Mesopotamian journal of Cybersecurity
Vol.5,No.2, **pp**. 349–360
DOI: https://doi.org/10.58496/MJCS/2025/022; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Research Article

Blockchain-Based Metadata Management in Distributed File Systems

Huda A. Alameen[1,*], Furkan Rabee[1]

[1] *Department of Computer Science, Faculty of Computer Science and Mathematics, Najaf, Iraq.*

ARTICLEINFO

Article history

Received 02 Aug 2024
Revised 28 Nov 2024
Accepted 16 Dec 2024
Published 25 May 2025

Keywords

Blockchain

Distributed File System

NameNode

DataNode

Ethereum

ABSTRACT

Managing large-scale data in distributed environments is essential for developing the distributed file system (DFS) concept, which ensures reliable, scalable, and fault-tolerant data storage across multiple nodes. In a DFS, DataNodes divides large datasets into blocks, assigning replicas to enhance data redundancy. The NameNode is a central control unit that manages metadata that governs data storage and retrieval. However, the NameNode presents a potential single point of failure, creating challenges in ensuring metadata, integrity, trustworthiness, and overall system reliability in distributed environments. This study proposes a new method to address these challenges by designing and implementing a new distributed file system architecture using blockchain as a repository to store the metadata of the NameNode. The proposed system achieves several significant improvements by integrating the blockchain into the DFS architecture. The tamper-proof and immutable nature of blockchain ensures metadata integrity. The metadata recorded on the blockchain can be accessible and recoverable at any time because multiple replicas are maintained over the network, and it becomes resistant to unauthorized modifications, enhancing trust and data reliability. The proposed architecture simulated the DFS via Python and integrated it with Ganache as an Ethereum platform via the Web3 library. The results show that the proposed system achieves the best time to upload files in the DFS compared with the traditional Hadoop distributed file system; the metadata stored in the blockchain enhance the overall system performance by improving metadata trustworthiness, management, and data integrity. The performance metrics for the proposed system are memory utilization and the file execution time for files ranging from 1 MB to 100,000 MB. The results show that even as the file size increase and the number of executions increases, the system retains efficient memory utilization, requiring less RAM for larger files. Although the system's ability to handle large datasets is demonstrated by its scalability in memory usage, adjustments are required to counteract the longer processing times linked to larger files. This paper examines the trade-offs and limitations of integrating blockchain with DFS and issues with scalability, latency, and storage costs. Despite these obstacles, the proposed approach shows that blockchain offers a workable option for safe and dependable metadata management in that DFS. This makes room for additional research to increase productivity and reduce resource use.

1. INTRODUCTION

The distributed file system (DFS) is a primary approach for managing unstructured data. The exponential increase in information has significantly enhanced the relevance of DFS in recent years [1]. DFS is engineered to distribute files over multiple computer nodes, facilitating efficient storage and retrieval in a distributed framework. The primary objective of DFS is to partition input files into smaller segments referred to as blocks. These blocks are then stored on distributed nodes within a network, offering a scalable and fault-tolerant storage solution for extensive computing environments [1, 2]. Metadata are essential for ensuring the scalability, integrity, and performance of these systems. It delineates file permissions, architectures, and interrelations. However, traditional metadata management systems face challenges such as scalability, fault tolerance, and security, especially as data centers become more complex and distributed. Distributed file systems (DFSs) have emerged as the foundation of contemporary data processing applications and big data analytics. Notable implementations include the Hadoop Distributed File System and the Google File System (GFS) [2, 3]. The DFS is composed of multiple DataNodes that store data blocks and a singular NameNode that oversees the metadata for the entire file system, encompassing location, mapping, and data length [4]. A significant limitation of this architecture is that the NameNode serves as a singular point of failure. Should it fail, the entire system becomes inoperative [5]. Moreover, interfacing with HDFS via APIs is a prevalent method; however, it presents security vulnerabilities that may be exploited

*Corresponding author. Email: hudaa.alameen@uokufa.edu.iq

in multiple manners [6, 7]. The principal objective of this study is to discern the deficiencies in current metadata management techniques and to propose remedies to mitigate these issues. Metadata are often stored and managed centrally or semii-centrally in distributed file systems, making them prone to bottlenecks, security breaches, and single points of failure that can harm system performance. In addition, as metadata volume increases, particularly in environments with high network churn or splitting, maintaining consistent and synchronized metadata over multiple nodes becomes increasingly challenging. In such distributed contexts, the main issues are ensuring metadata consistency, availability, and security. While centralized or hierarchical metadata servers face performance constraints and a higher risk of failure, fully distributed systems brone with reversing data consistency and preventing unauthorized access [4]. To solve these challenges, this work suggests a novel metadata management system based on blockchain. Many of the problems with present systems have a possible answer in blockchain technology, with its distributed, consensus-driven, unchangeable character. The proposed approach aims to reduce the risks connected with central points of failure, guarantee tamper-proof metadata records, and provide a scalable approach for synchronizing metadata among distributed via blockchain. Our approach integrates blockchain functionality into existing distributed file systems, enhancing their capabilities without significantly impacting performance[32]. The primary objectives of this work are to design a blockchain-based metadata management framework, implement it as a functional prototype, and evaluate its security and effectiveness compared with traditional methods. This research is significant because it has the potential to transform how metadata are managed in distributed file systems. The implementation of blockchain can reduce susceptibility to intrusions and failures, improve security and reliability, and enhance the efficiency of file system operations. This work provides a roadmap for future advancements in distributed storage and data management while introducing new possibilities for incorporating blockchain technology into critical infrastructure[33]. The proposed system addresses these challenges by enhancing the security of HDFS through the storage of relevant metadata on the blockchain. Our approach ensures the DFS metadata become immutable, replicable, and up-to-date with even the slightest modifications. Consequently, our data center storage will be safer, more reliable, and consistency updated. Therefore, the main contributions of this paper are listed below:

- A distributed file system linked with blockchain is designed to improve metadata storage, thus improving traceability and security.
- The development of file systems using a blockchain-based solution for file metadata storage enhances the capacity to track changes and maintain a safe file action record.
- Implementing real-time monitoring of changes made to data blocks in the DFS DataNodes. This is achieved by tracking metadata updates and instantly reflecting changes in the blockchain through new transactions, introducing an innovative feature to the system.
- Tis provides a more efficient and safer architecture for managing large datasets in distributed environments.

