Poster: Efficient Blockchain-based Software Systems via Hierarchical Bucket Tree

Weili Chen, Zibin Zheng,
Mingjie Ma
School of Data and Computer Science,
Sun Yat-sen University
Guangzhou, China
{chenwli9,mamj3}@mail2.sysu.edu.
cn,zhzibin@mail.sysu.edu.cn

Pinjia He
Computer Science and Engineering
The Chinese University of Hong Kong
Hong Kong, China
pjhe@cse.cuhk.edu.hk

Peilin Zheng, Yuren Zhou
School of Data and Computer Science,
Sun Yat-sen University
Guangzhou, China
zhengpl3@mail2.sysu.edu.cn,
zhouyuren@mail.sysu.edu.cn

ABSTRACT

As a promising technology, blockchain overcomes many shortages in traditional areas. However, the low efficiency prevents it from being widely used in practice. Through analyzing the transaction history in blockchain, we found that the account usage frequency is highly heterogeneous. Based on this observation, we propose a new account structure to improve the efficiency. The preliminary experiments show that the proposed structure has obvious advantages in efficiency compared with other account structures.

KEYWORDS

Blockchain, account structure, usage behavior

ACM Reference Format:
Weili Chen, Zibin Zheng, Mingjie Ma, Pinjia He, and Peilin Zheng, Yuren Zhou. 2018. Poster: Efficient Blockchain-based Software Systems via Hierarchical Bucket Tree. In *ICSE '18 Companion: 40th International Conference on Software Engineering Companion, May 27-June 3, 2018, Gothenburg, Sweden,* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, Article 4, 3 pages. https://doi.org/10.1145/3183440.3195009

1 INTRODUCTION

Blockchain is a new technology which influences many industries. A blockchain is a ledger of continuously growing batches of records called blocks that use cryptographic validation to link themselves together. Two characteristics, *Decentralization* and *append-only*, make blockchain solve the bottleneck of some industries. *Decentralization* means blockchain is maintained by independent peers and no one can control the whole system. And *append-only* means the information in blockchain cannot be changed. With the popularity of Bitcoin, many blockchain-based projects have appeared. Ethereum [3] and Hyperledger (fabric) [4] are the most famous among them. Ethereum is an open-source project which can implement smart contracts [5]. And Hyperledger is a project that is launched by the Linux Foundation and aims to advance cross-industry blockchain technologies.

Although blockchain has many potential applications, it still has many problems [6]. Low transaction processing speed is the most important one. For example, Visa can process 14 thousand transactions per second [2]. However, Bitcoin, the most famous blockchain-based application, can only process 7 transactions on average per second. In general, transaction processing speed depends on both *consensus stage* and *summary stage*. In the summary stage, a summary information of transactions is computed and used as a key ingredient in the consensus stage. However, the hash operation is heavily used in the summary stage. Thus, if we can reduce the number of hash operations, we will get a higher processing speed.

The number of hash operations in the summary stage depends on the account usage frequency and the account structure (i.e., the tree structure). In order to understand the account usage frequency, we collected all the transaction history of Ethereum and found that the usage frequency in Ethereum is highly heterogeneous. According to this observation, we propose a new tree structure to reduce the hash operations. The preliminary experiment shows that the proposed account structure can significantly reduce the number of hash operations.

The rest of the paper is organized as follows. Section 2 introduces three different account structures. Section 3 presents our discovery in account usage frequency and the proposed account structure. Section 4 shows preliminary experiment results and section 5 concludes the paper.

2 ACCOUNT STRUCTURE IN BLOCKCHAIN

A key technology in blockchain-based system is getting consensus on the current state of the ledger among all these decentralized peers. To this end, all accounts in the blockchain-based system need to be organized in a special structure and a summary information that represents the current state of the ledger needs to be computed. Generally, all accounts are organized in the leaf-nodes of a Merkle Tree (we call this account structure or tree structure) and the summary information is the root of the tree which is computed through hash operation from bottom to top. The account structure influences the number of hash operations to get the summary information and further affects the transaction processing speed in blockchain-based software systems.

Among these blockchain-based systems, Hyperledger uses Bucket Tree as the account structure. In Bucket Tree, which is also called *fabric tree*, the leaf-nodes are buckets, a container of multiple accounts. All buckets are arranged at the same height in Bucket Tree, thus it is a complete tree. To get the summary information when the

ICSE '18 Companion, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5663-3/18/05.
https://doi.org/10.1145/3183440.3195009

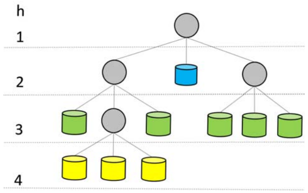

Figure 1: A Simple Hierarchical Bucket Tree

accounts (e.g., balance) changed by transactions: (1) the accounts in the same bucket serialized and hashed together; (2) a number of the hashed values (the number is predefined by the parameter) concatenate together and hashed to get a parent node. The second step executes repeatedly until the root of the tree constructed. The root hash value represents the new state of the ledger and will be used in consensus stage. This process makes the information in the system cannot be tampered with after it accepted by all peers.

