DISTRIBUTED DECENTRALIZED DATA STORAGE USING IPFS

Vijayalakshmi M[1], Hemang Thakur[2], Aritra Chattopadhayay[3]

[1]*Asst. Professor, #CSE Department, SRM Institute of Science & Technology*
[2]*RA1511003010284, #CSE Department, SRM Institute of Science & Technology*
[3]*RA1511003010378, #CSE Department, SRM Institute of Science & Technology*

---***---

Abstract **–** Distributed Decentralized Data Storage is a type of data storage network that depends upon IPFS (Inter Planetary File System)Protocol to save the files. The files are encrypted on the client side before upload. Anything uploaded on the IPFS network generates a hash. This hash is stored on the server. This server is also running an IPFS daemon to access the IPFS network. When a file is to be uploaded it is first sent to the server, running PHP backend, The file is encrypted here and passed on to the IPFS daemon, which then uploads this file to IPFS network and returns a hash to refer to the file, which then gets stored on the server for further reference(upload and download).

There are provisions to periodically check for data integrity, storage sharing and a feature of revenue for sharing storage.

Keywords **–** *Decentralized, peer-to-peer, IPFS, client-side encryption, cloud storage.*

I. INTRODUCTION

We aim to develop a decentralized data storage facility that can store and retrieve data as and when required.

Any file to be saved online can be locally encrypted then uploaded on this network. Then the file is broken into pieces and are stored on different devices on the network, maintaining 51% redundancy to avoid data loss. To keep track of each piece we use smart contracts based on the Blockchain technology.

Decentralisation can be understood as transfer of authority from any central authority to a pool or group of nodes. The construct itself has been around for a short time associated an earlier construct can be paralleled to the introduction of the web wherever the unfold of knowledge was democratised. The term is currently being coined against Blockchain technologies and applications like Bitcoin and Ethereum that change monetary transactions and computing power.

Storage is outlined because the retention of recoverable information on a laptop or alternative electronic system. we tend to use storage on a usual from the transportable and computers and it's simply understood from the files we tend to place onto a USB stick. From the times of getting to place files on a disc to having the ability to put files within the 'cloud', storage has return a protracted approach.

A. Decentralized Storage Basics

It is an arrangement of having the capacity to store your documents without answering on extensive, concentrated storehouses of information that don't undermine vital qualities, for example, protection and privacy of your data.

A Decentralized distributed cloud is a decentralized model of organized online capacity where information is put away on various PCs (hubs), facilitated by the members participating in the cloud.

For the decentralize plan to be feasible, the absolute storage contributed in total must be at any rate equivalent to the measure of capacity required by end clients. Nonetheless, a few nodes may contribute less capacity and some may contribute more. There might be remunerate models to repay the nodes contributing more.

II. OBJECTIVE

To Create an online file storage system that can act as a content hosting and sharing platform. This can also be used an online Drive for storing and retrieving data such as files and media. This platform serves as a node to access the IPFS network to store data.

This also includes privacy and data security for the data stored. Data is encrypted before upload, hence even if anyone is able to access the hash of the file still the data is protected with the password which is in the user's mind.

III. DISTRIBUTED STORAGE TECHNOLOGY

Decentralized and distributed storage is a potential solution that Blockchain companies are researching and trying to implement. It is a system that stores data on a distributed network, containing many private or publicly owned nodes contributing to the total storage overcoming the problems of violation of privacy and physical protection of data .

This system can be compared to Torrents and torrenting software like LimeWare. Decentralized storage networks operate in a similar manner but with further advanced cryptography and encryption as well as the added incentive mechanisms.

A. InterPlanetary File System (IPFS)

The InterPlanetary File System (IPFS) is a peer-to-peer distributed file system that seeks to connect all computing devices with the same system of files. In some ways, IPFS is like the web, but IPFS can be seen as a single BitTorrent swarm, exchanging objects within one Git repository. In other words, IPFS provides a high throughput content-addressed block storage model, with content addressed hyperlinks. This forms a generalized Merkle DAG, a data structure upon which one can build versioned file systems, blockchains, and even a Permanent Web. IPFS combines a distributed hash table, an incentivized block exchange, and a self-certifying namespace. IPFS has no single point of failure, and nodes do not need to trust each other. We'll use IPFS protocol in our system to store and share files

IV. CONSTRAINTS

The following are the constraints for the proposed system:-

1) Data processing is a bit slower on IPFS as compared to centralized systems.

2) Decentralized system will have no control over the authenticity of data or user.

V. DESIGN

A. Conceptual Design

We are able to create a network where each node shares its storage adding to a combined pool of storage available i.e. more nodes in the network, more storage space for data. Absence of central storage reduces the factor of unavailability since this network is always available since each node will be owned by a user who may be located anywhere in the world and full-fledged failure at all places is very unlikely. Any Computational load will be shared across all nodes on the network resulting in reduced load on all. All data stored is first encrypted locally and then uploaded making it impossible for other nodes holding data to access the data without the right Credentials.