The content of this paper is as follows: in Section 1, we present a general overview of distributed file systems and their drawbacks in the introduction. We p an overview of the relevant studies on the blockchain and distributed file system (DFS) in Section 2. An initial overview of the techniques used in the proposed system is given in Section 3. The suggested methodology is thoroughly explained in Section 4. The suggested system's implementation is covered in Section 5. The results and discussion of the suggested system are detailed in Section 6. The limitations of our suggested system are detailed in Section 7. Finally, our conclusions are presented in section 8.

2. RELATED WORK

Many researchers have investigated the importance of blockchain technology in many fields, especially in enhancing distributed file systems. Here, several of these studies were considered.

Mothukuri et al. [5] proposed an approach to enhance the security of HDFS (Hadoop distributed file system) via a blockchain approach called BlockHDFS. The authors suggest the use of the Hyperledger Fabric platform to leverage files' metadata to create a secure and traceable system. In the implementation, they added only minimal metadata to the blockchain. They also proposed future work, such as developing their BlockHDFS system to work in real time with the file system and track all data between NameNode and DataNodes of the HDFS in the secure ledger with multiple nodes. Zhang and Wang. [7] This paper presents the design and implementation of a secure medical big-data ecosystem on the Hadoop platform. The primary motivation for this work is the need to enhance the intelligence of medical systems and improve the security of medical big data. They designed a personal health system to allow patients to access their rehabilitation status and treatment anytime and anywhere. The system ensures the storage and analysis of distributed medical health data from different independent institutions, maintaining their independence. Additionally, this paper explores the use of blockchain, a distributed accounting technology, to increase the security and privacy of medical data sharing. The proposed system leverages the capabilities of the Hadoop big data platform to provide personalized health management services for stroke patients, facilitating efficient

patient management by medical staff. Badr et al. [8] proposed an approach to securely storing medical data. They use chaskey cryptography to ensure the confidentiality and integrity of medical data and blockchain technology to prepare decentralized, immutable, and scalable storage capabilities. The significant challenges addressed relate to integrity, privacy, and security in the healthcare area. Al-Zubaidie and Jebbar. [9] proposed a two-phase authentication activity, hashing, and smart contract to strengthen transaction verification and help prevent unauthorized access. Researchers have used micro segmentation to isolate these processes in financial systems to improve security by containing potential breaches. Honar Pajooh et al. [10] used blockchain technology to design and implement layer-based distributed data storage for a large-scale IoT system. The system uses the hyperledger fabric (HLF) platform to address challenges such as the need for centralized servers and third-party auditors. Instead, HLF peers are used for transaction verification and record auditing, eliminating the need for a centralized authority. The lightweight verification tags are stored on the HLF blockchain ledger. In contrast, the metadata are stored in an off-chain big data system to reduce communication overhead and enhance data integrity.

Abdul-Sada and Furkan Rabee. [11] proposed a decentralized application called SONR, which uses evolutionary algorithms that simulate a natural reserve to protect endangered species; using a genetic algorithm, researchers aim to select the best IDs from the blockchain population. To improve the selection and mutation processes of the GA, they plan to enhance its effectiveness in preserving rare species. The study also seeks to develop improved fitness metrics to evaluate the quality of the created IDs and overcome limitations in testing smart contract programs within the execution environment. Additionally, researchers prefer security and scalability, intending to improve the application to meet recognized standards in these areas. Tyagi et al. [12] introduced a blockchain technology architecture approach to manage and secure data collected from IOT devices. They provide a distributed data storage architecture, thus eliminating the need for a centralized network topology using blockchain features such as decentralization, immutability, distributive, improved security, transparency, instant traceability, and increased efficiency through automation. The results show that the proposed approach ensures a high level of performance and can be used as a massive, scalable data storage solution for IoT devices via blockchain technologies. Tahayur and Al-Zubaidie. [13] used blockchain technology, artificial intelligence, and digital signatures (Ed25519) to enhance data security in e-agriculture applications. They also propose a security system that merges Ed25519 signatures with consortium blockchain blocks to exclude data manipulation in Internet of Things (IoT)-based agricultural applications. They also used an artificial bee colony (ABC) algorithm to improve the security and randomness of signatures while maintaining performance efficiency through advanced deep learning (ADL) techniques. Various cyberattacks, such as block discarding, selfish mining, and inference attacks, are countered via the designed system. As a result, the reliability and security of the agricultural data were enhanced.

Nair et al. [14] presented an approach to migrate data from Amazon S3 (centralized cloud systems) to an interplanetary file system as a decentralized file system (DFS), and the critical challenges in data integrity, ownership, and authorization were addressed. In this work, the researcher aims to retain data in its original cloud-based permissions while migrating to a decentralized network by introducing a custom blockchain framework for maintaining access control policies during this transition; this is critical, especially for distributed environments where central authority over data is absent and decentralized trust is paramount. Gupta and Dwivedi. [15] introduced an approach to enhance the security and traceability of the Hadoop distributed file system (HDFS) by using a blockchain-based framework. They integrate HDFS with the hyperledger fabric platform to store metadata in a blockchain. They also use an encryption technique to ensure that data are transmitted securely by using NodeJS to achieve interaction between HDFS and the blockchain layer. The critical issues in HDFS are addressed, such as a lack of traceability and security in traditional file systems. This study aims to secure big data settings by enabling file-level traceability and documenting all file changes to prevent tampering. Table I shows the previous works in terms of the techniques used in each study, along with the disadvantages and limitations of these studies.

