# Large-scale Operation of Blockchain Nodes for Mission-critical Applications

*Abstract*—**Blockchain nodes constitute a blockchain network that stores blockchain data. The size of blockchain data continues to increase with the continuous operation of blockchains for nearly 10 years. As a result, enterprises using blockchain nodes for mission-critical applications must operate enough nodes to increase redundancy and operate stably. In this regard, this paper reports on the technology and practices of The "G" (anonymized). in the large-scale operation of blockchain nodes. The "G" (anonymized). has built a Kubernetes cluster of blockchain nodes operating stably to provide enterprises with commercial services. In addition, The "G" (anonymized). has implemented and practices several technologies related to Kubernetes clusters of blockchain nodes for stable service delivery. However, the synchronization process of large-sized blockchain data is time-consuming. To address this issue, the time required to start up a node was significantly reduced by automatically creating a snapshot of the volume containing the chain data. Moreover, a load balancing function was developed to indicate the block height of each node to ensure that the load balancer does not route client requests to nodes with significantly lower block heights than healthy nodes. Further, consistent hashing was adopted as its routing algorithm. This approach enabled an even distribution in allocating remaining nodes when reallocating to client nodes due to node failure.**

***Keywords—Blockchain, Kubernetes, Load Balancing, SRE***

## I. Introduction

### A. Background

#### 1) Birth of Blockchain

Blockchain is a method of recording data that emerged with the implementation of Bitcoin. Subsequently, the Ethereum blockchain was conceived, which could store program codes in blocks as well as transaction data. Program codes recorded in a block in Ethereum can be used to perform various processes, with their results recorded on the blockchain—referred to as smart contracts in Ethereum.

#### 2) Consortium Blockchain

Blockchain, as a leading emerging technology, has also been within the focus of enterprises. As Bitcoin is a technology for transferring money, the financial industry focused on it early. R3 LLC and its consortium were established in 2004 to research and develop the use of distributed ledgers in the financial industry. Major financial institutions from various countries have joined the consortium. In particular, R3 LLC developed Corda, a distributed ledger technology (DLT) inheriting from the blockchain concept. The Linux Foundation announced the creation of Hyperledger Projects in 2015. IBM and others developed Hyperledger Fabric. Numerous PoCs and projects, such as streamlining trade operations, were implemented using Hyperledger Fabric. However, participation in these two blockchain or DLT networks is restricted to a select group, i.e., only companies and those authorized by the consortium may participate. Therefore, this restricted group is referred to as the Consortium Blockchain or Permissioned Blockchain. The Consortium Blockchain contradicts the "trustless" blockchain, as written in the Bitcoin whitepaper and Gavin Wood's Web 3.0 [1]. The concept of "trustless" implies not trusting third-party organizations. Nevertheless, the Consortium Blockchain PoCs and projects have increased the understanding of blockchains among large companies, thus making Consortium Blockchain valuable in this respect. During this period, Ethereum, Hyperledger Fabric, and Bitcoin or Corda were called the three major Blockchains.

#### 3) Renewed Interest in Public Blockchains

In recent years, enterprises and governments have begun to pay attention not only to Consortium Blockchains but also to Public Blockchains. According to an Ernst & Young survey, 75% of respondents expected their organizations to implement solutions related to Public blockchain solutions in the future [2]. Shin'ichiro Matsuo, a Research Professor at the Department of Computer Science at Georgetown University and one of the most prominent researchers in the blockchain field, said in a report issued by the Japanese Ministry of Economy, Trade and Industry (METI) in 2022, that "The private type is no different than the timestamp technology that emerged in the 1990s, and it is meaningless unless it is public" [3]. Significantly, this expert opinion appeared in a report published by a ministry under the influence of the government's will.

One factor that increases the complexity for enterprises to implement Public Blockchains for business is scalability. The problem with Public Blockchains' scalability is that an increase in transactions results in miners taking too long to record their transactions. Ethereum developers aimed to solve scalability by moving the Consensus Algorithm from PoW (Proof of Work) to PoS (Proof of Stake). Another approach involved developing new Public Blockchain Platforms compatible with the Ethereum Virtual Machine (EVM), Ethereum's smart contract execution environment. Other emerging platforms compatible with the EVM include Polygon, which launched in 2019 as Matic Network and rebranded in February 2021, and Avalanche, which launched in 2020.

Thus, enterprises have increasingly focused on using Public Blockchains, especially after 2020. Various public blockchain platforms are flourishing today instead of the three major blockchains of the past. Given this background, the challenges for enterprises to decide to operate their own Public Blockchain nodes are high because it is rare that enterprises have engineers able to use blockchain technology, which is advancing rapidly. One viable option for enterprises is using the services of specialized companies with a high level of expertise in operating blockchain systems infrastructure.

