Hindawi
IET Computers & Digital Techniques
Volume 2024, Article ID 2280828, 14 pages
https://doi.org/10.1049/2024/2280828

Research Article

Adaptive Shrink and Shard Architecture Design for Blockchain Storage Efficiency

Daniel Soesanto,[1,2] **Igi Ardiyanto**,[2] **and Teguh Bharata Adji**[2]

[1]*Department of Informatics Engineering, University of Surabaya, Surabaya, Indonesia*
[2]*Department of Electrical and Information Engineering, Gadjah Mada University, Jogjakarta, Indonesia*

Correspondence should be addressed to Teguh Bharata Adji; adji@ugm.ac.id

Received 9 July 2023; Revised 29 January 2024; Accepted 30 January 2024; Published 21 February 2024

Academic Editor: Sudarshan K. Srinivasan

One of the problems in the blockchain is the formation of increasingly large data (big data) because each block must store all the transactions it makes. With the problem of the appearance of extensive data (big data), many studies aim to maintain the data in small amounts. This research combines a sorting data technique and a proper compression technique to obtain efficient data storage on the blockchain. The result of this research is a blockchain platform called Adaptive Shrink and Shard Blockchain (AS^{2}BC), which conceptually and computationally can minimize the use of storage space in the blockchain up to 22 times smaller.

1. Introduction

Big data is essential in building any intelligent system [1–6]. Intelligent system researchers need extensive ground truth data to make predictions with high accuracy [7–16]. In addition, organizations that store sensitive data from their users also require large amounts of storage. Storage of this sensitive data requires not only easily accessible media but also reliability in terms of security. Bitcoin, which Satoshi [17] introduced through his white paper in 2007, has simultaneously led to a new data storage alternative using the blockchain. Blockchain is a distributed storage medium that eliminates the role of third parties in every process. The blockchain mechanism reduces dependence on outsiders and increases the security of data entering this blockchain network. Blockchain is currently used in various systems, ranging from cloud computing-based systems and IoT to cryptocurrency systems [18–20]. In several European countries, health technology and livestock are already using blockchain [21, 22]. The purpose of using this blockchain system is to increase the security of existing data access and guarantee the reliability of stored data [23]. For example, blockchain keeps patients' medical history in the health sector so that this sensitive data can be guaranteed security [24]. In the livestock sector, however, in several studies, this blockchain is claimed to reduce the risk of food fraud [25–28]. Blockchain reduces the risk of food fraud in the food supply chain by implementing this blockchain network, from raw materials to finished goods such as frozen meat. This method makes the data entered from upstream (breeders) to downstream (end consumers or end traders) irreversible. For example, at the beginning of entering meat with grade C, then until the end, this meat will still be recorded as grade C.

Data storage mechanisms on a distributed blockchain cause each node to duplicate all the data in the network [29]. This distributed storage causes problems using devices with limited capabilities [30], not only in terms of the capacity needed to store but also in terms of the speed of access when retrieving the data. Devices with limited abilities will quickly fill their storage capacity, whereas the access speed will decrease linearly if the capacity is full. This problem will have an even more impact if it occurs in cases of utilizing big data on the blockchain network [31, 32]. It is also impossible to place conditions on the use of devices with high speed and capability to become blockchain nodes because it will reduce the scalability of the blockchain system itself.

Several researchers have been looking for ways to overcome this storage problem on the blockchain. Memory-Optimized and Flexible-Blockchain (MOF-BC), Proof of Property, and Block Summarization are among them. Each of these methods optimizes storage by deleting or ignoring no longer needed data

[33–35]. So, in these methods, an assessment is made of each data that will enter the blockchain network.

Between 2018 and 2021, several studies developed distributed storage methods but avoided data duplication [36–46]. These methods are Incentive Mechanism (2018), Forkbase (2018), Network Coded–Distributed Storage (NC–DS) (2018), Segment Blockchain (2020), Inter Planetary File System (IPFS) (2019), Sia Coin-Storj-FileCoin-Programmable Decentralized Storage and Delivery Network (PPIO) (2021), Light Chain (2019), Superlight (2019). Each method has similarities: distributing data to be stored on the blockchain network so that the storage load is not only on one node.

The BC Big Data Management System method, researched by Chen et al. [32], utilizes external storage outside the blockchain network, reducing blockchain capacity more efficiently. The blockchain network is only used to store essential data or keep pointers that will be used to retrieve data from external storage.

The last technique found in previous research is the blockchain network user clustering technique. In this technique, each user in the blockchain network will be separated according to their individual needs and interests [47, 48]. This is so that each person does not need to duplicate all the data on the blockchain network but only needs to replicate and own data from the cluster or group where the user is.

However, from every research done before, only one, Block Summarization, uses the compression method to optimize its storage. Even though the compression method is a classic method, it is still very efficient in saving storage capacity. Besides that, the distributed storage and cluster methods are also suitable methods to implement because they only need to store the data that needs to be stored. Therefore, this research contribution will combine methods that select essential data to be stored, namely methods of deleting or ignoring unimportant data, distributed storage, and clusters. Then, the stored data will also be compressed, resulting in genuinely efficient data in terms of storage capacity. Therefore, in a nutshell, this research was conducted to make it easier for nodes with low storage capacity to join a blockchain network so that the scalability of the blockchain can increase. To the best of our knowledge, there has never been any research related to increasing storage efficiency on blockchain that combines erasure, distribution, grouping, and data compression factors into one new method. This research was conducted from the analysis stage to designing a blockchain platform capable of optimization and data storage efficiency.

2. Related Work and Motivations

The increase in scalability can not only be seen from the rise in consensus speed but also needs to be considered for handling ample data storage on the blockchain. The concept of distributed data storage on the blockchain is beneficial for maintaining data integrity [49] as long as its size does not always increase.

2.1. Distributed Data Storage Research. MOF-BC is one of the studies that seek to reduce the data that must be stored on the blockchain to avoid a massive increase in node sizes [33]. In this method, three parties are generally involved: the user, the service provider, and the blockchain network. These three parties have the right to determine which data must still be stored or which can be deleted.

In another study, the concept of distributed storage was optimized, but trying not to repeat data already stored in a node. So, there are nodes that have extensive data, and there are also small ones. So that the user who owns the node does not mind storing other people's data, the system is given incentives that are adjusted to the amount of data stored by the node [36].

In 2018, Wang et al. [37] introduced the concept of data storage on the blockchain with the Forkabse principle. This method is similar to a version control system in that the system will only store data that changes so that each node does not need to have precisely the same data as the other nodes.

Other studies have taken things further by making each node store data differently from every other node [38]. This is done using NC-DS. However, this method has weaknesses in security if the blockchain is used as a means of financial transactions. Because this method cannot validate data with other nodes, but this drawback can be overcome if you implement blockchain segments. Data duplication from different nodes is still being carried out in the blockchain segment, but only 50% of all nodes [39] because data verification on the blockchain can be valid if at least 51% of nodes also have the same data.

2.2. Grouping Data Storage Research. Data grouping and separation for storage efficiency on the blockchain was also carried out by Zheng et al. [40] in 2019. The concept introduced by Zheng et al. [40] uses a separate place to store information on the blockchain, while user nodes only need to keep the hash, which will be related to the actual data. Because the data is not stored in each node, and the existing nodes only store the hash, which is the address of the data, the storage size at each node can be reduced significantly. Apart from Zheng, the research conducted by Changir et al. [41] also has a similar concept. Because at Zheng, each node in the blockchain network only needs to store the metadata of every existing actual data. This metadata will then be used to relate to the actual data [41], where the actual data is outside the blockchain network itself.

Light Chain also implements an almost similar concept, and it is just that the difference in Light Chain, which stores data, is that each node has made the transaction itself [42]. So if node A has completed five payment transactions, then this history will be stored at node A but will not be broadcast to other nodes. Later, if other nodes will make transactions with node A and need to know the transaction history of the previous node A, then the other node will see the information directly on node A. The pointer to know this information will be a hash as the address of each data on different nodes. A concept similar to this is Superlight, where in this concept, each node also does not need to duplicate all data in a block but only saves the block header [43]. The storage of this header block is also verified later using the Boneh–Lynn–Shacham Signature (BLS Signature), so the data becomes more secure but still compact. But this is also a drawback of this Superlight because the computation time required by the BLS Signature is quite long.

In 2021, Li et al. [47] introduced the Intra Cluster Integrity Strategy (ICIStrategy), which groups blockchain users into prebuilt clusters. Users in each cluster only need to duplicate data for each node in the same cluster as that user. This strategy reduces the storage capacity each node has to provide because there is no need to store all data from each node on the blockchain network. This strategy is also in line with Rapid Chain, which was introduced by Zamani et al. [48] earlier in 2018 and referred to grouping nodes as a sharding technique.

In 2019, Sanka and Cheung introduced the concept of separating data storage on the blockchain by selecting core and non-core data [32]. The essential or core data will be stored directly on the blockchain network. At the same time, data considered unimportant will be stored in a centralized database. However, this non-core data will still be hashed, and only the hash will be held on the blockchain network to maintain the integrity of this non-core data.

