Advances in Parallel Computing Algorithms, Tools and Paradigms
D.J. Hemanth et al. (Eds.)
© 2022 The authors and IOS Press.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/APC220008

Decentralized Web Hosting Service Using IPFS and Ethereum Blockchain

S.Muthurajkumar[a,1], A.Vignesh[a], S.Kugan[a] and R.Arunsha[a]
[a]*Madras Institute of Technology (MIT) Campus, Anna University, Chennai, India*

Abstract. On the Internet, web applications are served from a centralized location i.e., server, for higher maintainability. However, in the centralized architecture, if there is an occurrence of server failure or crash, the web applications cannot be serve to the end-users until the server goes live again. In addition, in the existing centralized architecture for web hosting services, integrity of the hosted websites entirely relies on the third-party applications which checks for any possible threats in the system. In order to provide data integrity within the system and to overcome the above-mentioned single point of failure, we proposed the decentralized solution for hosting web applications, which provides more data availability to the end-users and maintains the integrity of the data. The proposed model makes use of the Interplanetary File System (IPFS) for storing and retrieving web applications, which provides high availability and reliability. In addition, the proposed model uses the Blockchain Technology for authenticity and confidentiality. The smart contracts are deploy on the Ethereum Block chain, which aids the service provider to manage the hosting service system. The proposed model also comparatively decreases the time taken to transfer the file over the IPFS using optimal path-finding algorithm. The proposed algorithm has a lesser time complexity when compared to the Bitswap protocol used in IPFS. The use of blockchain with IPFS cumulatively provides better authenticity via Ethereum Smart Contracts, which reduces risk and failure.

Keywords.IPFS, Block chain, Web Hosting, service provider

1. Introduction

The decentralized apps are generally build with the help of blockchain technologies. In case of decentralized applications, the Interplanetary File System (IPFS) provides the data storage and retrieval in a decentralized and distributed architecture. The main purpose of using blockchain technology, Ethereum blockchain, Smart contracts, IPFS are precisely explain in the following sections.

A Blockchain is a decentralized, distributed, and oftentimes public, digital ledger consisting of records called blocks that are link using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data. By design, a blockchain is resistant to modification of its data, because once recorded, the data in any given block cannot be alter retroactively without alteration of all subsequent blocks. The above-mentioned mechanism allows the participants to verify-

[1] S. Muthurajkumar, Madras Institute of Technology (MIT) Campus, Anna University, Chrompet, Chennai, India; E-mail: muthurajkumarss@gmail.com

and audit transactions independently and relatively inexpensively. They are authenticate by mass collaboration powered by collective self-interests. Such as design facilities robust workflow where participants' uncertainty regarding data security is marginal. The use of blockchain removes the characteristic of infinite reproducibility from a digital asset. It confirms that each unit of value was transfer only once, solving the long-standing problem of double spending. A blockchain has been describe as a value-exchange protocol. A blockchain can maintain title rights because it provides a record that compels offer and acceptance.

Ethereum is a cryptocurrency just like bitcoin. The currency is generally refer to as ethers. The ethers are validate by a proof of work that are rewards for the miners. Every account has an ETH balance. Ethereum users are classified into two namely user accounts and contracts. Every account has the functionality to create and call a contract. To validate transactions, encryptions are used. These encryptions used here are symmetric encryption. Generally, transactions are sign with the sender's private key. Here we used truffle to simulate the Ethereum. On each transaction, the Ethereum virtual hardcode itself with a gas cost which is proportional to the utilization of resources. Gas prices is the number of ethers paid to miners per unit. Higher the gas price, more incentive for a miner to add in blockchain.

Smart contracts are mainly use as triggers as they perform a series of actions that has to be automate without involvement of humans. The main objective of the smart contract avoids third party interference in the work. Smart contracts are mainly use in case of transactions (money exchange) as they are more trustable with no man in the middle problem. In our work, we used smart contacts in verifying the storage miner node and during performing transactions.

