Journal of Computer Science and Frontier Technologies
ISSN Print: 3104-4204
ISSN Online: 3104-4212
CD ISSN: 3104-4220
https://doi.org/10.63313/JCSFT.9018

Chord Ring Distributed Storage Method for Ethereum Data

Lijie Ma
Xi'an Shiyou University, Xi'an 710000, China
Email: 3228281841@qq.com

How to cite this paper: Ma, L. (2025). Chord Ring Distributed Storage Method for Ethereum Data. Journal of Computer Science and Frontier Technologies, 1(2), 87–94. ISSN Print: 3104-4204; ISSN Online: 3104-4212.
https://doi.org/10.63313/JCSFT.9018
Published: 2025-10-27

Abstract

To address the limitations of blockchain data storage capacity and uneven dis-tribution, this paper proposes a Chord dual-ring distributed storage method based on virtual nodes. Building upon the original Chord protocol, this approach introduces virtual rings to construct a "storage ring-virtual ring" dual-ring structure. Target virtual nodes are located through routing table lookups, and data is distributed across the storage ring via a name mapping mechanism. Sim-ulation experiments validate the proposed scheme's effectiveness by evaluating load bal-ancing and query success rate across varying sharding granularities. Results demonstrate that this approach not only efficiently achieves sharded storage for blockchain data but also ensures balanced distribution of block data.

Keywords

Storage capacity; Balance; Chord dual-ring structure

1. Introduction

With the development of the digitial economy,mainstream blockchain systems represented by Bition face severe storage scalability challenges[1].The data capacity of full nodes has exceeded 500GB,leading to inefficient data synchronization.Therefore.overcoming blockchain data storage bottlenecks has become one of the core critical issues in this field.Currently, [2]sharding technology has emerged as an effective solution to address blockchain storage capacity constraints.Several representative projects, such as Monoxide[3], OmniLedger[4], and Ethereum 2.0[5][6], adopt sharding structure to tackle storage limitations.In Monoxide, the network is partitioned into multiple parallel consensus groups as shards. Accounts are mapped to corresponding shards based on the first byte of their addresses to achieve state sharding. OmniLedger assigns validators to fixed shards using distributed random numbers, with accounts similarly allocated to shards based on their addresses. Ethereum 2.0 employs a beacon chain to randomly assign validators to shard committees. Each committee validates blocks through data availability sampling, using mechanisms like content hashes to index data storage locations. While these sharding solutions effectively alleviate storage capacity constraints, they generally fail to resolve load balancing issues across shards. This imbalance may cause excessive storage load on certain nodes, impacting overall system performance.

Meanwhile, with the rapid advancement of current blockchain technology and distributed storage systems, structured peer-to-peer networks face new challenges in data storage and retrieval. The Chord protocol, proposed by Stoica et al[7]. in 2001, aims to achieve efficient, deterministic resource lookup in fully distributed environments. [8]Each node within the Chord ring is treated as a natural shard, offering potential for storage scalability. However, its data distribution method based on consistent hashing often leads to load imbalance among nodes[9], limiting its practical application in real-world scenarios.

To address the aforementioned issues, this paper proposes a Chord dual-ring structure based on virtual nodes. Its core contributions are threefold. First, by constructing a Chord dual-ring structure, [10]virtual nodes are introduced to form virtual rings on top of the classic Chord ring. A name mapping mechanism establishes associations between storage nodes and virtual nodes, thereby enhancing the system's scalability and adaptability. Second, a corresponding data storage method is designed in conjunction with the virtual node mechanism. After locating the target virtual node via the routing table, data is actually stored on the corresponding physical storage node, achieving balanced load distribution across storage nodes. Finally, simulation experiments validate the system performance, evaluating load balancing effectiveness under different data sharding granularities and query success rate under varying node failure probabilities. The results confirm the structure's advantages in load balancing and system reliability.

2. Blockchain Data Sharding Structure Based on the Chord Double-Ring Structure

2.1. Chord Dual-Ring Construction

This paper analyzes the classic Chord protocol and introduces virtual nodes to distribute data relatively evenly across storage shards[11]. It then implements efficient data storage based on routing tables.