The sorting of data storage carried out by Ehmke et al. [34] introduced the concept of Proof of Property in their research. In this concept, each user does not need to know the entire historical data of each node but only needs to understand what properties each user has and does not need to know the history or origin of these properties [34]. This concept is similar to MOF-BC, where unimportant information does not need to be stored.

2.3. Compression Data Storage Research. In addition to reducing data storage by distributed data storage and grouping users into certain clusters, there is also research that develops a compression technique called Block Summarization. This compression technique is claimed to compress data on the blockchain up to a ratio of 0.54 of the original data size. In addition, the validation of each transaction on the designed blockchain network can be carried out independently without having to check the similarity of data on other nodes in the blockchain network [35].

3. Materials and Methods

The need for ample data storage on the blockchain needs to be responded to with suitable storage methods so that the scalability of this blockchain can still be accepted and implemented in the industry at large. However, in making storage arrangements, it should also be noted that big data requires a lot of data. This data is used for various analyses related to the needs of the blockchain, which ultimately impacts the processing speed of the transaction.

The data storage management method developed in this study is called Adaptive Shrink and Shard Blockchain Storage Management or AS^{2}BC. In this method, data will be sorted, and only a summary will be stored with a maximum duplication of 50% of the total number of nodes, and data that is no longer needed will also be deleted. Transaction verification is done by collecting all the required data from each node that stores data related to the transaction. This method refers to previous research regarding ways of shrinking or reducing data on the blockchain [32–34, 38, 41, 43]. In the AS^{2}BC method, the depreciation method will use block summarization, where in this research, the transaction data stored at each node is reduced [35]. This reduction has an impact on reducing the need for storage space needed for a transaction in a block. This ultimately also impacts reducing the burden on the nodes participating in the blockchain network. For nodes with devices with low specifications, they can store only essential changes from a block, known as block summarization storage. Meanwhile, nodes with high computational power and large storage capacity can still store complete blocks.

After the data is cleaned and only essential data is taken, the data will also be compressed to be smaller. Similar research has also been carried out on collecting data in the blockchain using the Block Summarization algorithm [35]. Meanwhile, in AS^{2}BC, data compression will use the SHA256 hash method. This is because this hash method will produce a hash length that is always the same no matter how long the data is entered. The results of this hash will be stored in each user's transaction data so that later, it can be seen that the data comes from that user. However, because SHA256 is a one-way encryption, which means it cannot be decrypted, it will still need a place to store the original data regarding this user's transactions. Meanwhile, to retrieve this initial data, the system will use the hash created to be a pointer to the data.

The original data storage for each user will use a distributed concept, or in this method, it is called a shard. Data storage techniques with storage deployments like this are also carried out by similar research on optimizing storage media on the blockchain [40, 42, 49]. This shard method will divide the existing data on each node in the blockchain network. So that users are interested in providing storage space on their nodes, an incentive mechanism is also given in the form of a certain nominal that will be paid by the data owner, as has been done in previous studies regarding incentives for distributed storage on the blockchain [36]. However, even though the data has been distributed, if it is still an extensive network, then over time, the data storage size at each node will also be fuller, even though that node does not necessarily need the data stored. Therefore, AS^{2}BC also implements the division of nodes into groups of nodes. This division is based on the average nominal transaction that has been made, so if users often make large numbers of transactions, they will be grouped with those with the same nominal transaction behavior. However, the division of users into groups of nodes is also not permanent, so it can always change if the average nominal transaction of these users changes. Grouping transactions within network nodes is also carried out by similar research on increasing storage space efficiency on blockchain networks [47, 48].

Figure 1 shows the architectural model of AS^{2}BC, where each node in the blockchain network will be divided and grouped according to the average amount of transactions stored in that node. If the amount is included in a large transaction, it will enter into a group of nodes called the Group of Big (G_{ob}), while vice versa, it will enter into the Group of Small (G_{os}). Later, each node only needs to duplicate data and consensus on transactions in a group of nodes. The determination of the average

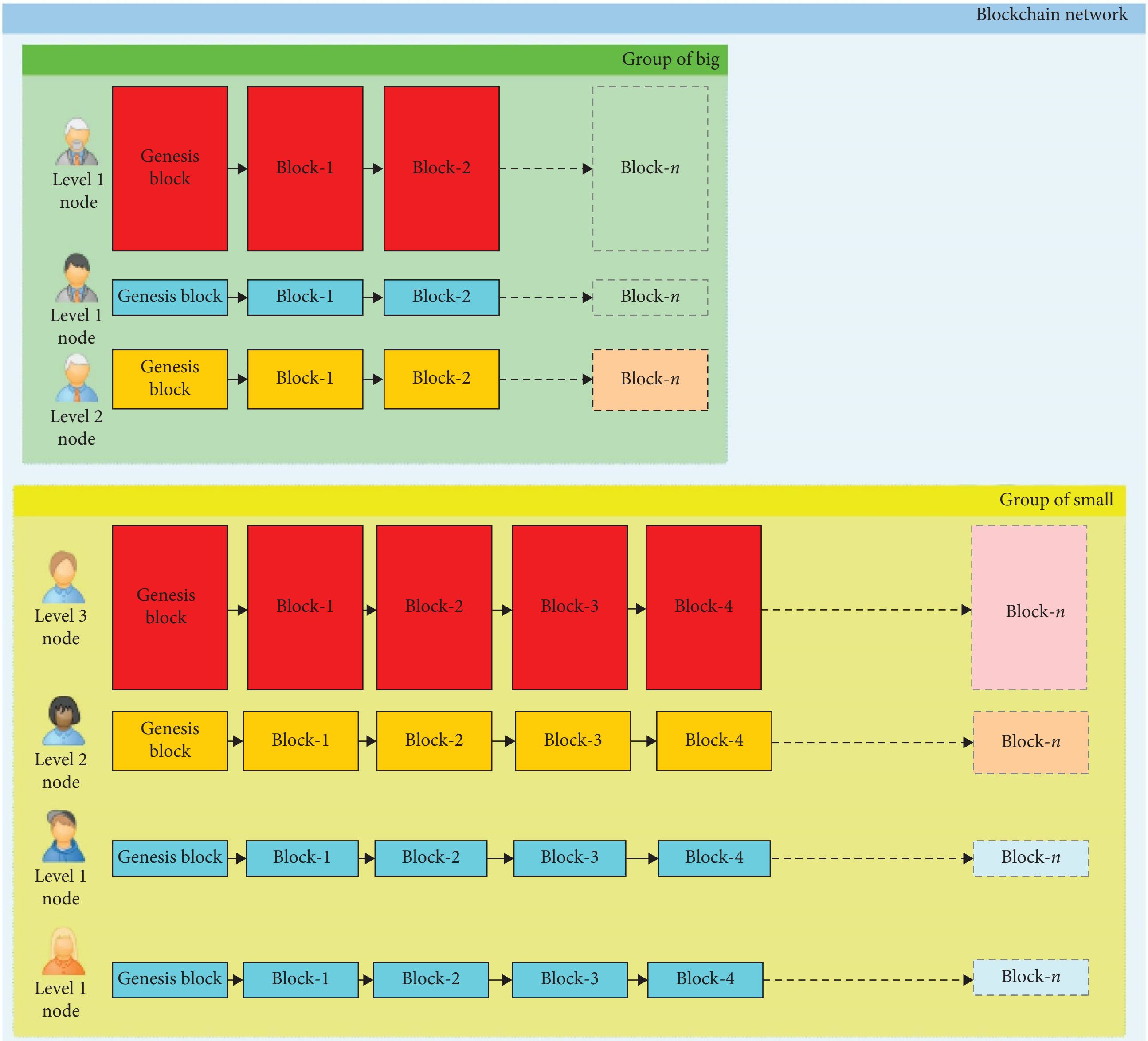


Figure 1: AS^{2}BC architecture.

number of transactions that enter into each group will follow the equation shown in Equations (1) and (2). Figure 1 also offers three levels of nodes, namely levels 1–3. Level 1 shows the node with the highest compression level, so it has the smallest size, and conversely, the higher the level, the lower the compression level, so the larger the storage size required by the node. An explanation of the differences in each node level will be further discussed in Section 4.

$$G_{\mathrm{ob}} = \left(\frac{\sum_1^{nu} \text{nominal } tx}{n} \right) > \tilde{x}, \tag{1}$$

$$G_{\mathrm{os}} = \left(\frac{\sum_1^{nu} \text{nominal } tx}{n} \right) \leq \tilde{x}. \tag{2}$$

In Equations (1) and (2), the calculation for determining the G_{ob} and G_{os} is shown, obtained from the total nominal transaction (tx) from the first transaction to the nth transaction at the node (nu) divided by the total number of transactions of the node (n), and compared with the median of all nominal transactions on the blockchain network ($\tilde{x}$). If it is larger, it will be put in the G_{ob}, and if it is smaller or equal to the median, it will be placed in the G_{os}.