2. Literature Survey

Tas R and Tanrıover O O [11] had proposed a decentralized web application solution that have developed with blockchain. They had simulated the blockchain with truffle, a development environment, Ganache is a personal block chain for Ethereum development and this used to deploy contracts, develop applications, and run tests and solidity to write smart contracts. Gourisetti S N G, Mylrea M and Patangia H [2] discussed about the different types of blockchains and provided a comparative study between the different algorithmic approaches. Provides deep insight and comparison of multiple commonly used mechanisms based on user requirement

In [13] had proposed a modified blockchain framework, which aims to reduce the size of blocks in the blockchain to improve decentralization. Weber I, Staples M, Zhu L, Bosch J, Bass L, Pautasso C and Rimba P [12] havedeveloped about the architecture and the algorithms that used in the blockchain and listed out some of the challenges and difficulties by comparing blockchain protocols in different respects and provided some existing approaches to solve the problems.

Guerar M, Merlo A, Migliardi M, Palmieri F and Verderame L [3] have proposed a model that projects their end outcomes in which securing data using blockchain and providing a decentralized way using IPFS. Zheng Q, Li Y, Chen P and Dong X [14] had proposed an IPFS-based blockchain based data storage model to solve storage issues in blockchain using IPFS to improve the scalability. The data volume of blockchain grows continuously due to the features that cannot deleted and can only added. Hence, the miners deposit the transaction data into the IPFS network and pack

the returned IPFS hash of the transaction into the block. Utilizing the characteristics of the IPFS network and the features of the IPFS hash, the blockchain data has greatly reduced.

Kumar R and Tripathi R [7] had proposed a model to solve storage issues in IPFS based blockchain. IPFS creates hash for all the files in their system. The system generally accesses using distributed hash table. Kumar R, Marchang N and Tripathi R [6] had proposed a decentralized solution for maintaining the records of the patients rather than storing in a centralized database. Jianjun S, Ming L and Jingang M [5] had proposed a solution for data sharing with the help of Ethereum block chain and IPFS. Here they had used block chain to provide the integrity and verification of the users and for faster access and better availability, IPFS has used as they are in decentralized manner.

Singla V, Malav I K, Kaur J and Kalra S [9] had proposed a leave application with the help of Truffle based on Ethereum with smart contracts. Chen Y, Li H, Li K and Zhang J [1] had proposed a storage model to improve the blockchain storage model. IPFS was a peer-to-peer version-controlled file system that synthesizes learnings from successful systems. Huang H, Lin J, Zheng B, Zheng Z and Bianhad J [4] summarized about the blockchain based distributed file systems and about IPFS and swarm systems discussing the open issues and series of challenges that constrain their development.

Pham V, Tran C, Nguyen T and Nguyenhad T [8] had proposed a secure decentralized model, whichhas built in the combination of IPFS, ABE and MA-ABE. The encryptions standards has used to encrypt the document that was need to be share between multiple organizations. Steichen M, Fiz B, Norvill R, Shbair W and State R [10] had proposed a blockchain based extension to IPFS to provide access control (ACL-IPFS). The ACL-IPFS leveraged smart contracts in Ethereum to handle the access control list. The user could register files, grant or revoke access to through the smart contracts.

3. Proposed System Architecture

As of the literature survey we did, the centralized architecture i.e., Client-server architecture used in web hosting services cannot provide high data availability and confidentiality.

The hosted website is going to be down when there is server maintenance or if the load of the server gets high. Due to these drawbacks in the existing architecture, we proposed a decentralized web hosting service which makes use of the IPFS network. The proposed model transfers the web application, which are upload to the system in a time comparatively lesser than the Bitswap protocol with the help of optimal path finding algorithm.

The architecture diagram in Figure 3.1 depicts the web hosting service aided by IPFS and Ethereum Blockchain. In IPFS network, there are three types of nodes namely, storage miner node, retrieval miner node and gateway node.

- Storage Miner Node: The node provides the storage space required to store the files uploaded in the IPFS network. In addition, it provides the pinning service to save the file and skip this file during the garbage collection.
- Retrieval Miner Node: The node acts as an intermediary for transfer of data from storage miner node to gateway node and vice versa. The node offers the temporary storage space to increase the data availability in the network.

• Gateway Node: The gateway node converts the incoming HTTP requests to its equivalent IPFS hashes and vice versa using IPNS.