The storage ring is constructed as follows:

- Set the hash space bit length of the ring to m, with a hash space size of 2^m, forming a closed hash ring with a range of [0, 2^m-1]. Each storage shard node is assigned a name $node_i$(i=0,1,...,n-1) and its IP address;
- Apply the FNV-1A hash algorithm to compute a fixed-length hash of each node's IP address. Take the hash modulo 2^m to generate a unique identifier ID for each node;
- Sort all storage shard nodes in ascending order by their ID values to form the

logical Chord ring topology;

- Within this storage ring, each node's predecessor is defined as the nearest node counterclockwise on the ring, while its successor is the nearest node clockwise. This establishes a complete circular routing network.

The virtual ring is constructed as follows:

- The hash space of this ring is identical to that of the storage ring. Virtual nodes are named $node_{ij}$(j=0,1,...,j-1), establishing a mapping relationship with storage shard nodes through their names.
- The IP address of a virtual node is generated by appending the suffix "#s" (s=0,1,2,3...) to the IP address of its corresponding storage shard node. The hash value for the virtual node's IP is computed using the same FNV-1A hash algorithm, and then modulo 2^m is taken to obtain the virtual node's identifier ID. In case of ID conflicts, the system recalculates by incrementing the s value to ensure the uniqueness of all virtual node IDs.
- All virtual nodes are arranged in ascending ID order to form a logically meaningful virtual ring.
- Within this virtual ring, each node's predecessor is defined as the nearest node counterclockwise on the ring, while its successor is the nearest node clockwise.
- Each virtual node maintains a routing table containing m entries. The successor node for the i-th entry (i=0,1,..., m-1) dynamically determines its successor node via the formula $key_i = (ID + 2^i) \bmod 2^m$. Here, key_i represents the computed key value for the i-th routing entry. The first virtual node with a key value greater than or equal to this value becomes the successor node for that routing entry. The routing tables of all virtual nodes are stored in the corresponding storage shard node routing tables.

2.2. Data Storage

Based on the dual-ring system structure designed in this paper, the specific workflow for data storage operations is as follows:

- When a client needs to store data, it first computes the SHA-256 hash algorithm on the entire data set to generate a fixed-length hash value.
- By performing a modulo operation on the hash space 2^m, the HashMod value for the data in the identifier space is obtained. This value serves as the unique locator identifier for the data within the identifier space.
- The system matches the computed HashMod value against the ascending sequence of virtual nodes $\{id_0, id_1, id_2, ..., id_{v-1}\}$ within the virtual ring. Using the routing table, it rapidly locates the smallest virtual node id_j satisfying $id_j \geq$ HashMod.
- Based on the pre-established mapping between virtual nodes and storage shard node names, the system persists the data and its hash value to the storage shard node corresponding to the target virtual node.

This process achieves balanced data distribution, effectively resolving the potential load imbalance issues present in traditional Chord protocols.

3. Simulation Experiments and Results Analysis

3.1. Dataset

In this study, we utilized the real blockchain dataset provided by the IEEE CSCloud 2023 conference as the experimental foundation. This dataset comprises 1,048,575 complete transaction records. Each transaction record comprises three key fields: sender address, recipient address, and transaction amount. To ensure data quality and validity, we performed rigorous preprocessing on the raw dataset: removing anomalous records containing null values, eliminating invalid transactions with identical sender and recipient addresses, and discarding redundant entries with identical hash values. Following this data cleansing process, we ultimately selected 500,000 valid transaction records meeting our criteria as the experimental sample, establishing individual transactions as the fundamental data storage unit.

3.2. Evaluation Indicators

In this experiment, two metrics were employed to evaluate the system's performance: the proportion of transaction volume under different sharding granularities and the query success rate under varying downtime probabilities.

3.3. Experimental Results and Analysis

3.3.1. Sharded Transaction Volume Share

To evaluate the effectiveness of introducing a dual-ring structure with virtual nodes and its storage strategy in improving transaction equilibrium, this experiment adopts the proportion of sharded transaction volume as a key evaluation metric. The experiment sets the number of shards c to 10, 15, and 20, with virtual nodes v counts at multiple gradients of 0, 10, 50, 100, 150, 250, and 300 to investigate the specific impact of varying virtual node scales on transaction distribution equilibrium. Table 1 presents the transaction volume distribution under different virtual node counts across various sharding granularities.