It is also shown in Figure 1 that there is a clear separation between the G_{ob} and G_{os}. So that the nodes of each group simply handle the data in their group. In addition, in each group, one or more nodes provide their blocks as storage services for other users in the group (marked with red and yellow blocks). This storage service also exists as level 3, which means it provides storage for full nodes, and level 2 partially includes storage space for full nodes. Later, the node owner will get incentives from every data deposited in their storage service. The consequence of increasing the size of a block is the hash processing time of that block. The node that entrusts the data can access it through its hash, where the

Table 1: Comparison of methods.

Methods	Optimization consideration factors					
	1	2	3	4	5	6
MOF-BC [33]	☑	—	—	—	—	—
Incentive Mechanism [36]	—	☑	—	—	—	—
Forkbase [37]	—	☑	☑	—	—	—
NC-DS [38]	—	☑	—	—	—	—
Segment Blockchain [39]	—	☑	—	—	—	—
IPFS [40]	—	☑	—	☑	—	—
Sia Coin, Storj, FileCoin, PPIO [41, 44–46]	—	☑	—	☑	—	—
Light Chain [42]	—	☑	—	—	—	—
Superlight [43]	—	☑	—	—	—	—
ICIStrategy [47]	—	—	—	—	☑	—
RapidChain [48]	—	—	—	—	☑	—
BC Big Data Management System [32]	—	—	—	☑	—	—
Proof of Property [34]	☑	—	—	—	—	—
Block Summarization [35]	☑	—	—	—	—	☑
AS^{2}BC (proposed method)	☑	☑	—	—	☑	☑

hash is a pointer to the actual data in the storage service on other nodes. Node owners with adequate devices can offer themselves to the network to become storage service actors. The concept of blockchain as a service was also developed by Lu et al. [50]. However, in the study of Lu et al. [50], the service offered is not just storage between nodes but the use of the entire blockchain network for those who need it but have limited resources to develop their blockchain network.

Table 1 shows a comparison between the currently existing methods and the proposed method. As shown in Table 1, six factors are used to optimize data storage for each existing method, including the proposed method. These factors include (1) deleting data that is deemed unnecessary, (2) distributing storage in more than one block, (3) storing only data that changes like a version control system, (4) using storage media external outside the blockchain, (5) grouping (clustering) each user on the blockchain, and (6) compressing the incoming data so that it can have a smaller size. Table 1 also shows that the AS^{2}BC proposed method adapts several factors carried out by similar studies and combines them to get the best efficiency and reliability. As previously explained, it can be summarized that this proposed method applies a deletion process to data that is no longer needed, distributes stored data so that it does not need to pile up in one place, makes groupings of G_{ob}, and G_{os}, so that not all nodes must store all data on the network, as well as compress incoming data so that a smaller data size is obtained.

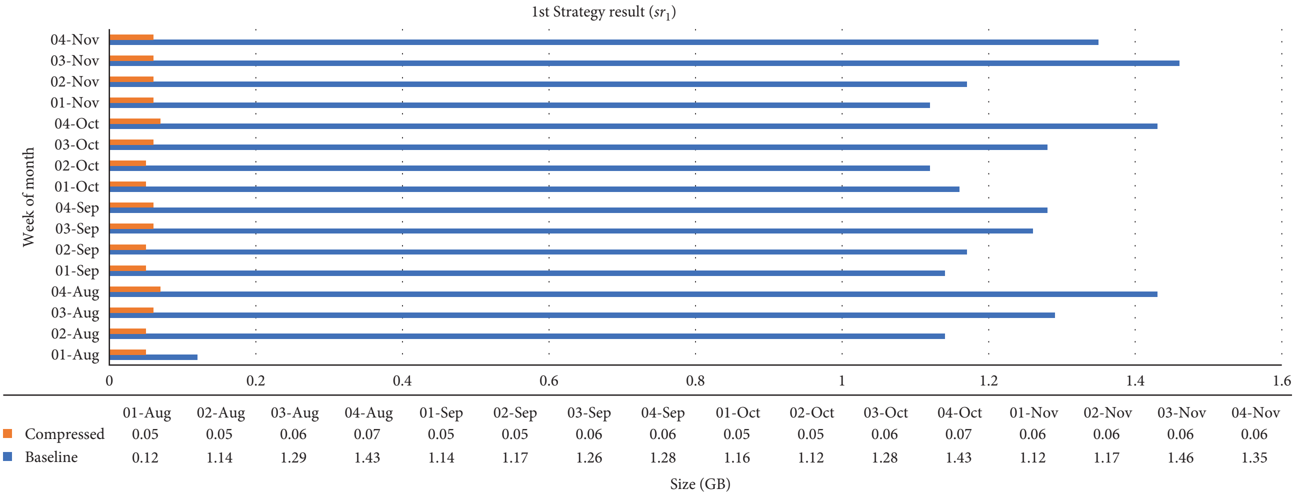
4. Result and Discussion

This section will describe the results of testing the four main strategies in the AS^{2}BC method. The cryptocurrency transaction data comes from Bitcoin from August 2022 to November 2022. The amount of transaction data taken during that period is 31,350,168 transactions, with a total size of 20,382 GB. The number and size of monthly transactions can be seen in Table 2, where the highest number was in October, with 7,973,133 transactions. Meanwhile, the largest total transaction size was in November, namely 5,231 GB.

Table 2: Transaction data recap.

Transaction month	Total
August	
Count	7,819,508
Size (GB)	5.078
September	
Count	7,683,590
Size (GB)	4.964
October	
Count	7,973,133
Size (GB)	5.109
November	
Count	7,873,937
Size (GB)	5.231

4.1. Data Compression. Any transaction-related data stored on the current blockchain network typically undergo no data compression or modification. However, due to the increasing number of transaction data and to speed up the consensus process, in one block processing, as many transactions as possible are included as long as the computing power of each node is still sufficient to carry out a complete consensus. But this also eventually causes the size of each block to be quite large. Especially now that the number of blocks is increasing, the total storage that must be provided by a node that wants to join the blockchain network will be even greater. This condition is a problem for the scalability of the blockchain itself because it also reduces the opportunity for new nodes to join. The data compression that will be carried out in this study uses the SHA256 algorithm.



	01-Aug	02-Aug	03-Aug	04-Aug	01-Sep	02-Sep	03-Sep	04-Sep	01-Oct	02-Oct	03-Oct	04-Oct	01-Nov	02-Nov	03-Nov	04-Nov
🟧 Compressed	0.05	0.05	0.06	0.07	0.05	0.05	0.06	0.06	0.05	0.05	0.06	0.07	0.06	0.06	0.06	0.06
🟦 Baseline	0.12	1.14	1.29	1.43	1.14	1.17	1.26	1.28	1.16	1.12	1.28	1.43	1.12	1.17	1.46	1.35

FIGURE 2: The 1st strategy result (compression).

However, because this encryption algorithm is only one-way, compression using this algorithm will only produce a hash, which is a pointer to the actual data. So, this data compression strategy will be related to the distribution strategy because the original data will be distributed to nodes with adequate capabilities. In this study, a simulation of compressed transaction data per block was carried out using the SHA256 algorithm, and the results were obtained per week, as shown in Figure 2, for data from August to November 2022.

The graph shows that the orange color results from compression, where the size becomes much smaller than the baseline. Not all nodes can store this compressed data, but there must still be nodes that store complete data with large sizes. Of course, nodes that store complete data have high computational capabilities and large storage capacities. This data compression calculation follows Equation (3). First, the number of transactions (txc) per day during the months from August to November 2022, multiplied by the number of bits generated from the SHA256 encryption algorithm. Then, because the result is still in bits, it must be divided by 2^{33} to become gigabytes.

$$sr_1 = \sum_{m=8}^{n} \frac{\text{sha}_{\text{bits}}\left(\sum_{d=1}^{ld} txc_d\right)}{2^{33}}. \tag{3}$$

4.2. Deleting Unimportant Data. In the second strategy, all nodes that store complete data in the first strategy will delete data that is considered unimportant. This strategy tries to implement the Block Summarization method that has been researched and developed by Palai et al. [35] in 2018. In this strategy, nodes that do not have high computing power and/or do not have a large storage capacity can store only essential changes from the data for each existing block, where this is referred to as Block Summarization. As for nodes with high computing power and storage capacity, they can keep all the blocks in the blockchain network intact. Implementation of Block Summarization can obtain a compression ratio of 54% [35]. So, with this level of compression, the data used in the baseline node in the AS^{2}BC research can shrink by 9.16 GB. The breakdown is a total of 2.68 GB in August, 2.62 GB in September, 2.69 GB in October, and 2.76 GB in November. The equation for calculating monthly depreciation can be seen in Equation (4), where the result of the first strategy (sr_1), which is total depreciation, can be calculated from depreciation per month from August to November 2022. The calculation starts from calculating the total transaction size per day from $d = 1$ to the last (ld) day, which is then converted to gigabytes by dividing by 2^{30}. The result is the total transaction size per day multiplied by the Block Summarization's compression ratio (c_r).

$$sr_2 = \sum_{m=8}^{n} \frac{c_r\left(\sum_{d=1}^{ld} \frac{txs_d}{2^{30}}\right)}{100}. \tag{4}$$

Suppose the depreciation is calculated per week from August to November. In that case, it can also be seen that the reduction in transactions will significantly impact the size of the transaction data per week. Therefore, as shown in Figure 3, the depreciation of transactions per week is shown where the y-axis is the size in GB, and the x-axis is each week from August to November.