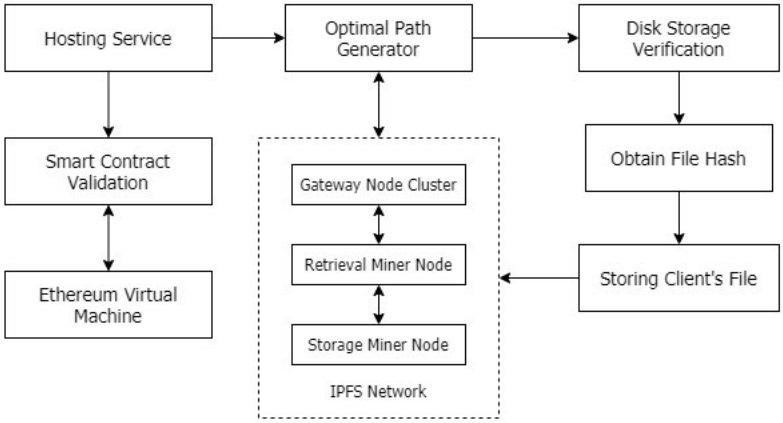

Figure 1. System architecture.

The proposed model consists of four main components namely, Hosting Service Administration, Optimal Path Generator, Disk Storage verification, Smart Contract validation. At first, the hosting service provider deploys the smart contracts on the Ethereum blockchain network denoting the ether cost provided for the storage miner nodes and received from the service requester nodes. After successful registration as a service requester, he/she uses the hosting service to upload his/her web application to be host in the network. Before uploading, the service requester's authenticity is validate and verified using the smart contracts. After successful validation, the optimal path generator finds the shortest path from the service requester's node to the nearest storage miner node, which has the storage capacity greater than the uploaded file's size. After this process, the file is upload to the network using the generated path.

4. Proposed Algorithm

The proposed model consists of three main parts namely, building a user interface using Angular, distributed file system using IPFS and providing authenticity and confidentiality using smart contracts deployed in Ethereum blockchain. At first, we built a web application using Angular framework, which includes the basic building blocks for a website i.e., HTML and CSS for imparting the web hosting service interface. Using this interface, the service requesters can register themselves for requesting the hosting service for a certain period of time, which costs him of about 0.5ETH, and the storage providers can register themselves for renting their local storage as an IPFS storage that in turn rewards him/her with an amount of 0.25ETH. The service provider can manage both the service requester and the storage providers

using this interface. The service requester can manage the files that he/she has uploaded to the network using this web application. The storage provider can also view the files that are permanently stored and temporarily stored in cache using this web interface.

If the account has sufficient balance, then the requester is successfully register in the system. Then the amount is transfer to the service provider, otherwise the service requester is notify with an alert message saying insufficient balance.

```
Algorithm: Register_Hosting_Service(host_owner, service_requester)
Input: Host_Owner: Owner of the deployed contract, Service_Requester
Output: Transaction_Status
1:  if service_requester.balance<host_service_amount then
2:  service_requester.notify('Balance Insufficient')
3:     return false
4:  end if
5:  host_owner.transfer(host_service_amount)
6:  host_owner.register_service(service_requester, 'requester')
7:  return true

Algorithm: Register_Storage_Node(host_owner, storage_provider)
Input: Host_Owner: Owner of the deployed contract, Storage_Provider
Output: Transaction_Status
1:  if host_owner.balance<storage_node_amount then
2:  host_owner.notify('Balance Insufficient')
3:     return false
4:  end if
5:  storage_provider.transfer(storage_node_amount)
6:  host_owner.register_service(storage_provider, 'storage')
7:  return true

Algorithm: Optimal_Path_Finder (source_node, file_size)
Input: Source_Node: IPFS Node, File_Size: Size of the file in bytes
Output: Shortest_Path: Shortest Path to Nearest Storage Miner Node
1:  queue = [ ] , visisted_nodes = [ ] , shortest_path = { }
2:  Enqueue(queue, source_node)
3:  for each node in queue do
4:  current_node = Dequeue(queue)
5:       for each peer in Adjacency_List(current_node) do
6:            outdegree = current_node.outdegree + peer.outdegree
7:            if peer not in visisted_nodes then
8:  visited_nodes.add(peer)
9:  Enqueue(queue, peer)
10:           else if Parent(peer).outdegree<current_node.outdegree then
11: Replace(Parent(peer), current_node)
12:           end if
13:           if Node_Type(peer) equals Storage_Node and
Storage_Capacity(peer) >= file_size and
shortest_path.outdegree< outdegree then
14:                Update shortest_path with current_path
15:           end if
16:      end for
```

17: end for
18: return shortest_path

As mentioned in the above algorithm, initialize an empty queue with the source node as the first node. After initialization, pop the first element from the queue and iterate through each adjacent node in the current node's peer list. While iterating, calculate the out-degree (the number of outgoing edges from the node) of each node along the path and store it. After calculating, check whether the current node is already visited or not. If the node is not visit, update the path and enqueue the visited node, otherwise check whether the out-degree calculated along the path is larger than the out-degree calculated until the visited node's parent. If the condition satisfies, replace the visited node's parent with the current node.