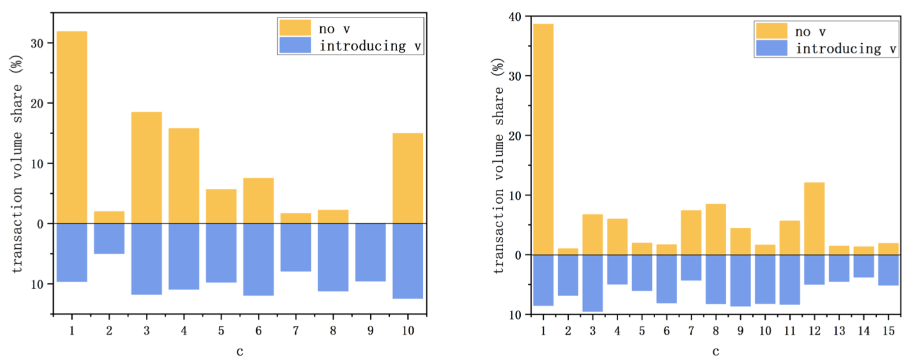
Table 1. Transaction volume distribution under different virtual node counts across various sharding granularities.

v	c=10			c=15			c=20		
	Max	Min	Var(10^{-4})	Max	Min	Var(10^{-4})	Max	Min	Var(10^{-4})
0	31.8%	0.1%	102.9	38.6%	0.9%	88.9	19.6%	0.1%	26.6
10	46.6%	2.4%	172.6	16.6%	1.6%	16.4	12.5%	2.1%	7.3

v	c=10			c=15			c=20		
	Max	Min	Var(10^{-4})	Max	Min	Var(10^{-4})	Max	Min	Var(10^{-4})
50	15.3%	5.0%	12.0	12.1%	3.0%	7.0	7.4%	3.0%	2.1
100	16.1%	5.2%	11.2	10.9%	3.5%	4.5	7.2%	2.5%	1.9
150	12.9%	5.2%	7.7	9.5%	3.7%	3.7	8.7%	2.5%	1.8
250	12.6%	5.2%	5.1	10.8%	3.4%	4.8	6.5%	3.5%	0.8
300	12.4%	4.9%	4.9	11.2%	3.6%	5.6	6.1%	3.2%	0.7

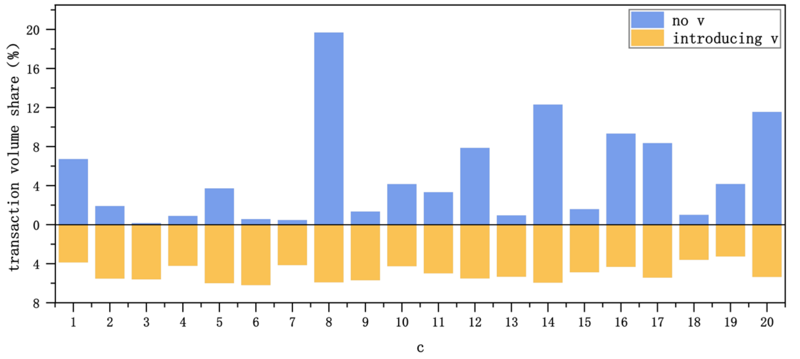
As shown in Table 1, the introduction of virtual nodes effectively mitigates transaction distribution imbalance, with the maximum transaction volume share and minimum transaction volume share progressively converging. When the sharding granularity is 10 or 20 and the number of virtual nodes is 300, the maximum and minimum transaction volume shares are closest, with the smallest variance, indicating the most balanced data distribution. When the sharding granularity is 15 and the number of virtual nodes is 150, the maximum and minimum transaction volume shares are closest, with the smallest variance, indicating the most balanced data distribution.

To visually demonstrate the optimization effect of virtual nodes on transaction volume distribution, the experiment compared the transaction volume distribution without virtual nodes versus with introduced virtual nodes. The number of introduced virtual nodes was selected according to Table 1. Figure 1 shows the sharded transaction volume share.



