Blockchain Implementation for Storage

Jhanavi J[1], Dr. M Dakshayini[2]
[1]*(Department of ISE, BMSCE)*
[2]*(Professor, Department of ISE, BMSCE)*

Abstract

With the growing number of connected devices and with the emergence of IoT, data has become a critical component in day to day life, which further requires management, storage, and retrieval of enormous amounts of data. Increasing need to share data across business lines is another important player for the increasing demand for storage. Considering this global drive of commercial dependence on data comes the issue of data breaches from the centralized data centers. Hence it is required to adapt Decentralized storage. Since Blockchain as a technology has the feature of Decentralized management, it can be adapted to Decentralized storage as well. This paper presents the solution to Decentralized storage using Blockchain.

Keywords - *Blockchain, Decentralized storage, Swarming, Sharding, Smart Contract, Ethereum*

I. INTRODUCTION

Centralized database, even though it has just bits and bytes are a tangible thing. Database contents stored inside memory or disk of the computer system are vulnerable to data corruption even if it is a trusted organization like bank and governments.

This is where the core value of blockchain is enabled. With blockchain the database is distributed among the peers in the peer-to-peer network without the need of a centralized administrator. Blockchain uses clever cryptography to lock down the distributed database. Blockchain achieves this with the help of its own “proof of validity” and “authorization constraints”. This aid’s independent verification and processing of transactions. Currently single systems and cloud-based databases are highly centralized which makes them vulnerableto hackers.

II. RELATED WORK

Blockchain is a decentralized ledger or data structure. It can be referred as blocks in a chain where the corresponding blocks refer to the blocks, prior to them. Once the details of the transactions or events are fed into the Blockchain, it is impossible to tamper the details are shared with the members of the network.Users of the Blockchain network is completely aware of the transactions taking place. We will draw an analogy to justify the concept. We consider it to be a book-based data structure where each page of the book refers to its previous page by a page. Here, book refers to the Blockchain, page refers to the book and an entry in any page refers to the blockchain transaction. The Blockchain protocol have several features. Following are the few of them:

- Immutable: It means that it is really difficult to tamper or alter a block.
- Irreversible: This feature prevents double spending.
- Distributed system: It means that a copy of the ledger is present with all its members.
- No Centralized Authority: It doesn’t depend on a central server to dominate and hence, a peer to peer system.
- Resilient: This feature shows that it is not prone to any sort of major attacks.

A. *Disadvantages of Cloud:*

The problem with cloud storage is that the data must be stored somewhere which is usually on a computer or is replicated across several computers. Hence it is still a centralized system.

III. BLOCKCHAIN FOR STORAGE

The key complementary technologies helping to solve the issue of decentralized storage are Sharding and Swarming.

Sharding is where databases are portioned along logical lines and these shards are stored together and accessed by the decentralized application using a unique partition key.Sharding separates very large databasesinto smaller, faster, more easily managed parts called data shards as the word shard means a small part of a whole.In the simplest sense, Sharding database involves breaking upbig database into many, much smaller databases that share nothing and can be spread across multiple servers. For example, we can split a customer database geographically.

The governing concept behind Sharding is based on the idea that as the size of a database and the number of transactions per unit of time made on the database increase linearly, the response time for querying the database increases exponentially.

Additionally, the costs of creating and maintaining a very large database in one place can increase exponentially because the database will require high-end computers. In contrast, data shards can be distributed across several much less expensive commodity servers. Data shards have comparatively little restriction as far as hardware and software requirements are concerned.One common example is splitting a customer database geographically.

Sharding has the following benefits:

- Smaller index size
- Smaller working set
- Improved Performance
- Simple Infrastructure

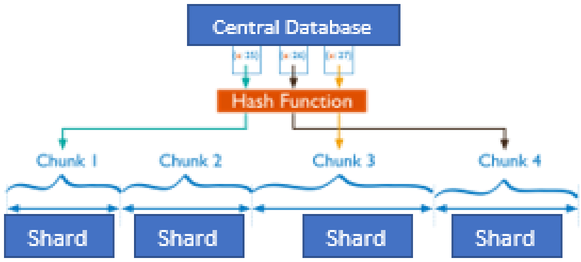

Figure 1: Creating Shards from Central Database using Hash Function

Hacking into one of them would reveal only a shard (fragment) of data without any context, since each computer is encrypted differently, and other shards are safe.

