SIMBA: An Efficient Simulator for Blockchain Applications

Seyed Mehdi Fattahi, Adetokunbo Makanju, Amin Milani Fard
Department of Computer Science
New York Institute of Technology, Vancouver, Canada
{sfattahi,amakanju,amilanif}@nyit.edu

***Abstract*—Predicting the performance of a blockchain application during the design phase is difficult and evaluation after it is built could be expensive. The ability to simulate a blockchain network during the design stage in order to evaluate it is therefore a necessity. In this paper, we present a simulator for blockchain applications, called SIMBA (SIMulator for Blockchain Applications). SIMBA extends an existing simulator by adding the Merkle tree feature to blockchain nodes to improve efficiency and allowing more realistic evaluations not possible with the base tool to be performed. Results of our experiments show that the inclusion of Merkle trees has a high impact of up to 30 times reduction in the verification time of block transactions without an impact on block propagation delay. Since block verification is a critical part of the computational load of nodes on the network, this performance improvement significantly affects the overall performance of each node and consequently the entire network.**

***Index Terms*—Blockchain, simulation, Merkle tree, security**

I. Introduction

A blockchain is a growing list of records shared between participating parties, which is secured using cryptographic methods. The parties are nodes of a peer to peer network that shape a blockchain system or network. The records are transactions or a ledger of transactions stored in blocks. Each record in a block is verified by consensus of a majority of the members (parties or nodes) in the network. At the least, each block contains a cryptographic hash of the previous block, a timestamp, and transaction data. The data in blocks are immutable and cannot be altered as result of the protection offered by cryptographic hashes. Blockchain technology has gained widespread interest in industrial environments such as cryptocurrencies, e.g., Bitcoin and Ethereum, smart contracts, supply chain management, and intellectual property protection.

While evaluation is a key requirement in developing blockchain systems, experiments on the blockchain networks is challenging because a large number of nodes are necessary for realistic experiments and such experiments on blockchains are costly. For instance, there are over a million client nodes and more than 10,000 full nodes in the Bitcoin network [6]. Hence, simulation is an alternative way to evaluate a large scale system and it can be performed in a simulated setting with reasonable result at a low cost. Existing simulation tools, such as Bitcoin Simulator [4], Shadow-Bitcoin [5], and Minichain [7], create a model to describe network resources and also use a discrete-event simulation model. They do not, however, have the flexibility to extend the model, to simulate different blockchain systems, or create a new private blockchain system. BlockSim [3] as a simulation framework provides a tool for the design, implementation, and evaluation of blockchains. However, it does not simulate some features of real blockchain networks in particular Transactions Merkle Trees. A Merkle tree is a structure that provides a secure and efficient way to verify the consistency of a large group of data records. This structure improves the performance of transactions verification in blocks, and is a feature of most blockchain implementations such as Bitcoin and Ethereum.

In this paper, we present our extended version of Blocksim called SIMBA [2] that improves on the realism of simulations by including Merkle trees in each block, making it more efficient for transaction verification. We evaluate SIMBA by comparing block verification time and propagation delay using Merkle tree and without it and results show that using Merkle Trees can significantly improve the performance of nodes to verify the consistency of transactions in blocks.

II. System Design

Our prosed blockchain simulator, called SIMBA [2], is based on BlockSim [3]. BlockSim is an open-source blockchain simulator and is accessible via GitHub [1]. BlockSim provides a framework and a set of models common to blockchains. These models are extendable, when necessary, to evaluate design decisions. BlockSim uses a discrete-event simulation model that is suitable to model a blockchain system. Blockchain design evaluation with BlockSim consists of conceptualizing the underlying models i.e. block, transaction, network, messages, node; determining the input parameters for the models; checking if the conceptual model is accurate by comparing the simulated results with the measurements taken from a private Ethereum network; and measuring how long it takes to propagate a block and a transaction.

The inclusion of Merkle trees in blockchains plays a key role in the scalability of the network. However, it was not included in the BlockSim implementation in order to reduce its complexity. Cryptographic hash functions are the underlying technology that allows for Merkle trees to work. Merkle Trees are used in Bitcoin and Ethereum blockchain networks to verify the existence of a transaction in a way that conserves both space and time more effectively. Merkle Trees enable nodes to quickly verify that a given transaction is included in a particular block. Without using a Merkle Tree, the node has

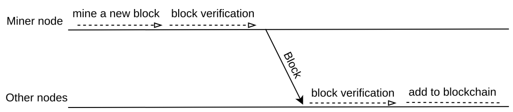

Fig. 1: Overview of the block verification process.

to load the entire block to verify that the transaction is part of the block. In a Merkle tree, each node is created by hashing the concatenation of its children. The tree can be constructed by taking nodes at the same height, concatenating their values, and hashing the result until the root is reached. Once created, data can be checked using only the root hash in logarithmic time to the number of leaf nodes. Auditing works by recreating the branch containing the piece of data from the root to the node containing the piece of data being verified.