4.3. Distribution Strategy. This third strategy is related to the two previous strategies. In the first strategy, there is a group of nodes, which can have the smallest data size because it will be compressed using SHA256, and then these nodes only need to store the hash of the results. At the same time, the original data will be stored on another node with a large storage capacity. If the smallest node is considered the default node and denoted as N, then the node where the complete data is stored will be referred to as N'. In the second strategy, N' is reoptimized so that there is a second tier to increase node scalability. So, the second strategy, N', is subject to the summarization method to make the data smaller by eliminating unnecessary data and only storing the necessary node changes. Therefore, from the second strategy results, N' is

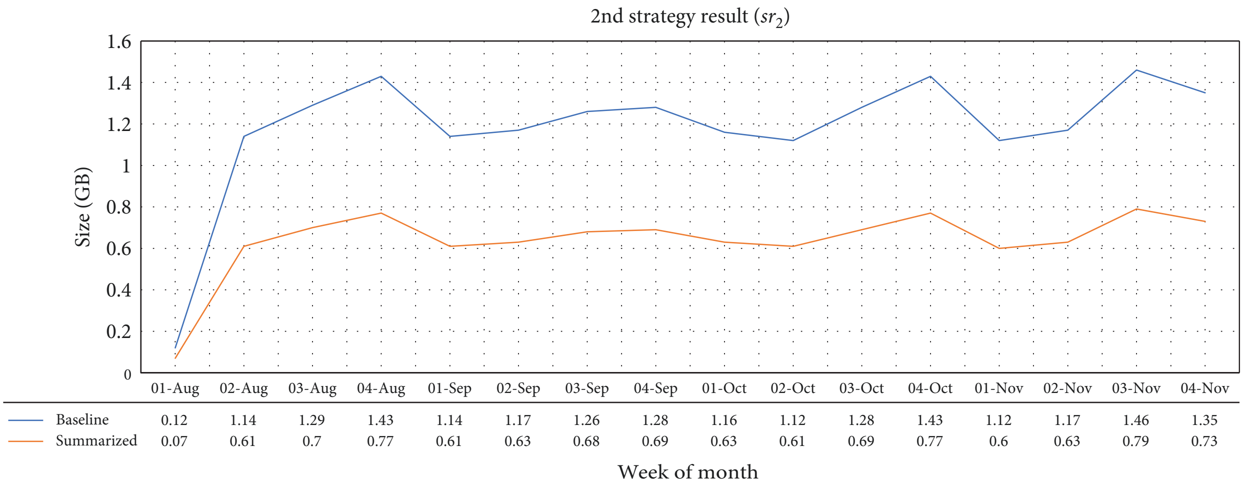


FIGURE 3: The 2nd strategy result (summarization).

TABLE 3: Comparison of distributed nodes (in GB).

Months	Level 1	Level 2	Level 3
August	0.233	2.742	5.078
September	0.229	2.681	4.964
October	0.238	2.759	5.109
November	0.235	2.825	5.231

no longer a node with the complete data but a level 2 node containing summarization results. The node with the complete data becomes a level 3 node denoted by N''. This mechanism is what is meant by the distribution strategy in AS^{2}BC. Table 3 shows a monthly comparison of node sizes for each level. Level 1 shows the most petite size compared to the others, so level 1 will have the most number of nodes in it. At this level 1, the nodes that join do not need high computing power or large storage capacities. However, the data stored is incomplete, so if verification is necessary, it will be connected to levels 2 and 3 through the hash pointer belonging to level 1. This hash pointer will exist at each level, and to associate, it will use a matching mechanism from the hash owned by each level.

4.4. Grouping. Grouping is the last strategy of the AS^{2}BC method, in which every transaction with a relatively nominal amount is grouped into one, which is then referred to as one group. There will only be two groups for all transaction data, G_{ob} and G_{os}. Each will store data groups with large transaction values and data groups with small transaction values. This grouping will follow Equations (1) and (2), as explained in the previous subsection.

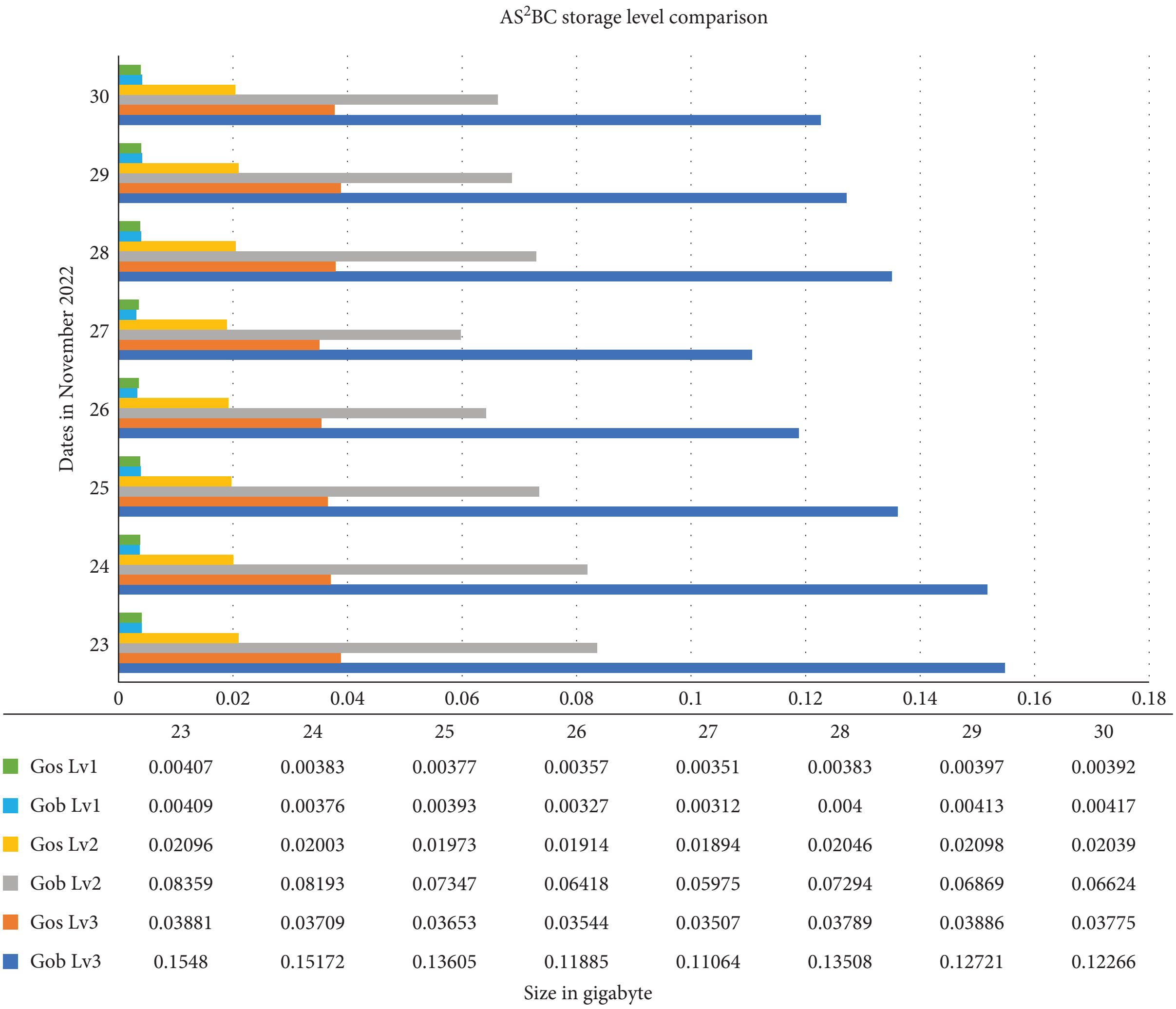
The impact of this strategy is being able to reduce the overall node size significantly. But at the beginning, when a node joins, it must decide whether to take a small or a big group. In this study, due to the limitations of data processing tools that still use Excel, where the maximum data row in Excel is 1,048,576, and the maximum column is 16,384 [51], the grouping calculation period is carried out in the last week of November 2022. Through this calculation, the median value is obtained. Transactions amounted to 444,408 USD, and the specifications for each group are shown in Table 4. In these results, it can be seen that the G_{ob} has the same number of transactions when dividing but has a larger size of 1.06 GB. Meanwhile, the G_{os} has a size of 0.30 GB. Nodes that will join the G_{ob}, of course, must have a larger storage capacity. However, this will not be a problem because the larger the transactions made (those that fall into the G_{ob}), the greater the included transaction fees. Therefore, the node owner will get a more significant profit from the G_{ob}, but must also have a more enormous capital because they have to provide a storage capacity of approximately 3.6 times that of the nodes that join the G_{os}. Calculating the storage capacity of the group of big (G_{obc}) can be formulated, as shown in Equation (5). The equation indicates that the measure begins by calculating the total size of all transactions that fall into the G_{ob} categories based on a predetermined median, then dividing it by the total size of all transactions in the G_{os}.