TABLE I. PREVIOUS-PROPOSED WORKS

Works and Years	Methods Used	Drawback
V. Mothukuri, S. S. Cheerla, R. M. Parizi, Q. Zhang, and K. K. R. Choo, 2021	The researcher uses Hyperledger Fabric and HDFS, and NodeJs are used to connect between them.	There are real-time issues due to using the NodeJs that transfer data from WebHDFS to the blockchain, which are executed periodically. This method is not optimal for real-time data changes. WebHDFS is an HTTP-based interface that allows users to interact with HDFS over the web. If proper access controls and authentication mechanisms are not implemented, unauthorized access to metadata can be achieved.
X. Zhang and Y. Wang, 2021	They use Hadoop for data storage and Map Reduce for data processing, and they also use a data synchronization module with blockchain to ensure ample data security and consistency.	This work depends on Hadoop to store and process big data, so it does not test its actual deployment in large-scale healthcare environments, particularly its scalability and performance under real-world loads.

A. M. Badr, L. C. Fourati, and S. Ayed, 2023	Utilizes a combination of Chaskey cryptography, blockchain technology, and data fragmentation techniques.	No rigorous testing is conducted to assess the system's resilience against various attacks and identify potential vulnerabilities, which is essential to ensure its reliability and security.
Al-Zubaidie, Mishall, Jebbar, Wid, 2024	Using microsegmentation, smart contract, and Two-Phase Commit Algorithm (2PC) to enhance security in the banking system	After blocks are created and hashed in blockchain, updating and manipulating data becomes difficult or impossible, so the main challenges in this paper relate to updating data in blockchain.
H. Honar Pajooh, M. A. Rashid, F. Alam, and S. Demidenko, 2021	The researcher uses Hyperledger Fabric (HLF) integrated with edge computing, Hadoop, off-chain Metadata Storage, and a lightweight mutual authentication platform to collectively provide a secure, scalable, and effective provenance system for large-scale IoT data management in a distributed environment for Big Data.	As the data grows, system performance goes downhill, affecting the system's scalability. In addition, there are constraints on resources like network usage and CPU while processing massive data or high transaction rates. When the transaction rate exceeds threshold rates, the latency increases significantly. Finally, studies are limited to small-scale prototypes, and there are constraints in the system's function in real-world deployments.
H. H. Abdul-Sada and Furkan Rabee, 2023	The paper uses the Ethereum blockchain platform with Solidity smart contracts and Remix IDE to implement the genetic algorithm in a decentralized application (DApp) environment.	The main issues in this work are the limited size of the Solidity smart contract, memory constraints, the lack of Sophisticated tools for code validation, and the gas cost.
A. K. Tyagi, S. Dananjayan, D. Agarwal, and H. F. T. Ahmed, 2023	The researcher uses blockchain technology integrated with IOT to improve security and authentication in various industries.	The main limitations of this work are the high computational cost, limited throughput, low scalability, and security issues.
D. H. Tahayur and M. Al-Zubaidie, 2024	The researcher uses blockchain Technology, Ed25519 Digital Signatures, the Artificial Bee Colony (ABC) Algorithm, and Advanced Deep Learning (ADL).	This approach has implementation complexity, potential scalability, and high resource consumption issues. Using blockchain in distributed environments also raises privacy concerns. It has yet to be fully tested in real-world applications.
Nair et al, 2022	The researcher uses the combination of blockchain and IPFS to provide a decentralized and secure system for data storage and access control.	The suggested approach concentrates on migrating static access control policies to a decentralized network; however, it does not mention how policies can be dynamically modified or revoked in real-time after the migration. This can make it more difficult to manage policies as the system changes over time.
M. K. Gupta and R. K. Dwivedi,2023	the researcher uses Node Js to integrate HDFS with Hyperledger fabric to improve storing files in DFS and use ES256 to encrypt these files.	The main constraint on this study is the low amount of real-world testing, which results in a need for more testing in large-scale, dispersed real-world situations. Total dependency on NodeJS to connect blockchain and HDFS leads to performance bottlenecks.

3. THEORETICAL BACKGROUND

3.1 Blockchain Technology

Blockchain is a decentralized and distributed digital ledger technology that securely records, stores, and verifies transactions across multiple computers or nodes in a network. It is designed to provide transparency, immutability, and trust in a trustless environment [16, 8]. The concept of blockchain was initially developed for the cryptocurrency Bitcoin and was first described in 2008 by Satoshi Nakamoto in a whitepaper. Today, blockchain technology is being explored and implemented across various industries, including finance, supply chain management, healthcare, and voting systems, to enable secure and transparent transactions and data management [2, 17]. As shown in Fig. 1, each block in a blockchain consists of the following components:

1- The data stored in that block (e.g., transaction details).
2- Hash of the block.
3- Hash of the previous block.
4- Timestamp.
5- Nonce.

The first block in the blockchain, known as the genesis block, has its previous hash set to zero [17, 18].