Fig. 1 depicts how SIMBA verifies a block. We change some classes of BlockSim and added Merkle tree verification functionality to the "receive block bodies"' stage in the node class that is responsible for assembling the block header and the block body received and then insert it in the blockchain. Then we assess its impact on blocks' transactions verification time and block propagation delay.

III. Experiment Results

We evaluate two metrics of verification time and propagation time for a block. We selected these two metrics to demonstrate the performance gains of the inclusion of Merkle trees to the simulator and to demonstrate that their inclusion does not negatively impact the performance of the network. All simulations were conducted on a computer with 187 GB RAM and a 2.60 GHz Intel Xeon processor. We configured the simulation with the following parameters for an Ethereum network for 864,000 seconds duration time for all evaluations. We set the number of miner and non-miner nodes n to 50, 100, and 300, and block size (number of transactions in a block) s to 50, 100, 200, and 300.

Block Verification Time. We assess the impact of using Merkle tree on block verification time. Average verification time per block for $n = 50$ is shown in Fig. 2. For the sake of space we did not present results for $n = 100$ and 300 as it looks very similar in the case of average time. As shown in the chart, the average verification time per block without using Merkle trees is from 5 to 30 times higher than when using Merkle tree. Similarly Fig. 3 shows total verification time for different number of nodes. This leads us to conclude that irrespective of the number of nodes on the network (n), the amount of time spent in verifying blocks does not increase significantly for the network with Merkle trees while a linear increase can be expected without Merkle trees.

Block Propagation Time. Let $t_{i,j}$ and $t^*_{i,j}$ be the propagation time between node i and j in simulation without and with Merkle trees respectively. We compare the block propagation time of the two simulations by calculating the *mean absolute difference (MD)* as $\frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j=1}^{n}|t_{i,j}-t^*_{i,j}|$ for $i \neq j$. For this experiment we set number of nodes n to 300, and block size s to 100. The calculated MD value is less than 0.0001, that implies almost no change in the propagation delay between using Merkle tree and without using it. We therefore can state that the inclusion of the Merkle Trees does not significantly impact the performance of the network.

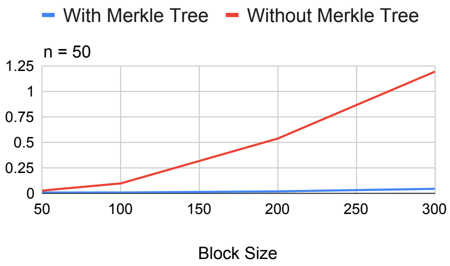

Fig. 2: Average block verification time (ms).

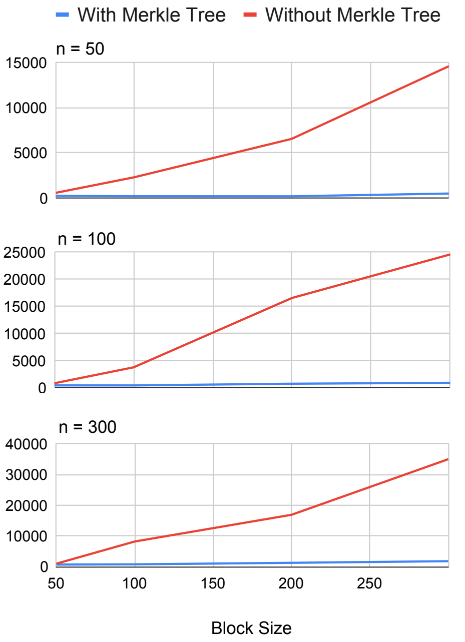

Fig. 3: Total block verification time (ms).

References

[1] A discrete event Blockchain Simulator. https://github.com/blockbirdLabs/blocksim, 2019.

[2] SIMBA: An Efficient Simulator for Blockchain Applications. https://github.com/nyit-vancouver/SIMBA, 2020.

[3] C. Faria and M. Correia. Blocksim: Blockchain simulator. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 439–446. IEEE, 2019.

[4] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16*. ACM Press, 2016.

[5] A. Miller and R. Jansen. Shadow-bitcoin: Scalable simulation via direct execution of multi-threaded applications. In *8th Workshop on Cyber Security Experimentation and Test*, 2015.

[6] S. Park, S. Im, Y. Seol, and J. Paek. Nodes in the bitcoin network: Comparative measurement study and survey. *IEEE Access*, 7:57009–57022, 2019.

[7] X. Wu, J. Yan, and D. Jin. Virtual-time-accelerated emulation for blockchain network and application evaluation. In *Proceedings of the 2019 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation - SIGSIM-PADS '19*. ACM Press, 2019.