TABLE 4: Group definition simulation.

Name	Tx count	AS^{2}BC	Baseline
G_{ob}	1,022,474	1.06 GB	1.35 GB
G_{os}	1,022,474	0.30 GB	

$$G_{\text{bc}} = \frac{\sum_{d_{\text{Gb}}=1}^{ld_{\text{Gb}}} txs_{d_{\text{Gb}}}}{\sum_{d_{\text{Gs}}=1}^{ld_{\text{Gs}}} txs_{d_{\text{Gs}}}}. \tag{5}$$

4.5. Final Result. The final result of this research is to obtain an adaptive blockchain system architecture consisting of two large groups, G_{ob} and G_{os}. As explained in Section 3 in Figure 1, the division of this group is based on the median value of the nominal transactions that occur on the blockchain in a specific timeframe. After that, in each group, there will also be three levels of nodes that can be chosen by users who will join. Each node level has different storage needs,



	23	24	25	26	27	28	29	30
🟩 Gos Lv1	0.00407	0.00383	0.00377	0.00357	0.00351	0.00383	0.00397	0.00392
🟦 Gob Lv1	0.00409	0.00376	0.00393	0.00327	0.00312	0.004	0.00413	0.00417
🟨 Gos Lv2	0.02096	0.02003	0.01973	0.01914	0.01894	0.02046	0.02098	0.02039
⬜ Gob Lv2	0.08359	0.08193	0.07347	0.06418	0.05975	0.07294	0.06869	0.06624
🟧 Gos Lv3	0.03881	0.03709	0.03653	0.03544	0.03507	0.03789	0.03886	0.03775
🟦 Gob Lv3	0.1548	0.15172	0.13605	0.11885	0.11064	0.13508	0.12721	0.12266

Figure 4: AS^{2}BC final result.

where level 1 is the node with the lowest storage, level 2 is the node with medium-size storage, and level 3 is the node with complete data (baseline), so it requires the most significant storage. This study conducted a whole trial from group division to level distribution using a transaction dataset from November 23, 2022 to November 30, 2022. The results can be seen in Figure 4, where it can be seen that every day, the G_{ob} always has storage that tends to be large, not because of the large number of G_{ob} transactions, but because every transaction on G_{ob} also has a larger size. Meanwhile, at node level 1, which on the graph is colored green (G_{os} level 1) and light blue (G_{ob} level 1), the storage requirements of G_{os} and G_{ob} can be said to be relatively similar. This causes the scalability of the blockchain to increase because the newly joined nodes can more easily enter G_{os} and G_{ob} level 1.

The algorithm summary of the entire mechanism of this method can be seen in Algorithm 1 as follows: the input received is transaction data that will enter the block. At the same time, the output produced is data that will enter each block level. The node level will correlate with the block level that has been determined from this process, which means that level 1 blocks will enter level 1 nodes and so on. The process that occurs in Algorithm 1 can be explained as follows. Existing transaction data will be conditioned and divided into big or small groups. Then, the same processing will be done in each group, namely the compression stage, which will produce block level 1 data. The summarized stage will have block level 2 data; finally, the original data will be stored in block level 3.

G_{ob} level 1 and G_{os} level 1 is the most optimal storage efficiency in AS^{2}BC, so to see the efficiency level more clearly, this method is compared to similar methods in this study. The first method to be compared is ICIStrategy. This method determines how many nodes will participate in the blockchain network, which will be divided into several clusters. Each cluster will also be determined by the capacity of the nodes that can join [47]. In this test, ICIStrategy uses 2,000 participating nodes; each cluster can accommodate 100 nodes, so there are 20 clusters. The second method being compared is Block Summarization. This method deletes data that is no longer needed, resulting in a large enough data compression ratio [35]. The third method being compared is the BC Big Data Management System, which divides data on the

```
Input: n transaction data Tx
Output: $L_{\mathrm{B}}$ block data per level
for $i = 0$ to $n–1$ do
if    $(\frac{\sum_{1}^{nu} \text{nominal } tx}{n}) > \tilde{x}$
        $L_{\mathrm{Bb}}\ [1] =$    $\sum_{m=8}^{n} \frac{\mathrm{sha}_{\mathrm{bits}}(\sum_{d=1}^{ld} txc_d)}{2^{33}}$
        $L_{\mathrm{Bb}}\ [2] =$    $\sum_{m=8}^{n} \frac{{}^{c}{}_{r}(\sum_{d=1}^{ld} \frac{txs_d}{2^{30}})}{100}$
        $L_{\mathrm{Bb}}\ [3] =$    Tx[i]
else
        $L_{\mathrm{Bs}}\ [1] =$    $\sum_{m=8}^{n} \frac{\mathrm{sha}_{\mathrm{bits}}(\sum_{d=1}^{ld} txc_d)}{2^{33}}$
        $L_{\mathrm{Bs}}\ [2] =$    $\sum_{m=8}^{n} \frac{{}^{c}{}_{r}(\sum_{d=1}^{ld} \frac{txs_d}{2^{30}})}{100}$
        $L_{\mathrm{Bs}}\ [3] =$    Tx[i]
end if
return $L_{\mathrm{B}}$
```

Algorithm 1: Process on AS^{2}BC.

blockchain by distributing most of the data to a centralized database [32]. This method's impact is that the storage size required on the blockchain network is significantly reduced because most of it has been entered into a centralized database. The last method to compare is IPFS, which reduces hash redundancy by storing once for transactions with the same hash [40]. Through this IPFS method, data on the blockchain can be compressed significantly. Figure 5 shows a comparison between the AS^{2}BC method at the most optimal level (level 1) with the ICIStrategy, Block Summarization, BC Big Data Management System, and IPFS methods. Figure 5 shows that the AS^{2}BC method for both G_{ob} and G_{os} has outperformed the other four methods.

In Table 5, a summary of the tests between methods is made, where in the AS^{2}BC method, the greatest (Max) and smallest (Min) efficiency of each group (G_{ob} and G_{os}) is compared with the other four methods. It was found that the most outstanding efficiency was when G_{ob} level 1 was compared to Block Summarization, where the efficiency obtained was G_{ob} level 1 25.22 times smaller than Block Summarization. Meanwhile, the lowest efficiency was obtained when G_{ob} level 1 was compared to ICIStrategy, where the efficiency obtained, namely G_{ob} level 1, was only 1.63 times lower than ICIStrategy. However, overall, the AS^{2}BC method consists of G_{ob} and G_{os} still has an efficiency level above the other four compared methods.

At level 2 nodes, the resulting storage efficiency is quite reasonable compared to the Block Summarization method. However, because level 2 is not the optimal level for AS^{2}BC, a level 2 node size is larger than the results of other methods, as shown in Figure 6. Later, level 2 is intended for users who still have a larger storage capacity.

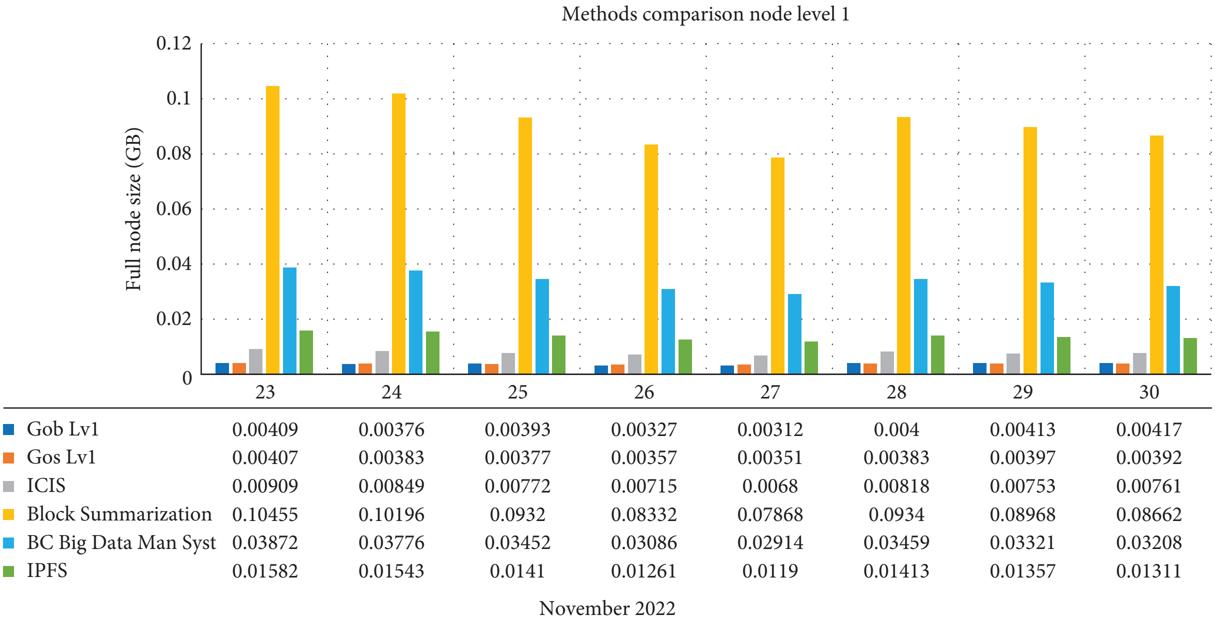
In Table 6, as in Table 5, a summary of the results of level 2 is also shown. At level 2, it can be seen that the most outstanding efficiency is obtained at G_{os} level 2 when compared with Block Summarization, namely that G_{os} level 2 is 4.15 times smaller than Block Summarization. Meanwhile, the most negligible efficiency was obtained when G_{ob} level 2 was compared with Block Summarization, where it was found that G_{ob} level 2 was 1.32 times smaller than Block Summarization.

