Dru can cover other blockchains as well.

De-anonymising the type of nodes in blockchain. In [8] authors focused on answering the question whether is it possible to reveal the character of nodes in a Bitcoin cryptocurrency blockchain. In order to do so, they used Dru to gather data on blocks and built a network upon transactions. This network has been used to compute variety of features that became the input to the classifiers. Separately, authors tagged nearly 9,000 of addresses with their types, such as exchanges, mining pools, miners etc. The results demonstrate that it is possible to reveal the character of nodes in a blockchain by the use of supervised learning algorithms, even with a limited history of transactions.

Predicting the adoption of privacy-preserving transactions. Another direction of research, also connected to privacy issues, is studied in [9]. Authors used the Dru platform to study the adoption of different types of transactions and correlated it with important changes in the ecosystem in recent years. Next, they used Dru's endpoints to generate a set of features that has been used for supervised learning to answer the question whether is it possible to predict if certain address will become a part of a anonymous transaction solely basing on information available in blockchain. The results indicate that this is feasible and provides more insights into how to implement specific privacy-preserving countermeasures.

IV. Conclusion

This work describes the way of looking at the blockchain from the perspective of a complex network. We introduce Dru - the open-source platform for blockchain analysis. The role of this platform is to enable blockchain-based platform owners and researchers to look at blockchain from a network science perspective. This approach can lead to developing secure, functional, and scalable solutions. We believe that the applications of the platform are also to be found outside cryptocurrency-related blockchains, for instance to study and optimise distributed ledgers supporting business processes.

Researchers that are interested in working with the platform are highly encouraged to do so by starting with the documentation[6], while the source code is available for download[7]. The platform uses a GNU GPL license, and the block engine has a MIT license. An instance demonstrating the capabilities of the platform for the Zcash cryptocurrency is also available.

At this point, the platform has selected limitations. As it supports the generic RPC calls, it is compatible with multiple blockchains, yet so far seamlessly integrates with these that are based on the Bitcoin client. Moreover, if some blockchains have specific properties, these have to be separately considered in the API endpoints.

For future development, besides supporting multiple popular cryptocurrencies and other blockchain-based systems, the extension of the Dru platform to support cross-implementation queries is considered. Extending the set of functionalities of the API is also one of the key goals for the project at the current stage. Moreover, we anticipate to follow the path of discovering anomalies by using the Dru platform.

Acknowledgment

This work has been funded by Statutory Funds of Department of Computational Intelligence, Wrocław University of Science and Technology. The development of the Dru platform was funded by the Zcash Foundation.

References

[1] J. Abou Jaoude and R. G. Saade, "Blockchain applications–usage in different domains," *IEEE Access*, vol. 7, pp. 45 360–45 381, 2019.

[2] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on blockchain technology?—a systematic review," *PloS one*, vol. 11, no. 10, p. e0163477, 2016.

[3] P. Treleaven, R. G. Brown, and D. Yang, "Blockchain technology in finance," *Computer*, vol. 50, no. 9, pp. 14–17, 2017.

[4] K. Korpela, J. Hallikas, and T. Dahlberg, "Digital supply chain transformation toward blockchain integration," in *proceedings of the 50th Hawaii international conference on system sciences*, 2017.

[5] D. Daniel and C. Ifejika Speranza, "The role of blockchain in documenting land users' rights: The canonical case of farmers in the vernacular land market," *Frontiers in blockchain*, vol. 3, p. 19, 2020.

[6] I.-C. Lin and T.-C. Liao, "A survey of blockchain security issues and challenges." *IJ Network Security*, vol. 19, no. 5, pp. 653–659, 2017.

[7] D. Dasgupta, J. M. Shrein, and K. D. Gupta, "A survey of blockchain from security perspective," *Journal of Banking and Financial Technology*, vol. 3, no. 1, pp. 1–17, 2019.

[8] R. Michalski, D. Dziubałtowska, and P. Macek, "Revealing the character of nodes in a blockchain with supervised learning," *IEEE Access*, vol. 8, pp. 109 639–109 647, 2020.

[9] R. Michalski, "Analysing and predicting the adoption of anonymous transactions in cryptocurrencies," in *Business Information Systems Workshops*, W. Abramowicz and G. Klein, Eds. Cham: Springer International Publishing, 2020, pp. 132–144.

[6]https://dru.readthedocs.io

[7]https://github.com/bergplace/Dru