### B. Purpose

Enterprises require a highly stable system infrastructure to use Public Blockchains such as Blockchain Nodes. Further, blockchain system infrastructures are unique in ways that general system infrastructures are not. Therefore, novel insights are required when configuring redundant configurations of Blockchain Nodes.

The “G” (anonymized). is one of the leading startups in the Blockchain field that has raised funds from prominent VCs and VCs from government-affiliated financial institutions, and provides stable Blockchain node services to enterprises. Many enterprises have adopted their services because they have provided services for numerous blockchain platforms without major system failures. They are able to achieve this result because they have developed and applied their own unique technology to build a redundant configuration of Blockchain nodes and ensure stable operation.

This paper describes the approach and implementation adopted by The “G” (anonymized). to build redundant configurations of blockchain nodes and provide stable node services. Sections II to IV describe the problems to be solved and the solutions developed. Section V provides a summary and future perspectives.

## II. Technology and Practices for Accelerating Blockchain Node Rescheduling

### A. Problems

#### 1) Increasing Chain Data Size

The chain data size of Public Blockchains continues to increase over the operating years. For example, the official site of go-Ethereum (Geth), the leading Ethereum Node Software, describes, "*With default cache size, the database grows by about 14 GB/week. This means that Geth users will rapidly run out of space on 1 TB hard drives*" [4]. Geth's chain data size reached nearly 1200 GB on 26 January 2022 [5]. Therefore, pruning was performed, reducing the chain data size to the 500 GB range [5]. "*Pruning is the process of erasing older data to save disk space. Since Geth v1.10, users have been able to trigger a snapshot offline prune to bring the total storage back down to the original 650 GB in about 4-5 hours*" [4]. However, the chain data size is still steadily increasing even after pruning. After the pruning in February 2023, Geth's chain data size exceeded 870 GB; as of August 2023, it again exceeded 1.1 TB [5]. With this increase in chain data size, the operation load of blockchain nodes increases, thus making it increasingly challenging for non-professional companies to operate nodes.

#### 2) Longer Synchronization Process for Chain Data

The synchronization of chain data is time-consuming due to CPU resource bottlenecks and disk IO used by the node. Full sync, one of the earliest synchronization methods, takes between one week and ten days for synchronization, even on a high-performance machine [6]. Fast sync, introduced after full sync, requires more than 10 h for synchronization due to network communication bottlenecks [6]. Even using the newest and least time-consuming method, snap sync (which has some tradeoffs [6]) requires more than 2 h [7].

#### 3) Challenges in Operating Node Clusters with Large Amounts of Chain Data

IT systems need to have redundant configurations to provide stable services. Currently, a typical approach is providing redundancy with container-based virtualization, such as Docker, and container orchestration tools like Kubernetes. When a redundant configuration is created by containerizing the node software comprising the blockchain network, the chain data is stored in a volume in the pod where the node software is running. Upon the failure of one worker node in the cluster's node pool, the pod running on that worker node is rescheduled to another worker node to maintain the number of replicas. As discussed above, the amount of chain data is increasing, and synchronization is time-consuming. Therefore, some action is required to rapidly complete the process of rescheduling pods.

### B. Solution

This section describes the solution that The “G” (anonymized). built to solve the problem described in the previous section. Figure 1 outlines the developed functions.

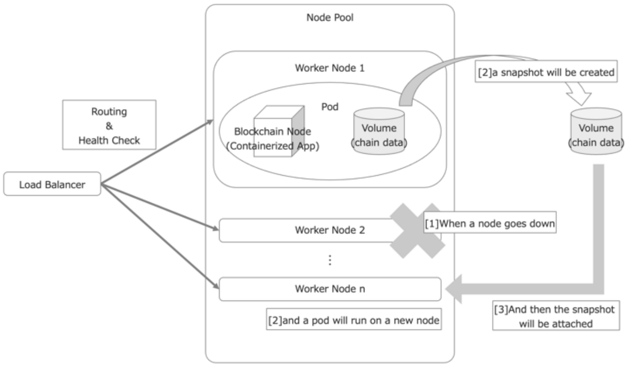

Fig. 1. Overview of Accelerated Blockchain Node Rescheduling

#### 1) The “G” (anonymized). Approach

The outline of the developed approach is described below. The functionality implemented automatically creates a snapshot of the volume where the chain data of the healthy worker node is stored. Subsequently, the volume snapshot is attached to the newly launched node.