At level 3 nodes, the efficiency of AS^{2}BC, when compared to other methods, is less visible. This is because level 3 nodes already store the most complete data. So, at level 3, AS^{2}BC storage efficiency is only visible at G_{os} level 3 compared to Block Summarization. Meanwhile, AS^{2}BC level 3 results with other methods tend to have more extensive data sizes, as seen in Figure 7.

In Table 7, as well as Tables 5 and 6, a summary of the comparison ratio is shown to see the efficiency of the AS^{2}BC method compared to other methods. The Table 7 shows that the most significant and negligible efficiency is only at G_{os} level 3 compared to Block Summarization, namely 2.24 times smaller and 2.02 times smaller, respectively. However, this is not a problem because level 3 nodes are given to users who want to provide data storage services. Meanwhile, ordinary users who require a high level of storage efficiency will be directed to become level 1 nodes with the most significant level of efficiency.

4.6. Security Implications. The distributed storage method is of great importance in the proposed AS^{2}BC method. In this method, data is not stored on the main node of the blockchain network. Meanwhile, the hash matching principle is used to retrieve data from each scattered storage location, as is the case when matching login passwords, where the hash technique used in this research is SHA256. In 2018 and 2022, several researchers conducted research proving that The SHA256 hash method is still a reasonably safe method for securing data stored in standard blockchain applications or intelligent systems [52, 53]. Apart from that, from 2014 to 2021, a group of people also researched the comparative analysis between several hash methods, where SHA is always compared. This research also revealed that the SHA256 method could still be the best choice in terms of security, among other hash methods. However, the SHA256 method has several weaknesses, such as larger storage capacity and slower speed when compared to other hash methods [54–57].

To the best of our knowledge, the only security threat in the SHA256 method is a collision attack, which has never happened. Still, in theory, it is possible [53, 56, 57]. Collision attacks are one of the security threats that exist in every hash technique. Collision attacks on SHA256 can occur in theory, one of which is using the brute force method [58]. In 2021, Hosoyamada and Sasaki [59] proved that collision attacks on SHA256 and SHA512 can occur using a Quantum Machine, although it is still not straightforward. However, with research that has led to the use of Quantum Machines, the security threat to hash methods such as SHA256 is increasingly accurate. Several of the world's leading vendors, such as Microsoft Azure Quantum, Google Quantum AI, and IBM Quantum, have started to develop and provide services for researchers to try to use Quantum Machines to carry out heavy computing processes.



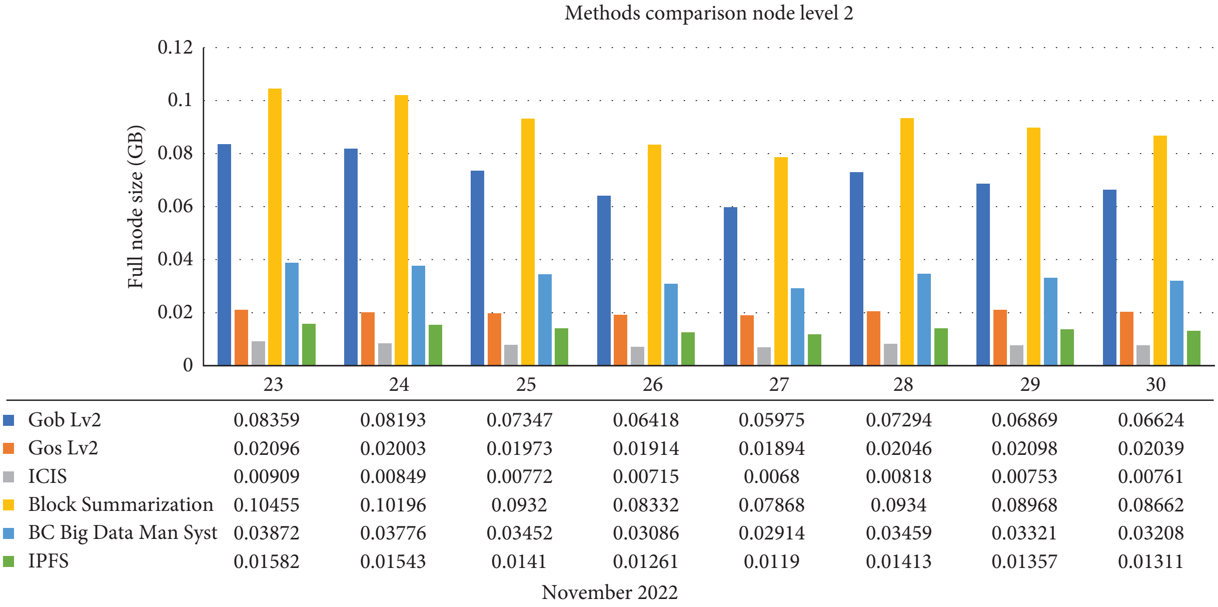
	23	24	25	26	27	28	29	30
🟦 Gob Lv1	0.00409	0.00376	0.00393	0.00327	0.00312	0.004	0.00413	0.00417
🟧 Gos Lv1	0.00407	0.00383	0.00377	0.00357	0.00351	0.00383	0.00397	0.00392
⬜ ICIS	0.00909	0.00849	0.00772	0.00715	0.0068	0.00818	0.00753	0.00761
🟨 Block Summarization	0.10455	0.10196	0.0932	0.08332	0.07868	0.0934	0.08968	0.08662
🟦 BC Big Data Man Syst	0.03872	0.03776	0.03452	0.03086	0.02914	0.03459	0.03321	0.03208
🟩 IPFS	0.01582	0.01543	0.0141	0.01261	0.0119	0.01413	0.01357	0.01311

Figure 5: Methods comparison node level 1.

Table 5: Benchmark result summary node level 1.

Compared methods	Max	Min
G_{ob} Lv 1 vs. ICIS	2.18	**1.63**
G_{os} Lv 1 vs. ICIS	1.94	1.67
G_{ob} Lv 1 vs. Block Summarization	*25.22*	18.87
G_{os} Lv 1 vs. Block Summarization	22.42	19.33
G_{ob} Lv 1 vs. BC Big Data	9.34	6.99
G_{os} Lv 1 vs. BC Big Data	8.3	7.16
G_{ob} Lv 1 vs. IPFS	3.81	2.85
G_{os} Lv 1 vs. IPFS	3.39	2.92

The numbers in italics are the highest level of efficiency achieved from the test results, while the numbers in bold are the lowest efficiency levels achieved from the test results.



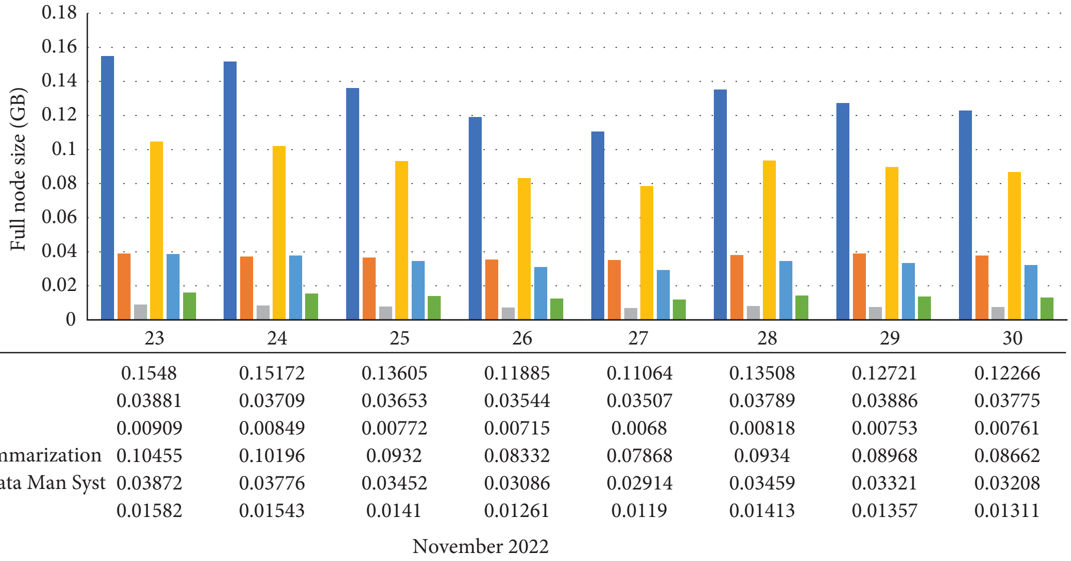
	23	24	25	26	27	28	29	30
🟦 Gob Lv2	0.08359	0.08193	0.07347	0.06418	0.05975	0.07294	0.06869	0.06624
🟧 Gos Lv2	0.02096	0.02003	0.01973	0.01914	0.01894	0.02046	0.02098	0.02039
⬜ ICIS	0.00909	0.00849	0.00772	0.00715	0.0068	0.00818	0.00753	0.00761
🟨 Block Summarization	0.10455	0.10196	0.0932	0.08332	0.07868	0.0934	0.08968	0.08662
🟦 BC Big Data Man Syst	0.03872	0.03776	0.03452	0.03086	0.02914	0.03459	0.03321	0.03208
🟩 IPFS	0.01582	0.01543	0.0141	0.01261	0.0119	0.01413	0.01357	0.01311

Figure 6: Methods comparison node level 2.