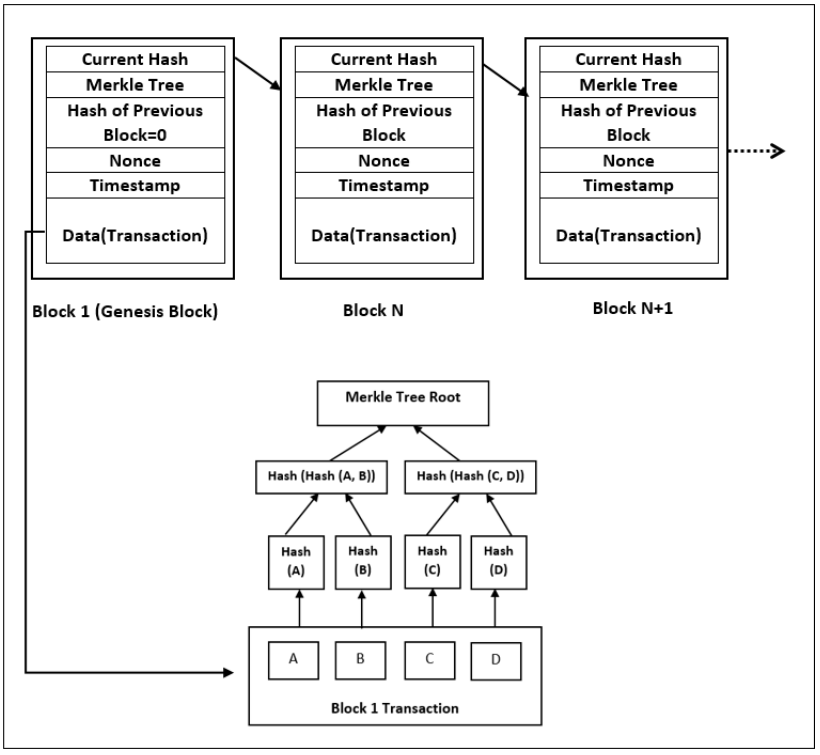

Fig. 1. Blockchain Structure

A block hash is calculated via a cryptographic SHA-256 hash function, which takes as input the block's data, including the nonce, timestamp, transactions, current hash, and previous hash [17]. The block hash serves as a unique identifier for the blocks and links it to the previous block in the blockchain [16, 19]. Any modification to the data within the block results in a completely different hash value, ensuring the integrity of the blockchain [19, 20].

3.2 Distributed File System

A distributed file system (DFS) allows users to access and manage data as if they were stored on a single, centralized file system, even though the file system is distributed across multiple networked computers [24, 1]. It provides a method for sharing, accessing, and storing files across a network of connected nodes [25]. Files in a DFS are divided into smaller sections called blocks, which are then distributed and replicated across multiple network nodes or storage devices [2]. Each node in the system contributes processing power and storage capacity, collectively operating the DFS. The main features and characteristics of a DFS include [26, 27]:

- Scalability: This is achieved by adding more nodes to the system, increasing its capacity to handle large amounts of data and distributing it across these nodes.
- Fault tolerance: Ensured by replicating the data across multiple nodes. If a node becomes unavailable or fails, data can still be accessed from other replicas, ensuring high availability.
- Data Consistency: Mechanisms are implemented to maintain consistency across replicas. Changes made to a file on one node are propagated to other replicas to ensure that all nodes have the most up-to-date version of the data.
- Unified Access: This provides a single interface for users to share and access files across the network. Regardless of the physical location of the data, users can read, write, and modify files as if they were stored on a local file system.
- Security: Includes mechanisms to protect data confidentiality and integrity. Encryption, access controls, and authentication measures ensure that unauthorized users cannot access or modify files.

The most popular examples of distributed file systems include the Hadoop Distributed File System (HDFS), the Ceph File System (CephFS), and the Google File System (GFS) [5, 28]. These systems have seen extensive use in numerous applications, such as big data analysis, cloud storage solutions, and environments for distributed computing.

The DFS architecture, as shown in Fig. 2, follows a master–slave architecture consisting of a NameNode, a Secondary NameNode, and a Data Node [26].

- NameNode: The NameNode serves as the controller node and is the sole NameNode in an HDFS cluster. It is responsible for storing metadata about files and directories. The NameNode tracks the details of each block in fragmented files and updates the metadata whenever operations such as file creation, renaming, or deletion occur [29]. DataNodes periodically send block reports, called heartbeats, to the NameNode as part of the system's health monitoring and communication mechanism. These heartbeats also notify the NameNode of any corrupted data blocks [1, 2].
- Secondary NameNode: The secondary NameNode is used to mitigate the single point of failure issue of the NameNode. It acts as a backup for the NameNode and assists in managing and maintaining the file system's metadata. The secondary NameNode performs periodic checkpoints to increase fault tolerance and improve recovery capabilities in the event of NameNode failure [2, 7].
- DataNodes: DataNodes provide the system’s storage and processing capacity. They are built using inexpensive commodity hardware and are deployed in clusters. Input files are divided into multiple blocks, which are stored in the DataNodes, with each block replicated at least three times for redundancy. Data nodes send regular heartbeat signals to the NameNode to maintain connections and verify their operational status [5, 26].

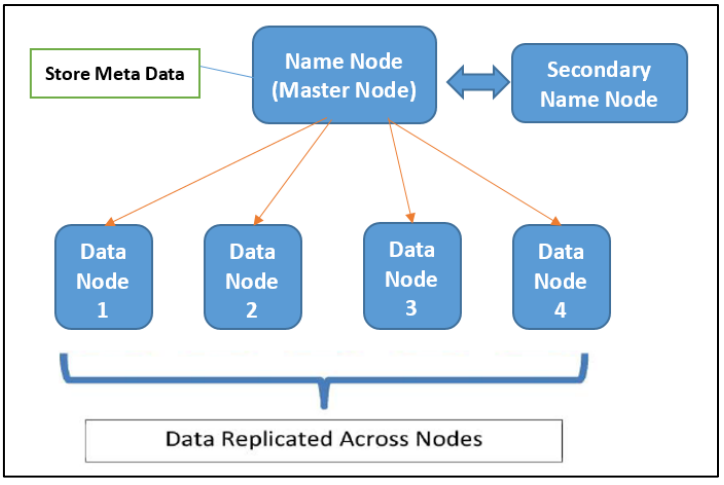

Fig. 2. DFS Structure.