The collective storage of shards is accomplished by the second complementary technology, known as swarming.Just as blockchain utilizes a network of nodes, decentralized storageshould utilize large groups of nodes – referred to as "swarms" – to store and manage data. Swarms enable serverless hosting of data in a peer-to-peer network with storage and content distribution.

The swarm effect reduces latency and increases speed by retrieving data in parallel from the nearest and fastest nodes – much like torrents do. Because there are many, geographically dispersed nodes in a swarm, its reliability and scalability increase.

IV. IMPLEMENTATION

While it is affordable for a database to be softly linked with the blockchain (that is, not all database transactions must pass through the blockchain), it must be resistant to the malicious behavior of other DB nodes, provide sufficient level of replication, and have mechanisms to motivate participants to support the network.

As for the principles of data storage, it is proposed to provide DB with the following properties:

1. The database is public, the user (client) of the database is identified by its public key (the key of his blockchain account) - which is the user ID.
2. Each user can make transactions to the database, each transaction must be signed by this user.
3. A new record created by the user imprints his ownership of the data.
4. Change to the record after creation can only be made by the owner (or the user for whom the trust is established through the permissions mechanism, implemented as a smart contract on the blockchain).
5. Everybody can read all the records.
6. To ensure that there is no conflict between the records keys of different users, all the records keys of the user are prefixed with user ID.
7. More complex permissions can be configured using a smart contract in the blockchain (for example, trust between specific users, rights to create / delete tables), etc.
8. All permissions must be checked both for transactions and for replication.

The mandatory cryptographic signature of each record ensures that its modification or deletion by a malicious node without knowing the record owner private key is impossible. That is, thus constructed database is Byzantine fault tolerant even without a consensus protocol.

A. Motivation

The public network assumes that participants can freely join it, providing equipment that enhances computing power, storage capacity and network distribution. To stimulate such behavior, equipment owners should receive a reward motivating them to perform honest work.

Following rewards are supposed:

- Incentive for retrieving data
- Incentive for data storage

Rewards are paid by the user making inquiries. Main approach used is called “checkbook approach”. Before accessing the database, the user must reserve part of funds on a special smart contract "checkbook." Next, the address of this contract is used by the node to receive a reward - the "checkbook" contract stores the money of its owner and allows third parties to cash the signed checks by simply sending the transaction with the check as data to the contract method "cash".

- The contract tracks the total amount issued to each recipient at the time of the connection.
- The owner of a check must necessarily remember the total amount sent.

A check is cashed if

- The address of the contract corresponds to the address on the check.
- Check is signed by the owner (user ID - public key).
- The total amount on the check is greater than in the previous redeemed check for this recipient.

Then if it is necessary to reward the node the user just sends a check. The recipient node can save only the last received check from each user and periodically cash it out by sending it to a "checkbook" contract, which allows to conserve some of the blockchain transactions with some confidence.

B. Retrieval Incentive

The data on the DB nodes have a certain level of replication, that is, the data with a specific key is stored only on a part of the nodes, for example, on *N*. However, the user can connect to any node for the data. The node, to which the user connects, acts further as a "coordinator".

When the user makes request, by the value of the data key, the *N* nodes responsible for storing these keys are calculated and the requests are routed to them. The data returned by the nodes is checked by the coordinator for compliance with electronic signatures, compared by the time stamp, and the most recent record is returned to the user.

The work of the coordinator and replicas storing the data is subjected to payment. The proportions of payment are subject to more detailed calculation, but to stimulate correct behavior it is necessary to fulfill the following principles:

1. The faster the node returned data, the larger payment share it deserves
2. If the node returned old data, the payment decreases
3. The node, that did not return data, does not receive anything
4. The coordinator receives a fixed small share

To protect against malicious coordinators and users who will not pay, each node maintains a list of users from which it expects payment and coordinators sending requests from these users. If the debt level exceeds a certain threshold, the node may stop accepting requests from the specified users and coordinators. Upon receiving checks, the lists are corrected.