Table 6: Benchmark result summary node level 2.

Compared methods	Max	Min
G_{ob} Lv 2 vs. ICIS	0.11	0.08
G_{os} Lv 2 vs. ICIS	0.36	0.32
G_{ob} Lv 2 vs. Block Summarization	**1.32**	0.94
G_{os} Lv 2 vs. Block Summarization	*4.15*	3.75
G_{ob} Lv 2 vs. BC Big Data	0.49	0.35
G_{os} Lv 2 vs. BC Big Data	1.54	1.39
G_{ob} Lv 2 vs. IPFS	0.2	0.14
G_{os} Lv 2 vs. IPFS	0.63	0.57

The numbers in italics are the highest level of efficiency achieved from the test results, while the numbers in bold are the lowest efficiency levels achieved from the test results.



	23	24	25	26	27	28	29	30
🟦 Gob Lv3	0.1548	0.15172	0.13605	0.11885	0.11064	0.13508	0.12721	0.12266
🟧 Gos Lv3	0.03881	0.03709	0.03653	0.03544	0.03507	0.03789	0.03886	0.03775
⬜ ICIS	0.00909	0.00849	0.00772	0.00715	0.0068	0.00818	0.00753	0.00761
🟨 Block Summarization	0.10455	0.10196	0.0932	0.08332	0.07868	0.0934	0.08968	0.08662
🟦 BC Big Data Man Syst	0.03872	0.03776	0.03452	0.03086	0.02914	0.03459	0.03321	0.03208
🟩 IPFS	0.01582	0.01543	0.0141	0.01261	0.0119	0.01413	0.01357	0.01311

Figure 7: Methods comparison node level 3.

Table 7: Benchmark result summary node level 3.

Compared methods	Max	Min
G_{ob} Lv 3 vs. ICIS	0.06	0.04
G_{os} Lv 3 vs. ICIS	0.19	0.17
G_{ob} Lv 3 vs. Block Summarization	0.71	0.51
G_{os} Lv 3 vs. Block Summarization	*2.24*	**2.02**
G_{ob} Lv 3 vs. BC Big Data	0.26	0.19
G_{os} Lv 3 vs. BC Big Data	0.83	0.75
G_{ob} Lv 3 vs. IPFS	0.11	0.08
G_{os} Lv 3 vs. IPFS	0.34	0.31

The numbers in italics are the highest level of efficiency achieved from the test results, while the numbers in bold are the lowest efficiency levels achieved from the test results.

The pointer used to retrieve data uses a password mechanism, which will be encrypted on the client side when sending the data, so the security level of this data will also be equivalent to the security level of the login process in general. Judging from the review conducted by Cangir et al. [41] in 2021 regarding possible attacks from four distributed blockchain storage mechanisms, it can be concluded that most attacks can occur from the side of miners who commit fraud, either to take data, fake identities, or disrupt the smooth processes in the blockchain network itself. These attacks can occur because there is data scattered in several places, and of course, the potential for this attack also exists in the AS^{2}BC mechanism proposed in this research. However, the potential for this attack should be smaller in AS^{2}BC because, in this method, distribution is not only carried out horizontally, which means the original data is spread across several nodes but it is also spread vertically. This vertical distribution is seen in Table 3, where the original data will be at level 3 nodes, while the existing data pointers will be nested, namely pointers from level 3 to level 2 and from level 2 to level 1.

This mechanism causes SHA256 hash authentication to be carried out and is also multilevel, so an attacker, of course, needs double the effort to be able to control the original data on the AS^{2}BC network. Apart from that, further research still needs to be carried out to improve the safety of this AS^{2}BC method.

5. Conclusion and Future Works

The proposed design of the AS^{2}BC method can computationally reduce the total storage capacity on the blockchain. Utilization of the concepts of deletion, distribution, grouping, and compression makes the proposed method conceptually superior to methods carried out in similar studies. The size obtained at the level 1 node can be up to 22 times smaller than the baseline when using transaction data from August 2022 to November 2022. However, because one of the variables in the AS^{2}BC method is the transaction nominal, this method will not provide optimal results if applied to a blockchain network with transaction nominals that are too homogeneous. Future research is expected to incorporate existing blockchain platforms, such as Ethereum, to become one of the test materials for this design. Tools such as Hyperledger can also be an option for testing the implementation of the concept carried by the AS^{2}BC method.

Data Availability

The data that support the findings of this study are openly available in Blockchair at https://gz.blockchair.com/bitcoin/transactions/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

A scholarship from the University of Surabaya funded the first author. The generous financial support from the University of Surabaya is gratefully acknowledged, as it has been instrumental in enabling and enriching our research endeavors.

References

[1] H. V. Jagadish, J. Gehrke, A. Labrinidis et al., “Big data and its technical challenges,” *Communications of the ACM*, vol. 57, no. 7, pp. 86–94, 2014.

[2] V. Snasel and J. Kacprzyk, *Big Data in Complex Systems*, Springer, 2015.

[3] R. L. Villars and C. W. Olofson, *WHITE PAPER Big Data: what It is and Why You Should Care Informatio N Everywhere, But WHERE’S The Knowledge?*, IDC Analyze the Future, 2011.

[4] M. A.-u.-d. Khan, M. F. Uddin, and N. Gupta, “Seven V’s of big data understanding big data to extract value,” in *Proceedings of the 2014 Zone 1 Conference of the American Society for Engineering Education*, pp. 1–5, IEEE, Bridgeport, CT, USA, April 2014.

[5] I. Anagnostopoulos, S. Zeadally, and E. Exposito, “Handling big data: research challenges and future directions,” *The Journal of Supercomputing*, vol. 72, pp. 1494–1516, 2016.

[6] H. Özköse, E. S. Arı, and C. Gencer, “Yesterday, today and tomorrow of big data,” *Procedia—Social and Behavioral Sciences*, vol. 195, pp. 1042–1050, 2015.

[7] S. Athey, “Beyond prediction: using big data for policy problems,” *Science*, vol. 355, no. 6324, pp. 483–485, 2017.

[8] G. de los Campos, A. I. Vazquez, S. Hsu, and L. Lello, “Complex-trait prediction in the era of big data,” *Trends in Genetics*, vol. 34, no. 10, pp. 746–754, 2018.

[9] R. Northcott, “Big data and prediction: four case studies,” *Studies in History and Philosophy of Science Part A*, vol. 81, pp. 96–104, 2020.

[10] F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi, and M. Dehmer, “An introductory review of deep learning for prediction models with big data,” *Frontiers in Artificial Intelligence*, vol. 3, Article ID 4, 2020.

[11] L. Yao and Z. Ge, “Big data quality prediction in the process industry: a distributed parallel modeling framework,” *Journal of Process Control*, vol. 68, pp. 1–13, 2018.

[12] S. Niu, S. Wang, J. Wang, J. Xia, and G. Yu, “Integrative ecology in the era of big data—from observation to prediction,” *Science China Earth Sciences*, vol. 63, pp. 1429–1442, 2020.

[13] M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang, “Disease prediction by machine learning over big data from healthcare communities,” *IEEE Access*, vol. 5, pp. 8869–8879, 2017.

[14] I. Ahmed, M. Ahmad, G. Jeon, and F. Piccialli, “A framework for pandemic prediction using big data analytics,” *Big Data Research*, vol. 25, Article ID 100190, 2021.

[15] P. C. Reddy and A. S. Babu, “Survey on weather prediction using big data analystics,” in *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pp. 1–6, IEEE, Coimbatore, India, February 2017.

[16] G. Gui, F. Liu, J. Sun, J. Yang, Z. Zhou, and D. Zhao, “Flight delay prediction based on aviation big data and machine learning,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 140–150, 2020.

[17] N. Satoshi, “Bitcoin: a peer-to-peer electronic cash system,” *Journal for General Philosophy of Science*, vol. 39, no. 1, pp. 53–67, 2008.