#### 2) Details of Creating Chain Data Volume Snapshots

The “G” (anonymized). independently implemented a load balancer accepting requests from clients who want to use blockchain nodes. The health check function of this load balancer has special features required for a Kubernetes cluster of blockchain nodes. If the health check function detects an abnormality in a worker node in a node pool, it automatically creates a snapshot of the volume where the chain data in the pod of the healthy worker node is stored. The Kubernetes function also deploys the blockchain node pod on another worker node. After deploying the Pod, the Pod newly mounts the Volume where the snapshot is saved as its own Disc Volume. In this way, Blockchain Nodes can be launched quickly.

### C. Achievement

With this solution, The “G” (anonymized). can significantly reduce the time required to reschedule pods and achieve stable operation of Blockchain nodes. Even using snap sync described in II-A- (2), our measurement took approximately 4 h. This method, in contrast, requires less than 1 h, significantly reducing the time required.

## III. Development of a Function to Check the Status of the Latest Block Holdings

### A. Problem

A situation may arise where the blockchain node cannot add the latest block immediately. In this case, the blockchain node cannot return the latest status of the blockchain to the client who made the request. Routing requests from clients should not be sent to such a node, as it will not correctly function as a blockchain node. Therefore, whether the chain data of each blockchain mode in the node pool has the latest block must be checked.

### B. Solution

This section describes the solution that The “G” (anonymized). built to solve the problem described in the previous section. Figure 2 shows an overview of the developed functions.

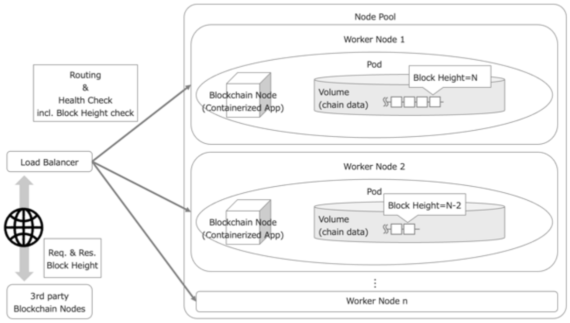

Fig. 2. Overview of the Latest Block Holding Status Check Function

#### 1) The “G” (anonymized). Approach

To address the limitations mentioned above, The “G” (anonymized). has improved the health check function of the load balancer that accepts client requests. Typical load balancers for Kubernetes clusters cannot check the volume contents of each pod. In particular, the implemented chain data check function compares the chain data stored in the volume of the blockchain nodes they are running with the chain data held by the third-party blockchain nodes. "third-party blockchain nodes" here refers to companies like Infura[8] that provide Blockchain Nodes as a service to companies and individuals. This comparison is intended to confirm that the Block Height is accurate. If anomalies are detected by the chain data check function due to the comparison, the load balancer stops routing to that node.

#### 2) Details of the Chain Data Check Function

The chain data check function checks the Block Height and Block Hash of the chain data of the blockchain nodes that The “G” (anonymized). operates. Block Height is a numerical value representing the number of blocks in the blockchain that the latest block has. By comparing the Block Height of their blockchain nodes with the value of the blockchain nodes that the third party holds, the function determines whether they have the latest block. Block Hash is a unique value that each block has. By comparing the Block Hash of the latest block with the value held in third-party blockchain nodes, the function can check whether the contents of the latest block are correct. The chain data check function can be used to stop routing to nodes whose block height is significantly lower than the others or has incorrectly added nodes. If routing to a node is stopped, pod rescheduling is automatically performed to ensure redundancy, as described in Section II.

The time interval for automatically performed Block Height check varies depending on the target blockchain platform. The time to generate a block is called Block Time. For Blockchain Platforms with long Block Times, checks are performed at relatively long intervals. However, checks are implemented at intervals of at most 1 min. For blockchain platforms with short block times (for example, a few seconds), checks are performed at a short interval equivalent to the block time. Moreover, the threshold in the number of blocks where routing is stopped if a difference is detected depends on the specifications of the target blockchain platform. Blockchains with short Block Time and fast-increasing Block Height do not tolerate relatively minor differences.

### C. Achievement

The “G” (anonymized). has deployed commercial services for blockchain nodes using Kubernetes clusters and load balancers with this special feature in more than 20 blockchains with different Block Times. Consequently, stable services are provided by maintaining a node pool consisting solely of normal blockchain nodes, as described above.

## IV. Method of Matching the Client to Node Receiving the Request for Each Request

### A. Problems

#### 1) Minor Differences in the Availability of Latest Blocks of Blockchain Nodes

The status of holdings of the latest blocks of blockchain nodes may differ slightly for each node because of the potential slight variation when blockchain nodes receive the latest blocks and complete the process of adding them to the top of the chain.

#### 2) Necessity of Matching Client and Node Receiving the Request for Each Request

As noted above, the status of blocks near the top of the chain may differ among different blockchain nodes. Therefore, the requests from one client to the same blockchain node for each request should be routed. In particular, routing to a different blockchain node for each request may result in inconsistencies in the contents of the latest block compared to those of previous requests.