Bucket Tree is more effective than Merkle Tree in Bitcoin [1] and Merkle Patricia Tree (MPT) in Ethereum. Firstly, Bucket Tree is fuller than MPT, so fewer hash operations are needed when transactions happen among accounts. Secondly, Hyperledger adds the notion of account into system, so a user can easily know how many *digital coins* he/she has while there is strictly no notion of account in Bitcoin. Finally, there are multiple accounts stored in one bucket. So when transactions happen, only one hash operation will be needed to get a hash value. In this paper, we proposed a new account structure and compared it with *fabric tree*.

3 ACCOUNT USAGE FREQUENCY AND HIERARCHICAL BUCKET TREE

Based on the account structures in Hyperledger, we take the account usage frequency into account to get a higher processing speed. We downloaded the Ethereum transaction ledger and extracted one-year transactions. We calculated the account usage frequency and found that 70% of the transactions happened in only the most used 1% accounts and the most used 5% accounts account for 86% of transactions in the sample period. Thus the account usage frequency is highly heterogeneous. It can be inferred that other blockchain-based systems can also draw similar conclusions.

Due to this observation, we propose a new account structure named as Hierarchical Bucket Tree (HBT). We arrange several different types of buckets according to their usage frequency. Different from other tree structures, in HBT, if a certain type of account is used more frequently, they will be placed closer to the root of the tree. Namely, we put different types of account in different layers of the tree, and the lower frequency of account, the deeper they will be stored. We can set the number of account types based on actual needs. As shown in Fig. 1, there are three types of accounts that are put in three different layers of the tree respectively.

4 EXPERIMENTS

To illustrate the efficiency of the proposed tree structure, we simulated a number of transactions and counts the corresponding hash operations. The results show that HBT sharply reduces the number of hash operations as compared with the fabric tree.

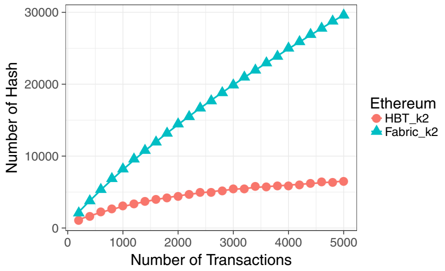

Figure 2: The Scatter Plot of Hash Counts

In our experiments, we set the number of accounts $N = 10^5$ and the number of forks of the tree $k = 2$, namely we built a binary tree which is similar to the Bitcoin system. Without loss of generality, we suppose that there are three different types of account in HBT. The ratio of each category is the same as Ethereum accounts usage frequency. Suppose that in a practical application, the most used category accounts for 80% transactions and the second most used category accounts for 15% transactions. Given for this information, the proportion of the number of accounts of each type can be calculated (for example, through simulation). Fig. 2 shows that the hash operations in HBT are much fewer than fabric tree when processing any number of transactions. Specifically, when the number of transactions is about 5000, the number of hash operations reduced by around 80%. Thus HBT is more effective than *fabric tree*.

5 CONCLUSION AND FUTURE WORK

Low efficiency of transaction processing in blockchain-based systems is an important problem. To overcome it, this study takes the impact of user behavior into account. As hash operations are heavily used in the system, we propose a new account structure based on the observation that the account usage frequency is highly heterogeneous. The preliminary experiment shows that the proposed account structure significantly reduces the number of hash operations. In the future, we will make sufficient experiments and strictly prove the efficiency of the proposed tree structure.

ACKNOWLEDGMENTS

The work described in this paper was supported by the National Key Research and Development Program (2016YFB1000101) and the National Natural Science Foundation of China (61722214, 61773410). Zibin Zheng is the corresponding author.

REFERENCES

[1] Andreas M Antonopoulos. 2014. *Mastering Bitcoin: unlocking digital cryptocurrencies.* "O'Reilly Media, Inc.".

[2] Long Chen. 2016. From Fintech to Finlife: the case of Fintech Development in China. *China Economic Journal* 9, 3 (2016), 225–239.

[3] Ethereum Foundation. 2014. Ethereum. (Aug. 2014). Retrieved August 2, 2017 from https://www.ethereum.org/

[4] Hyperledger. 2017. Hyperledger Announces Production-Ready Hyperledger Fabric 1.0. (11 July 2017). Retrieved August 2, 2017 from https://www.hyperledger.org/announcements/2017/07/11/hyperledger-announces-production-ready-hyperledger-fabric-1-0

[5] Nick Szabo. 1996. Smart contracts: Building blocks for digital markets. (Sept. 1996). Retrieved October 1, 2017 from http://www.fon.hum.uva.nl/

[6] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, and Huaimin Wang. 2016. Blockchain challenges and opportunities: A survey. *Work Pap.–2016* (2016).