[18] K. Christidis and M. Devetsikiotis, “Blockchains and smart contracts for the internet of things,” *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[19] H. Yang, J. Yuan, H. Yao, Q. Yao, A. Yu, and J. Zhang, “Blockchain-based hierarchical trust networking for JointCloud,” *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1667–1677, 2020.

[20] D. Haryadi, Harisno, V. H. Kusumawardhana, and H. L. H. S. Warnars, “The implementation of e-money in mobile phone: a case study at PT bank KEB,” in *2018 Indonesian Association for Pattern Recognition International Conference (INAPR)*, pp. 202–206, IEEE, Jakarta, Indonesia, September 2018.

[21] I. Radanović and R. Likić, “Opportunities for use of blockchain technology in medicine,” *Applied Health Economics and Health Policy*, vol. 16, pp. 583–590, 2018.

[22] L. Yang, X.-Y. Liu, and J. S. Kim, “Cloud-based livestock monitoring system using RFID and blockchain technology,” in *2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, pp. 240–245, IEEE, New York, NY, USA, August 2020.

[23] M. Das, X. Tao, and J. C. P. Cheng, “BIM security: a critical review and recommendations using encryption strategy and blockchain,” *Automation in Construction*, vol. 126, Article ID 103682, 2021.

[24] P. Mukherjee, L. B. Barik, C. Pradhan, S. S. Patra, and R. K. Barik, "hQChain: leveraging towards blockchain and queueing model for secure smart connected health," *International Journal of E-Health and Medical Communications*, vol. 12, no. 6, pp. 1–20, 2021.

[25] S. K. Lo, X. Xu, C. Wang et al., "Digital-physical parity for food fraud detection," in *International Conference on Blockchain, ICBC 2019: Blockchain – ICBC 2019*, vol. 11521, pp. 65–79, Springer, Cham, 2019.

[26] M. Tripoli and J. Schmidhuber, "Optimising traceability in trade for live animals and animal products with digital technologies," *Revue Scientifique et Technique de l'OIE*, vol. 39, no. 1, pp. 235–244, 2020.

[27] M. Guo, X. J. Liu, and W. Zhang, "Using blockchain technology in human food chain provenance," *WIT Transactions on The Built Environment*, vol. 179, pp. 391–396, 2018.

[28] K. S. Loke and O. C. Ann, "Food traceability and prevention of location fraud using blockchain," in *2020 IEEE 8th R10 Humanitarian Technology Conference (R10-HTC)*, pp. 1–5, IEEE, Kuching, Malaysia, December 2020.

[29] F. Casino, T. K. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications: current status, classification and open issues," *Telematics and Informatics*, vol. 36, pp. 55–81, 2019.

[30] I. Makhdoom, M. Abolhasan, H. Abbas, and W. Ni, "Blockchain's adoption in IoT: the challenges, and a way forward," *Journal of Network and Computer Applications*, vol. 125, pp. 251–279, 2019.

[31] N. Deepa, Q.-V. Pham, D. C. Nguyen et al., "A survey on blockchain for big data: approaches, opportunities, and future directions," *Future Generation Computer Systems*, vol. 131, pp. 209–226, 2022.

[32] J. Chen, Z. Lv, and H. Song, "Design of personnel big data management system based on blockchain," *Future Generation Computer Systems*, vol. 101, pp. 1122–1129, 2019.

[33] A. Dorri, S. S. Kanhere, and R. Jurdak, "MOF-BC: a memory optimized and flexible blockchain for large scale networks," *Future Generation Computer Systems*, vol. 92, pp. 357–373, 2019.

[34] C. Ehmke, F. Wessling, and C. M. Friedrich, "Proof-of-property: a lightweight and scalable blockchain protocol," in *WETSEB '18: Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain*, pp. 48–51, Association for Computing Machinery, Gothenburg, Sweden, May 2018.

[35] A. Palai, M. Vora, and A. Shah, "Empowering light nodes in blockchains with block summarization," in *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5, IEEE, Paris, France, February 2018.

[36] Y. Ren, Y. Liu, S. Ji, A. K. Sangaiah, and J. Wang, "Incentive mechanism of data storage based on blockchain for wireless sensor networks," *Mobile Information Systems*, vol. 2018, Article ID 6874158, 10 pages, 2018.

[37] S. Wang, T. T. A. Dinh, Q. Lin et al., "ForkBase: an efficient storage engine for blockchain and forkable applications," *Proceedings of the VLDB Endowment*, vol. 11, no. 10, pp. 1137–1150, 2018.

[38] M. Dai, S. Zhang, H. Wang, and S. Jin, "A low storage room requirement framework for distributed ledger in blockchain," *IEEE Access*, vol. 6, pp. 22970–22975, 2018.

[39] Y. Xu and Y. Huang, "Segment blockchain: a size reduced storage mechanism for blockchain," *IEEE Access*, vol. 8, pp. 17434–17441, 2020.

[40] Q. Zheng, Y. Li, P. Chen, and X. Dong, "An innovative IPFS-based storage model for blockchain," in *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pp. 704–708, IEEE, Santiago, Chile, December 2018.

[41] O. F. Cangir, O. Cankur, and A. Ozsoy, "A taxonomy for blockchain based distributed storage technologies," *Information Processing & Management*, vol. 58, no. 5, Article ID 102627, 2021.

[42] Y. Hassanzadeh-Nazarabadi, A. Küpçü, and Ö. Özkasap, "LightChain: a DHT-based blockchain for resource constrained environments," 2019.

[43] R. Blum and T. Bocek, "Superlight – a permissionless, light-client only blockchain with self-contained proofs and BLS signatures," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 36–41, IEEE, Arlington, VA, USA, April 2019.

[44] C. H. G. von Heyden, "Sia: simple decentralized storage David," *David Vor*, vol. 16, pp. 368–370, 2014.

[45] S. Wilkinson, T. Boshevski, J. Brandoff et al., *Storj A Peer-to-Peer Cloud Storage Network*, pp. 1–37, Storj Labs Inc, Atlanta, GA, USA, 2016.

[46] J. Benet and N. Greco, "Filecoin: a decentralized storage network," *Protocol Labs*, pp. 1–36, 2017.

[47] M. Li, Y. Qin, B. Liu, and X. Chu, "Enhancing the efficiency and scalability of blockchain through probabilistic verification and clustering," *Information Processing & Management*, vol. 58, no. 5, Article ID 102650, 2021.

[48] M. Zamani, M. Movahedi, and M. Raykova, "RapidChain: scaling blockchain via full sharding," in *CCS '18: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 931–948, Association for Computing Machinery, Toronto, Canada, October 2018.

[49] R. Li, T. Song, B. Mei, H. Li, X. Cheng, and L. Sun, "Blockchain for large-scale internet of things data storage and protection," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 762–771, 2019.

[50] Y. Liu, Q. Lu, H.-Y. Paik, X. Xu, S. Chen, and L. Zhu, "Design pattern as a service for blockchain-based self-sovereign identity," *IEEE Software*, vol. 37, no. 5, pp. 30–36, 2020.

[51] Microsoft 365 Support, "Excel specifications and limits," 2022, Microsoft. Accessed: Dec. 26. [Online]. Available: https://support.microsoft.com/en-us/office/excel-specifications-and-limits-1672b34d-7043-467e-8e27-269d656771c3.

[52] N. Joshi Padma, N. Ravishankar, M. B. Raju, and N. C. Ravi, "Building security barriers by modified algorithms in blockchain to prevent Sql injection and Xss," *Indian Journal of Computer Science and Engineering*, vol. 13, no. 2, pp. 477–488, 2022.

[53] S. Yaji, K. Bangera, and B. Neelima, "Privacy preserving in blockchain based on partial homomorphic encryption system for ai applications," in *2018 IEEE 25th International Conference on High Performance Computing Workshops (HiPCW)*, pp. 81–85, IEEE, Bengaluru, India, December 2018.

[54] S. Kumar and E. P. Gupta, "A comparative analysis of SHA and MD5 algorithm," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 3, pp. 4492–4495, 2014.

[55] S. Long, "A comparative analysis of the application of hashing encryption algorithms for MD5, SHA-1, and SHA-512," *Journal of Physics: Conference Series*, vol. 1314, Article ID 012210, 2019.

[56] P. P. Pittalia, "A comparative study of hash algorithms in cryptography," *International Journal of Computer Science and Mobile Computing*, vol. 8, no. 6, pp. 147–152, 2019.

[57] M. Parmar and H. J. Kaur, "Comparative analysis of secured hash algorithms for blockchain technology and internet of things," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 3, 2021.

[58] B. P. Kosta and P. S. Naidu, "Design and implementation of a strong and secure lightweight cryptographic hash algorithm using elliptic curve concept: SSLHA-160," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 2, pp. 624–635, 2021.

[59] A. Hosoyamada and Y. Sasaki, "Quantum collision attacks on reduced SHA-256 and SHA-512," in *Advances in Cryptology – CRYPTO 2021. CRYPTO 2021. Lecture Notes in Computer Science*, pp. 616–646, Springer, Cham, 2021.