3.3 Ethereum Technology

Ethereum is a decentralized platform that enables the execution of distributed applications and smart contracts without the risk of fraud, censorship, or third-party intervention. It is a blockchain-based platform with a Turing-complete programming language, allowing developers to create and publish distributed applications. Introduced by Vitalik Buterin, Ethereum is a public blockchain with no permissions required [20]. On Ethereum, transactions are executed via smart contracts, which implement predetermined rules in response to specific events [12]. These transactions are cryptographically signed instructions. Ether, Ethereum’s native cryptocurrency, is used to cover the cost of these transactions [22, 26]. In addition to serving as a means of payment, ether also acts as a pricing mechanism for decentralized applications [23]. The cost of computations on Ethereum is measured in terms of gas [20]. The Ethereum Virtual Machine (EVM), which runs on each network node, executes these commands. Potential applications on Ethereum include domains such as insurance, file storage, and market predictions [30,31].

4. PROPOSED METHODOLOGY

In traditional DFS, the NameNode represents a single point of failure, creating potential bottlenecks and a lack of fault tolerance owing to the absence of metadata replication. The failure of the NameNode can lead to system-wide failures, rendering the entire DFS inaccessible. This single point of failure poses a significant risk to the availability and reliability of

the system. The proposed system addresses this issue by storing the metadata of files from the NameNode on a blockchain. Blockchain is an exemplary solution for storing NameNode metadata because it is immutable, meaning that metadata are saved in numerous blocks and cannot be easily tampered with or edited. This ensures metadata integrity, consistency, and secure data management. First, we designed and implemented an overall distributed file system via Python.

- NameNode: Tracks file-to-block mapping information, the location of each block and its replicas, and the file system directories.
- Secondary NameNode: This node serves as a backup for the NameNode. If the NameNode fails, it can be recreated via this backup.
- DataNodes: Store the file blocks created by the user.
- Fault tolerance: If the NameNode or DataNode is deleted or blocked, the deleted component is recreated after the synchronization period.

When the DFS is loaded and a user creates or uploads a file via the command-line interface, the file is divided into blocks, and each block is replicated three times. Each replica is stored in a separate DataNode. The metadata, including the file name, the number of blocks, the DataNode where the blocks are stored, access times, and modification times, are then stored in the NameNode.

Second, the blockchain layer is configured to handle metadata operations such as file creation, deletion, and modification. Each metadata operation is recorded as a transaction on the blockchain. Ganache is used as the blockchain development tool and emulator, primarily for Ethereum-based blockchain development. Fig. 3 illustrates the architecture of the proposed system.

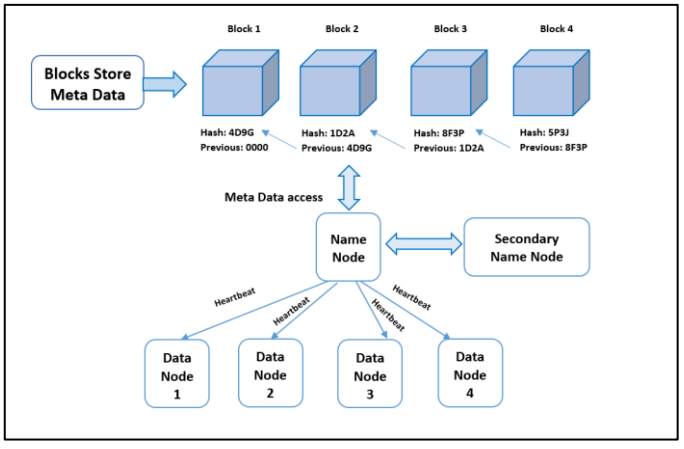

Fig. 3. Proposed System Structure.

Connecting the DFS with the blockchain via the Python Web3 library provides an interface for communicating with Ethereum, executing transactions, and reading or writing data on the Ethereum blockchain from a Python environment. The Web3 library leverages blockchain's decentralized structure to securely store metadata, ensuring that all entries are verifiable and tamper-proof. The key aspect of the proposed system is that any modifications to the metadata automatically create a new transaction on the blockchain. This guarantees that the metadata remain up-to-date and are safeguarded against manipulation or alteration. Furthermore, the system provides real-time capabilities by persistently monitoring metadata changes and reflecting them on the blockchain through the generation of new transactions.

5. IMPLEMENTATION

A new distributed file system (DFS) was designed and implemented via the Linux operating system and the Python programming language. The first step is to create and load the DFS on the basis of the provided configuration file. If the DFS does not exist, a new DFS is created with a NameNode, DataNodes, and secondary NameNode. Otherwise, the previously created DFS is loaded. A command-line interface is implemented via the argparse library, allowing users to execute commands such as put, ls, cat, rm, mkdir, rmdir, and MapReduce. For example, users can input a file via the following command:

python main.py --command put --arg1 file\path\filename --arg2/

where arg1 is the source path and arg2 is the destination path.

The NameNode and DataNodes are appropriately updated after executing these commands. The number of DataNodes, block size, and path to each DataNode, NameNode, and Secondary NameNode are specified. When a file is loaded, it is split into blocks and stored across different DataNodes. The metadata, such as the next DataNode, the number of blocks, and other information, are stored as JSON files in the NameNode. The blockchain is implemented via the Ganache platform, a local Ethereum blockchain environment designed for testing and development purposes. It securely stores all metadata as transactions, ensuring that the data are immutable and up-to-date. The Python Web3 library is used to connect the distributed file system (DFS) with Ganache. As shown in Fig. 5, each transaction is a dictionary object containing all the necessary information for execution. The main feature of the proposed system is the automatic creation of transactions whenever there are changes in metadata.

```
transaction = {
    'from': account_one,
    'to': account_2,
    'gas': gas ammount,
    'gasPrice': web3.to_wei('50',
'gwei'),
    'nonce':
web3.eth.get_transaction_count(accoun
t_1),
    'data': hex_content
}
```