### *B. Solution*

This section describes the solution that The "G" (anonymized). built to solve the problem described in the previous section. Figure 3 shows an overview of the developed functions.

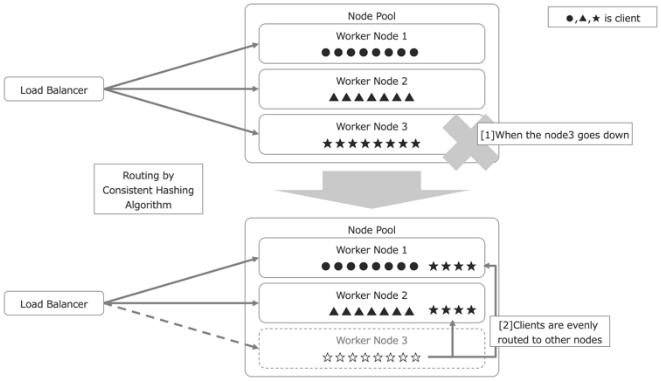

Fig. 3. Equalization of Client Rescheduled Nodes

#### *1) The "G" (anonymized). Approach*

As the routing algorithm for requests from clients to nodes, we used "Consistent Hashing"[9], a well-known method for evenly and stable allocation.

#### *2) Details of Routing by Consistent Hashing*

The "G" (anonymized) determines the routing destination node based on the Client ID assigned to clients. When nodes are reallocated due to a communication failure, etc., by using Consistent Hashing, the reallocation to the remaining nodes can be done evenly without any bias. Even when the number of nodes recovers, unbiased and even allocation is performed again. Routing clients evenly to nodes contributes to stable services.

#### *3) Achievement*

Adopting a Routing Algorithm based on Consistent Hashing, The "G" (anonymized). can stably respond to forced scale-ins when the worker node fails and scale-outs when the number of clients increases. As a result, services of The "G" (anonymized) meet high service standards and are used by more than 20 enterprises as commercial services.

## V. Conclusion

This paper describes The "G" (anonymized).'s approach and implementation to build a redundant configuration of blockchain nodes in a Kubernetes cluster to provide stable node services. With The "G" (anonymized). supporting more than 20 Blockchain Platforms, many enterprises are using their node services, with stable operations reported.

However, there is still room for further improvement to increase the stability of their services. For instance, while the current routing algorithm assigns Clients' IDs and routes them to ensure that the number of clients is equal for each node, in reality, the number of requests made by clients is not equal. The stability of node services can be further improved by having the number of requests be equal on each node instead of the number of clients. A possible solution to improve this point can be an approach recording the request per second (RPS) for each client and accounting for it when determining the node to route.

## Acknowledgment

We would like to thank Editage (www.editage.com) for English language editing.

## References

[1] Gavin Wood, "ĐApps: What Web 3.0 Looks Like", gavwood.com, April 2014, https://gavwood.com/dappsweb3.html (accessed on December.14.2023)

[2] Paul Brody, "How public blockchains are making private blockchains obsolete", ey.com, December 2019, https://www.ey.com/en_us/innovation/how-public-blockchains-are-making-private-blockchains-obsolete (accessed on December.14.2023)

[3] Ministry of Economy, Trade and Industry, "Concept of Web3.0 Business Environment Improvement", Ministry of Economy, Trade and Industry, December 2022, https://www.meti.go.jp/shingikai/sankoshin/shin_kijiku/pdf/010_03_01.pdf (accessed on December.14.2023)

[4] Etherscan, "Ethereum Full Node Sync (Default) Chart", Etherscan, January 2019, https://etherscan.io/chartsync/chaindefault (accessed on December.14.2023)

[5] The go-ethereum Authors, "Pruning", go-ethereum, April 2023, https://geth.ethereum.org/docs/fundamentals/pruning (accessed on December.14.2023)

[6] Shigeyuki Azuchi, "New sync method Snap sync introduced in Geth v1.10.0", Develop with pleasure!, March 2021, https://techmedia-think.hatenablog.com/entry/2021/03/21/165151 (accessed on December.14.2023)

[7] holiman, "Ethereum Snapshot Protocol (SNAP)", GitHub, November 2020, https://github.com/ethereum/devp2p/blob/master/caps/snap.md (accessed on December.14.2023)

[8] INFURA INC, "Web3 Development Platform _ IPFS API & Gateway _ Blockchain Node Service", INFURA INC, January 1900, https://www.infura.io/ (accessed on December.14.2023)

[9] Juan Pablo Carzolio, "A Guide to Consistent Hashing", Developers, April 2017, https://www.toptal.com/big-data/consistent-hashing (accessed on December.14.2023)