C. Storage Incentive

The reward for retrieval indirectly incentivizes storage of data but works only with relation to popular and often requested data. To stimulate long-term data storage, especially if data are rarely requested, a reward for storage is required.

The Ethereum Swarm describes the system of storage incentives. Nodes enter into a data storage contract with the information owner for a period. The storage can be paid at the time of data storage (update) or after a while, provided that the data is stored. If a loss of data is detected before the contract expiration, the node may be penalized, for which each node requires an initial registration with a security deposit.

When you save data, the node returns a receipt that proves that the node has accepted a file for storage. This receipt subsequently allows you to check whether the relevant data is still stored, and if not - to initiate a legislation, i.e. a special smart contract that will allow you to penalize the offensive node.

When the user initiates a data deletion operation, instead of physically deleting data, the data is replaced with a special "zero" record. The record can be physically deleted after expiration of its storage contract.

V. CONCLUSION

The considered DB can be attributed to the next generation of databases satisfying the mentioned principles:

1. ***Distribution*** DB supports an unlimited number of replicas, each of which can be a coordinator. That is, by referring to one of them, the user gets access to all data.

2. ***Publicity*** DB is designed to work in a public environment. New nodes can be added to the network and take on some of the load at any time.

3. ***Byzantine fault tolerance and resistance to other types of attacks in the public network*** Given that all data placed in DB is signed by their owner, the nodes cannot, at their discretion, change the data, nor can they corrupt data when replicating data to other nodes. Attempts to tamper with the data are immediately detected due to the electronic signature mechanism. For a tampering attempt a node-intruder may be deprived of the registration deposit and excluded from the network. To place the deposit, set access rights, the mechanisms for mutual settlements between the nodes, an external (for DB) blockchain is used, which should support Turing-complete smart contracts.

4. ***Shard support (the ability to replicate only a fraction of the data on each node to increase the maximum total amount of data)*** Each DB node is responsible for a certain interval of primary keys that it stores. The level of replication (the number of nodes storing copies of data with the same primary key) is specified separately and can grow with the growth of the network.

5. ***Ability to store structured data*** Data in DB supports the structure. This can be a JSON document with a structure that meets the needs of a specific application.

6. ***Ability to delete data*** Data deletion is supported in DB. You cannot guarantee instant removal, but in the end, with good behavior, the data will be deleted. A malicious node can deliberately store all the data that is deleted. However, it will not be able to do it for all the data, because it only receives requests in a certain range of primary keys.

This DB can be used in different decentralized projects. It relies on a blockchain supporting smart contracts. In this regard, it can be used for the needs of decentralized applications built on top of a variety of the blockchains such as Ethereum, etc.

REFERENCES

[1] A Blockchain-Based Access Control System for Cloud Storage' Ilya Sukhodolskiy, Sergey Zapechnikov Department of Cryptology and Cybersecurity National Research Nuclear University MEPhI (Moscow Engineering Physics Institute) Moscow, Russia

[2] 'Blockchain-based System for Secure Data Storage with Private Keyword Search' Hoang Giang Do School of Computer Science and Engineering Nanyang Technological University do0004ng@e.ntu.edu.sg

[3] 'Ensuring Data Integrity Using Blockchain Technology' Igor Zikratov, Alexander Kuzmin, Vladislav Akimenko, Viktor Niculichev, Lucas Yalansky ITMO University Saint-Petersburg, Russia

[4] 'An Overview of the Emerging Technology: Blockchain' Rishav Chatterjee School of Computer Engineering KIIT University Bhubaneswar, India rishavpiku@gmail.com

[5] 'Blockchain: Challenges and Applications' PinyaphatTasatanattakool Faculty of Science and Technology Rajamangala University of Technology Suvarnabhumi Bangkok, Thailand

[6] 'Blockchain Technology, Bitcoin, and Ethereum: A Brief Overview' by Dejan

[7] 'Sustainable Blockchain-Enabled Services: Smart Contracts' by Craig Wright

[8] 'Legally Speaking: Smart Contracts, Archival Bonds, and Linked Data in the Blockchain' by Darra L. Hofman