Fig. 4. Python Code to Create a Transaction

6. RESULTS AND DISCUSSION

A new system has been designed and implemented to improve the traditional distributed file system (DFS), making it faster and more secure by storing the relevant metadata on the blockchain. The proposed system introduces a transparent, reliable, and low-cost approach to storing metadata in the blockchain. Since data stored on the blockchain are immutable, any changes made to it are permanently recorded in a manner that cannot be denied or repudiated. This feature ensures an exceptional chain of custody, maintaining a transparent and tamper-proof record of all transactions and modifications. This provides a convenient and trusted way to log file changes, making it easier for administrators to verify a file's authenticity during investigations. The proposed system runs on the Ubuntu operating system. The hardware specifications are an Intel(R) Core™ i7-5500U CPU running with a 2.40 GHz CPU, 8 GB of RAM, and a 256 GB SSD. Extensive experiments were conducted by reading files ranging from 1 MB to 100,000 megabytes. Table II presents the execution time and memory utilization with different file sizes.

TABLE II. TIME AND MEMORY UTILIZATION

File Size (MB)	**Resources**	
	Time (Second)	***Memory (GB)***
1	0.111	1.129
50	1.461	1.103
100	3.037	1.046
1000	6.779	0.693
2000	14.547	0.672
50 000	20.397	0.487
100 000	32.458	0.453

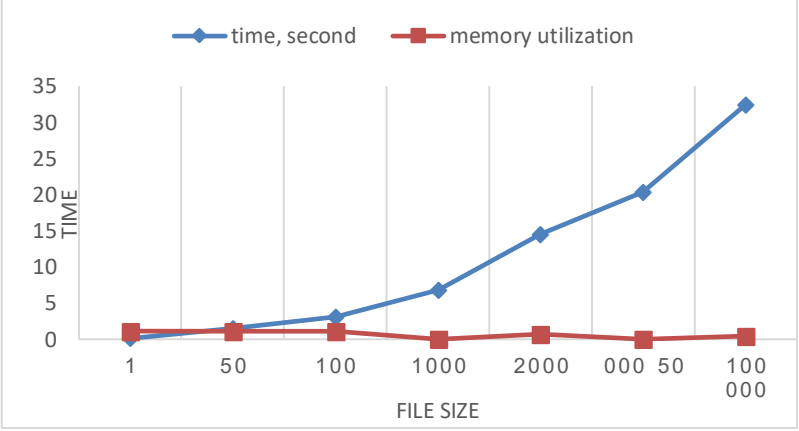

Fig. 5. Execution time and memory utilization of the proposed system

The table and graph above demonstrate that the execution time and memory utilization are influenced by the file size. The execution time increases as the file size increases; for small files (e.g., 1 MB), the execution time is as low as 0.111 seconds, but it increases to 32.458 seconds for larger files of 100,000 MB. Conversely, the memory utilization decreases as the file size increases; for a 1 MB file, the memory utilization is 1.129 GB, which decreases to 0.453 GB for a 100,000 MB file. This trend highlights the efficiency of the proposed system when handling larger files, as it utilizes memory more effectively due to optimizations designed to manage large datasets or files. These optimizations include compression techniques, which contribute to improved performance. The results also show that the number of DataNodes and the replication factor in the DFS significantly impact system performance. Increasing the number of DataNodes reduces latency by increasing the storage capacity, allowing the DFS to handle larger data volumes without capacity constraints or processing bottlenecks. This improvement increases throughput and reduces latency. Furthermore, adding DataNodes improves load balancing by providing the DFS with additional options for distributing workload and data across clusters. Additionally, the proposed system was compared with BlockHDFS [3] under the same environment. As shown in Fig. 7, the comparison reveals that the proposed system achieves better execution times for uploading files to the DFS and creating blockchain transactions than does BlockHDFS.

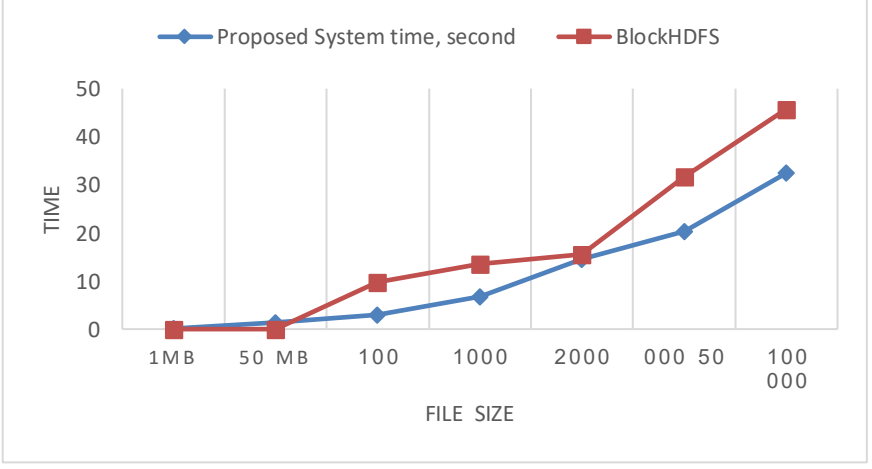

Fig. 6. Results of the comparison between the proposed method and BlockHDFS

The proposed system addresses several issues present in BlockHDFS, including the following:

- Real-time issues: In [9], metadata from WebHDFS are sent to the Hyperledger blockchain periodically after a set amount of time. This periodic approach introduces delays in metadata updates; in contrast, the proposed system creates and sends metadata to the blockchain immediately, enabling real-time operation.
- Security issues: WebHDFS is an HTTP-based interface that allows users to interact with HDFS over the web. Without proper access controls and authentication mechanisms, unauthorized access to metadata is possible.