5. Implementation and Result Analysis

The proposed model could effectively transfer data from the source node to the destination node in a lesser time compared to Bitswap Protocol. The optimal path-finding algorithm performs the search for the nearest storage node from the source node. If two nodes are of same length from the source node, the algorithm makes use of the out-degree computed along the path and the node, which has the highest out-degree in its path, is chose from further exploration until the storage node is found, else the node is not consider for exploration and not added in the queue. In Bitswap protocol, the algorithm recursively traverses through all the nodes in the peer list and updates globally the identified storage node. If the storage node is find, the algorithm stops and updates the path along the node. In the worst case, if the storage node is present deep in the network graph, the Bitswap algorithm visits every node in the graph which increases the overall time complexity of the algorithm. In the case of our proposed optimal path algorithm, if two nodes occur in the same level from the source node, then the algorithm stops exploring the node with minimum out-degree along the path which significantly reduces the overall time complexity.

Table 1. Hosting Services and their Gas Cost in Ethereum Blockchain

Contract Name	Gas Amount (ETH)	Contract Year
Hosting Service Registration	0.5	1
Storage Node Registration	0.25	1

The proposed model also imparts the registration of storage providers. The storage provider can use this system to register his node as a storage miner node in the IPFS network. Each storage miner node is reward with an amount of 0.25ETH from the service provider. Table 1 summarizes the hosting services and their gas cost in Ethereum Blockchain. The Figure 4.8 picturizes the transfer of the amount from the service provider to the storage miner node. Before transferring the amount, a contract is sign between the storage provider and service provider. The graph shown in Figure 2 compares the time complexity Bitswap and Optimal path-finding algorithm.

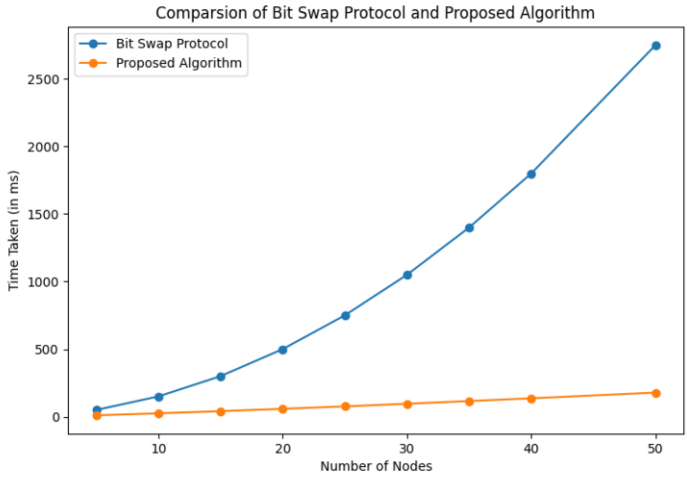

Figure 2 Comparison of Bitswap Protocol and Proposed Algorithm with Respect to the Number of Nodes in the Network

The amount of time taken to transfer the file from the source node to the storage miner node in our proposed model decreases when the size of the file which has to be uploaded increases. The time taken to transfer the file of size 1MB from node 1 to the nearest storage miner node i.e., node 4 in Bitswap and our proposed algorithm is 118ms and 98ms respectively. The graph shown in Figure 3compares the time taken to transfer the file with the size of the file, which has to be upload in the network.