(a) c=10

(b) c=10



(c) c=20

Figure 1. Sharded transaction volume share.

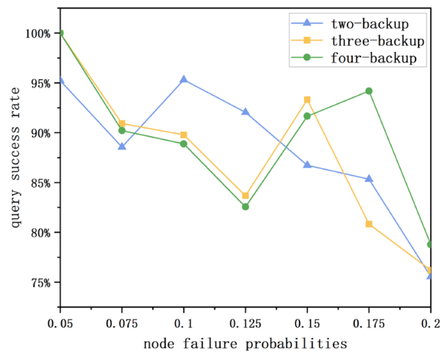
As shown in Figure 1, without introducing virtual nodes, transaction volume share remains uneven when sharding granularity c is set to 10, 15, or 20. As the sharding granularity increases, the difference between the maximum and minimum values decreases. This indicates that even without virtual nodes, increasing the number of shards can marginally improve transaction volume distribution. This occurs because a higher number of shards reduces the disparity between them, allowing each shard to handle a more refined data scope.

At a sharding granularity of 10, transaction volatility ranges as high as 31.86% and 0.03%, indicating severe sharding load imbalance. After introducing virtual nodes, the difference between maximum and minimum values progressively decreases, with values of 7.43, 7.61, and 2.94 across different sharding granularities, showing overall uniformity. This occurs because each storage shard node is mapped to multiple virtual nodes. These virtual nodes are arranged sequentially, enabling data to be evenly distributed across all shards. This effectively prevents the emergence of hotspot shards and achieves balanced transaction load distribution.

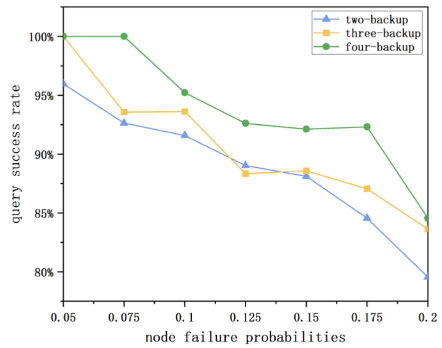
This experiment validated the effectiveness of the virtual node mechanism in improving transaction balance through variance analysis and transaction volume distribution comparisons. The results clearly demonstrate that introducing virtual nodes can effectively mitigate the load imbalance issue inherent in traditional consistent hashing algorithms, providing a reliable technical approach for load balancing optimization in distributed storage systems.

3.3.2. Query Success Rate

To investigate the impact of different redundancy strategies on system fault tolerance, this experiment compares the dual-ring structure with the classic Chord ring. The experiment sets the number of shards c to 50, 300 virtual nodes (determined using the method from Experiment 1), and node failure probabilities ranging from 5% to 20%. Figure 2 shows the query success rate for both structures under different redundancy strategies.



(a) classic Chord ring



(b) dual-ring structure

Figure 2. Query success rate

As shown in Figure 2, both the dual-ring structure and the classic Chord ring structure exhibit an overall downward trend in query success rate as node failure probability increases. As shown in (a), although the overall trend also declines with increasing failure probability, the curves for each backup strategy exhibit significant fluctuations. Particularly within the failure probability range of 0.125 to 0.175, both the four-backup and three-backup strategies show a marked decrease. The four-backup strategy performs best at low failure probabilities but lacks stability. In contrast, the two-backup strategy exhibits a relatively smooth curve; however, at a failure probability of 0.2, its query success rate drops to approximately 75%, significantly lower than the performance of the same-level backup in the dual-ring structure.

As shown in (b), the four-backup strategy delivers the best performance, maintaining approximately 85% query success rate even at a failure probability of 0.2. The three-backup strategy follows closely, achieving about 84% success rate under the same conditions. Even the two-backup strategy, with the fewest backups, maintains a query success rate close to 80% at the maximum failure probability. This demonstrates that increasing the number of backups in a dual-ring structure effectively enhances the system's fault tolerance.