Attackers can retrieve or modify sensitive metadata, compromising the confidentiality and integrity of the file system. The proposed system resolves this issue by creating metadata locally within the DFS and then securely uploading them to the blockchain. This approach ensures the security, confidentiality, and integrity of the file system.

7. LIMITATIONS

The proposed system was designed and implemented to address issues related to the architecture of traditional distributed file systems, making the DFS more efficient, scalable, and secure. However, the system still has several limitations, as outlined below:

- Execution time scalability: Although the proposed system achieves better execution times than other systems do in the study domain, the scalability of the execution time remains a concern. The processing time increases linearly with increasing file size, which may present challenges for extremely large datasets.
- Gas Fees in Public Blockchains: In public blockchain networks such as Ethereum, every transaction comes with costs referred to as gas fees. The regular updates to metadata in the DFS could lead to elevated gas expenses, which may affect the overall cost efficiency of the system.

8. CONCLUSION

Blockchain technology combined with a distributed file system is a great way to improve metadata management, thus enhancing data integrity, security, and transparency. The proposed system enhances traditional distributed file systems by integrating blockchain technology into their architecture, thereby addressing the single point of failure in the NameNode and improving the system's scalability, reliability, and security. The results indicate that the system is suitable for broad uses since our experimental evaluation reveals consistent memory consumption over several file sizes. Furthermore, compared with other current systems in the domain, the suggested system has improved execution times. However, as the file size increases, the need for extra optimization becomes clear, especially in terms of latency reduction and processing efficiency improvement. This work addresses fundamental problems, including scalability, latency, and storage overhead, to prove that blockchain is a long-term sustainable solution for DFS metadata management. The results confirm that by means of blockchain for metadata storage, safeguarding against modification or manipulation, the suggested system enhances the security and dependability of files in the DFS. Through tracking metadata changes and blockchain updating of new transactions, the system also offers real-time capabilities. As such, this is a new concept in the suggested system.

Conflicts of interest

The paper has no financial, personal, or professional conflicts of interest to declare.

Funding

The paper's acknowledgements section makes no mention of any sponsored or institutional financial assistance.

Acknowledgement

I would like to express my sincere gratitude and appreciation to the University of Kufa and the Faculty of Computer Science and Mathematics for their invaluable assistance in the preparation of this research. Their support and guidance have been invaluable in completing this work.

References

[1] J. Blomer, “A survey on distributed file system technology,” J. Phys. Conf. Ser., vol. 608, no. 1, 2015, doi: 10.1088/1742-6596/608/1/012039.

[2] R. Kumar and R. Tripathi, “Implementation of Distributed File Storage and Access Framework using IPFS and Blockchain,” Proc. IEEE Int. Conf. Image Inf. Process., vol. 2019-Novem, pp. 246–251, 2019, doi: 10.1109/ICIIP47207.2019.8985677.

[3] P. M. Dhulavvagol and S. G. Totad, “Performance Enhancement of Distributed System Using HDFS Federation and Sharding,” Procedia Comput. Sci., vol. 218, pp. 2830–2841, 2022, doi: 10.1016/j.procs.2023.01.254.

[4] X. Pan, Z. Luo, and L. Zhou, “Navigating the Landscape of Distributed File Systems: Architectures, Implementations, and Considerations,” Innov. Appl. Eng. Technol., pp. 1–12, 2023, doi: 10.62836/iaet.v2i1.157.