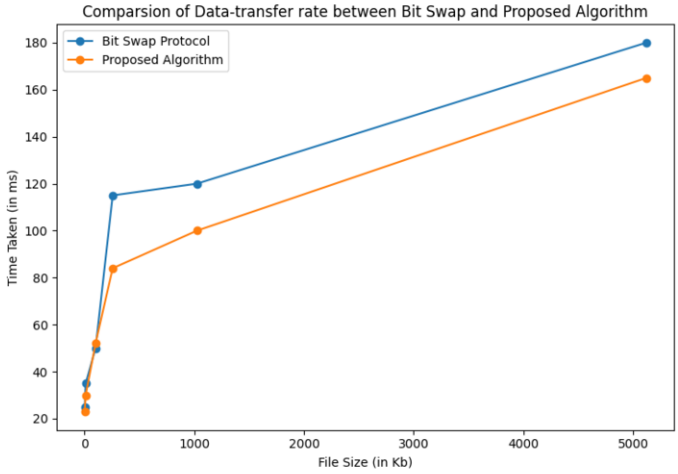

Figure 3 Comparison of Bitswap Protocol and Proposed Algorithm with Respect to the Time Taken and Uploaded File Size

6. Conclusion and Future Work

In this work, we proposed an efficient decentralized approach for hosting websites using IPFS and blockchain. With the IPFS, a distributed storage have provided where the contents are stored via optimal path find algorithm in the nearest storage miner node. To maintain the integrity and authenticity, blockchain is used. With the combination of IPFS and blockchain technology, decentralized applications can built in many different forms, which provides better data availability and integrity.Hence, it is advisable to safeguard the data in a better manner by an encryption standard and the files to be password protected. The encryption needs to done at the time of uploading the file, which provides better confidentiality and it, is decrypted when the file is requested from the network using the file's hash and client's secret key.

References

[1] Chen Y, Li H, Li K and Zhang J. An improved P2P file system scheme based on IPFS and Blockchain. 2017 IEEE International Conference on Big Data (Big Data); 2017. p. 2652-2657.
[2] Gourisetti S N G, Mylrea M and Patangia H. Evaluation and Demonstration of Blockchain Applicability Framework. IEEE Transactions on Engineering Management, 2020; 67(4):1142-1156.
[3] Guerar M, Merlo A, Migliardi M, Palmieri F and Verderame L. A Fraud-Resilient Blockchain-Based Solution for Invoice Financing. IEEE Transactions on Engineering Management. 2020; 67(4):1086-1098.
[4] Huang H, Lin J, Zheng B, Zheng Z and Bian J. When Blockchain Meets Distributed File Systems: An Overview, Challenges, and Open Issues. IEEE Access. 2020; 8: 50574-50586.
[5] Jianjun S, Ming L and Jingang M. Research and application of data sharing platform integrating Ethereum and IPFS Technology. 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), Xuzhou; 2020; p. 279-282.
[6] Kumar R, Marchang N and Tripathi R. Distributed Off-Chain Storage of Patient Diagnostic Reports in Healthcare System Using IPFS and Blockchain. 2020 International Conference on COMmunication Systems &NETworkS (COMSNETS); 2020; p. 1-5.
[7] Kumar R and Tripathi R. Implementation of Distributed File Storage and Access Framework using IPFS and Blockchain. 2019 Fifth International Conference on Image Information Processing (ICIIP); 2019; p. 246-251.
[8] Pham V, Tran C, Nguyen T and Nguyenhad T. B-Box - A Decentralized Storage System Using IPFS, Attributed-based Encryption, and Blockchain. 2020 RIVF International Conference on Computing and Communication Technologies; 2020; p.1-6.
[9] Singla V, Malav I K, Kaur J and Kalra S. Develop Leave Application using Blockchain Smart Contract. 11th International Conference on Communication Systems & Networks (COMSNETS); 2019; p.547-549.
[10] Steichen M, Fiz B, Norvill R, Shbair W and State R. Blockchain-Based, Decentralized Access Control for IPFS. 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData); 2018; p.1499-1506.
[11] Tas R and Tanrıover O O. Building A Decentralized Application on the Ethereum Blockchain. 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT); 2019; p. 1-4.
[12] Xu X, Weber I, Staples M, Zhu L, Bosch J, Bass L, Pautasso C and Rimba P. A Taxonomy of Blockchain-Based Systems for Architecture Design. 2017 IEEE International Conference on Software Architecture (ICSA); 2017; p. 243-252.
[13] Jayaraman, Indumathi, and Mokhtar Mohammed. (2020g): Secure privacy conserving provable data possession (SPC-PDP) framework. Information Systems and e-Business Management, 18(3), 351-377.
[14] Zheng Q, Li Y, Chen P and Dong X. An Innovative IPFS-Based Storage Model for Blockchain. 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), Santiago, Chile, 2018; p. 704-708.