In summary, the data redundancy strategy employed by the dual-ring structure significantly enhances the system's fault tolerance, with the number of redundant backups positively correlated to query success rate. Compared to the classic Chord ring structure, the dual-ring structure demonstrates more stable performance characteristics and higher query success rate across varying failure probabilities, validating its advantages in improving system reliability.

4. Conclusion

This paper addresses the issue of limited data storage capacity and uneven distribution in blockchain systems by proposing a Chord dual-ring distributed storage method tailored for Ethereum data. It introduces a virtual node layer to construct a "storage ring-virtual ring" collaborative structure. Through a name mapping mechanism between virtual nodes and storage nodes, this approach achieves balanced data distribution at the storage layer while preserving Chord's efficient routing advantages. Experiments demonstrate that virtual nodes effectively regulate data distribution, significantly reducing variance in transaction volume proportions across different sharding granularities. Fault tolerance comparisons confirm the stability of redundancy strategies within the dual-ring structure, where query success rate positively correlate with redundancy levels. Under identical failure probabilities, this structure outperforms classical structures in both query success rate and stability. While current data is limited to Ethereum transaction records, future work will extend this method to broader blockchain data sharding scenarios.

References

[1] Liu Minghao.Research on Blockchain Storage Expansion Solution[D].YanTai University,2025.DOI:10.27437/d.cnki.gytdu.2025.000109.

[2] Jiang Lingyun, Yang Jinglin, Ma Pengcheng, et al. A Review of Blockchain Scalability Research Based on Sharding Technology [J/OL]. Acta Electronica Sinica, 1-22 [2025-10-22]. https://doi.org/10.12263/DZXB.20241117.

[3] Wang and H. Wang, "Monoxide: scale out blockchains with asynchronous consensus zones," in Proc. 16th USENIX. Symp. Net. Systems. Design. Impl. (NSDI'19). Boston, MA:USENIX Association, Feb. 2019, pp. 95 – 112.

[4] Syta and B. Ford, "OmniLedger: a secure, scale-out, decentralized ledger via sharding," in Proc. IEEE. Symp. Secu. Priv. (SP), San Francisco, CA, USA, 2018, pp. 583-598.

[5] M. Cortes-Goicoechea, L. Franceschini and L. Bautista-Gomez, "Resource analysis of ethereum 2.0 clients," in Proc. 3rd Conf. Blockchain. Resear. & Appli. Inno. Net. Ser. (BRAINS), Paris, France, 2021, pp. 1-8. Giambastiani, B.M.S. (2007) Evoluzione Idrologica ed Idrogeologica Della Pineta di san Vitale (Ravenna). Ph.D. Thesis, Bologna University, Bologna.

[6] F. Cassez, J. Fuller and A. Asgaonkar, "Formal verification of the ethereum 2.0 beacon chain," in Tools and Algorithms for the Construction and Analysis of Systems, Cham, 2022:Springer International Publishing,pp.167-182.Honeycutt,L.(1998) Communication and Design Course.

[7] Stoica, Ion, et al. "Chord: A scalable peer-to-peer lookup service for internet applications." ACM SIGCOMM computer communication review 31.4 (2001): 149-160.

[8] Kong Ruiping. Research on Resource Location in P2P Networks Based on the Chord Routing Algorithm [J]. Computer Products and Circulation, 2020, (08): 64-65. DOI: CNKI:SUN:WXXJ.0.2020-08-060.

[9] Liu Minghao, Hong Jianlei, Wang Chengxiang, et al. Blockchain Storage Optimization Strategy Based on Consistency Hashing [J/OL]. Journal of Computer Applications, 1-8 [2025-10-22]. https://doi.org/10.11772/j.issn.1001-9081.2024121836.

[10] Wang Yuanlin, Ouyang Yong, Gao Rong, et al. Self-Supervised Contrastive Academic Warning Model Based on Virtual Node Strategy [J]. Computer Engineering and Design, 2024, 45(07): 2195-2202. DOI: 10.16208/j.issn1000-7024.2024.07.037.

[11] Rawat, Arpit, Amit Gupta, and Richa Gupta. "Chord Protocol: Distributed Hash Table for Peer-to-Peer Networking." 2024 7th International Conference on Contemporary Computing and Informatics (IC3I). Vol. 7. IEEE, 2024.