[5] V. Mothukuri, S. S. Cheerla, R. M. Parizi, Q. Zhang, and K. K. R. Choo, "BlockHDFS: Blockchain-integrated Hadoop distributed file system for secure provenance traceability," Blockchain Res. Appl., vol. 2, no. 4, p. 100032, 2021, doi: 10.1016/j.bcra.2021.100032.
[6] H. Guo and X. Yu, "A survey on blockchain technology and its security," Blockchain Res. Appl., vol. 3, no. 2, p. 100067, 2022, doi: 10.1016/j.bcra.2022.100067.
[7] X. Zhang and Y. Wang, "Research on intelligent medical big data system based on Hadoop and blockchain," Eurasip J. Wirel. Commun. Netw., vol. 2021, no. 1, 2021, doi: 10.1186/s13638-020-01858-3.
[8] A. M. Badr, L. C. Fourati, and S. Ayed, "A Novel System for Confidential Medical Data Storage Using Chaskey Encryption and Blockchain Technology," Baghdad Sci. J., vol. 20, pp. 2651–2671, 2023, doi: 10.21123/bsj.2023.9203.
[9] M. Al-Zubaidie and W. Jebbar, "Transaction Security and Management of Blockchain-Based Smart Contracts in E-Banking-Employing Microsegmentation and Yellow Saddle Goatfish," Mesopotamian J. CyberSecurity, vol. 4, no. 2, pp. 71–89, 2024, doi: 10.58496/mjcs/2024/005.
[10] H. Honar Pajooh, M. A. Rashid, F. Alam, and S. Demidenko, "IoT Big Data provenance scheme using blockchain on Hadoop ecosystem," J. Big Data, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00505-y.
[11] H. H. Abdul-Sada and Furkan Rabee, "The Genetic Algorithm Implementation in Smart Contract for the Blockchain Technology," Al-Salam J. Eng. Technol., vol. 2, no. 2, pp. 37–47, 2023, doi: 10.55145/ajest.2023.02.02.005.
[12] A. K. Tyagi, S. Dananjayan, D. Agarwal, and H. F. T. Ahmed, "Blockchain — Internet of Things Applications : Opportunities," Multidiscip. Digit. Publ. Inst., 2023.
[13] D. H. Tahayur and M. Al-Zubaidie, "Enhancing Electronic Agriculture Data Security with a Blockchain-Based Search Method and E-Signatures," Mesopotamian J. CyberSecurity, vol. 4, no. 3, pp. 129–149, 2024, doi: 10.58496/mjcs/2024/012.
[14] "Computational Intelligence and Neuroscience - 2022 - Nair - Blockchain-Based Decentralized Cloud Solutions for Data.pdf."
[15] M. K. Gupta and R. K. Dwivedi, "Blockchain Enabled Hadoop Distributed File System Framework for Secure and Reliable Traceability," Adv. Distrib. Comput. Artif. Intell. J., vol. 12, pp. 1–19, 2023, doi: 10.14201/adcaij.31478.
[16] M. S. Mohammed and A. N. Hashim, "Blockchain technology, methodology behind it, and its most extensively used encryption techniques.," Al-Salam J. Eng. Technol., vol. 2, no. 2, pp. 140–151, 2023, doi: 10.55145/ajest.2023.02.02.017.
[17] R. F. Ghani, A. A. S. Al-Karkhi, and S. M. Mahdi, "Proposed Framework for Official Document Sharing and Verification in E-government Environment Based on Blockchain Technology," Baghdad Sci. J., vol. 19, no. 6, pp. 1592–1602, 2022, doi: 10.21123/bsj.2022.7513.
[18] M. Ben Farah et al., "A survey on blockchain technology in the maritime industry: Challenges and future perspectives," Futur. Gener. Comput. Syst., vol. 157, no. April, pp. 618–637, 2024, doi: 10.1016/j.future.2024.03.046.
[19] A. F. Mahdi and F. Rabee, "A Blockchain Mining Proof of Work Approach Based on Fog Computing Virtualization for Mobile CrowdSensing," 2024 3rd Int. Conf. Distrib. Comput. High Perform. Comput. DCHPC 2024, 2024, doi: 10.1109/DCHPC60845.2024.10454071.
[20] M. Alessi, A. Camillo, E. Giangreco, M. Matera, S. Pino, and D. Storelli, "Make Users Own Their Data: A Decentralized Personal Data Store Prototype Based on Ethereum and IPFS," 2018 3rd Int. Conf. Smart Sustain. Technol. Split. 2018, 2018.
[21] S. S. Kushwaha, S. Joshi, and A. K. Gupta, "An efficient approach to secure smart contract of Ethereum blockchain using hybrid security analysis approach," J. Discret. Math. Sci. Cryptogr., vol. 26, no. 5, pp. 1499–1517, 2023, doi: 10.47974/JDMSC-1815.
[22] M. Bez, G. Fornari, and T. Vardanega, "The scalability challenge of ethereum: An initial quantitative analysis," Proc. - 13th IEEE Int. Conf. Serv. Syst. Eng. SOSE 2019, 10th Int. Work. Jt. Cloud Comput. JCC 2019 2019 IEEE Int. Work. Cloud Comput. Robot. Syst. CCRS 2019, pp. 167–176, 2019, doi: 10.1109/SOSE.2019.00031.
[23] K. Adel, A. Elhakeem, and M. Marzouk, "Decentralized System for Construction Projects Data Management Using Blockchain and Ipfs," J. Civ. Eng. Manag., vol. 29, no. 4, pp. 342–359, 2023, doi: 10.3846/jcem.2023.18646.
[24] Z. Wenhua, F. Qamar, T. A. N. Abdali, R. Hassan, S. T. A. Jafri, and Q. N. Nguyen, "Blockchain Technology: Security Issues, Healthcare Applications, Challenges and Future Trends," Electron., vol. 12, no. 3, 2023, doi: 10.3390/electronics12030546.
[25] M. Reena and S. More, "Integrated HDFS for secure traceability With Blockchain," 2022.
[26] F. K. Nishi et al., "Electronic Healthcare Data Record Security Using Blockchain and Smart Contract," J. Sensors, vol. 2022, 2022, doi: 10.1155/2022/7299185.

[27] G. Wan et al., “Decentralized Data Dominion : Unraveling the Power and Promise of Distributed File Systems To cite this version : HAL Id : hal-04670349 Decentralized Data Dominion : Unraveling the Power and Promise of Distributed File Systems,” 2024.

[28] H. Huang, J. Lin, B. Zheng, Z. Zheng, and J. Bian, “When Blockchain Meets Distributed File Systems: An Overview, Challenges, and Open Issues,” IEEE Access, vol. 8, pp. 50574–50586, 2020, doi: 10.1109/ACCESS.2020.2979881.

[29] G. Liao and D. J. Abadi, “FileScale: Fast and Elastic Metadata Management for Distributed File Systems,” SoCC 2023 - Proc. 2023 ACM Symp. Cloud Comput., pp. 459–474, 2023, doi: 10.1145/3620678.3624784.

[30] O. Ali, A. Jaradat, A. Kulakli, and A. Abuhalimeh, “A Comparative Study: Blockchain Technology Utilization Benefits, Challenges and Functionalities,” IEEE Access, vol. 9, pp. 12730–12749, 2021, doi: 10.1109/ACCESS.2021.3050241.

[31] Saihood, Ahmed, Mohammed Adel Al-Shaher, and Mohammed A. Fadhel. "A New Tiger Beetle Algorithm for Cybersecurity, Medical Image Segmentation and Other Global Problems Optimization." Mesopotamian Journal of CyberSecurity 4.1 (2024): 17-46.

[32] A. A. Almuqren, "Cybersecurity threats, countermeasures and mitigation techniques on the IoT: Future research directions," Journal of Cyber Security and Risk Auditing, vol. 1, no. 1, pp. 1–11, 2025.

[33] R. Almanasir, D. Al-solomon, S. Indrawes, M. A. Almaiah, U. Islam, and M. Alshar’e, "Classification of threats and countermeasures of cloud computing," Journal of Cyber Security and Risk Auditing, vol. 2025, no. 2, pp. 27–42, 2025. [Online]. Available: https://doi.org/10.63180/jcsra.thestap.2025.2.3