A major part of our value proposition is in its unique architecture. Where most networks implement a single, standard type of node, we propose a dual solution.

1. Storage Nodes: Anyone can lease additional hard drive space on their PC. We will utilize that hard drive space to store little, encoded bits of other people groups' records. In a perfect world, we need to boost medium sized capacity nodes. Enormous server farms would be excessively unified and rout our main goal.

2. Recovery Nodes: Retrieval nodes are one of a kind to our system. They are found near the storage nodes on the system. In any case, recovery nodes need high data transfer capacity and low dormancy. They get impetuses on the off chance that they are the quickest to find and recover a record for a client. Practically speaking, this reasonable methods the best recovery nodes will work on quick associations in high-thickness populace focuses where numerous storage nodes work all the while.

As a client, you'll indicate the file you need to transfer. From that point onward, the encryption calculation scrambles your file and partitions it into numerous fragments. These different portions get sent to different focuses on the system that have your file. So as to review every one of the bits of your file, you should know the file's hash. Along these lines, just the file proprietor has the hash so just the file proprietor can discover, reassemble, and unscramble the file.

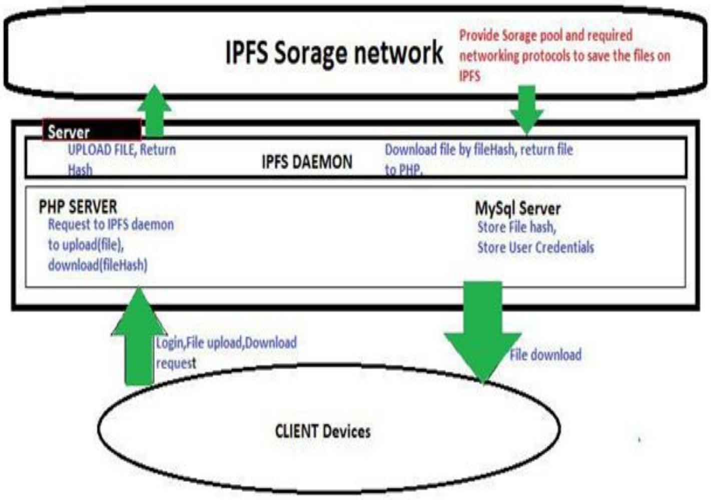
IPFS Sorage network
Provide Sorage pool and required networking protocols to save the files on IPFS
Server
UPLOAD FILE, Return Hash
IPFS DAEMON
Download file by fileHash, return file to PHP.
PHP SERVER
Request to IPFS daemon to upload(file), download(fileHash)
MySql Server
Store File hash, Store User Credentials
Login,File upload,Download request
File download
CLIENT Devices

Fig. 1 Decentralized Storage Architecture

B. Collaborative Design

In this project we are going to use decentralized and the distributed types together and for the internal communication we will be using PHP.

While storing data in present available methods (data centers) there may be a situation where the data centre building itself may be compromised. The datamay get stolen or destroyed. our system avoids any such kind of problem with a simple solution of not keeping everything at the same place. This is where IPFS comes in.

IPFS runs on a peer to peer network. All devices connected to it(called node) act as server as well as storage provider. any file is stored in lots of shards or fragments which are encrypted and connected to each other of one file. also there are multiple copies of each shard which ensures redundancy.All the shards can be called to form the file by calling the first shard by it's hash. All shards are found and assembled and the file is served.

Since computational load for processing requests and transfers are handled by each node on the network, resulting in less load for each node. The overall network is much more efficient and can support large loads.

VI. ADVANTAGES AND DISADVANTAGES

A. Advantages

The following are the advantages of the proposed decentralized system: -

1) Always available
2) *Robust:* 51% of the network must be taken down before the system deactivates which is not possible
3) *Security:* AES-256 Encryption
4) *Storage:* Theoretically infinite storage
5) High efficiency
6) Low workload
7) *Redundancy:* Distributed storage provides data distribution and redundancy. Stored on multiple locations

B. Disadvantages

The following are the disadvantages of the proposed decentralized system: -

1) Slow: A bit slower than centralized data storage solutions.
2) Size Limit: Max limit on file size exists i.e. 10GB for a single file
3) Permanent web: One thing to note here is, any data uploaded to IPFS is permanently stored there, until 51% nodes of IPFS are shutdown. Hence, it is a permanent web.

VII. REQUIREMENT ANALYSIS

A. Hardware

Just a single computing node to serve the contents and services to the other nodes in the network is required.

B. Software

1. IPFS Daemon
2. PHP Server
3. MySQL Server

C. Working

1) *Uploads:* The following are the processes or modules that are currently working as intended: -

- User logs into the network, using authorized credentials.
- Uploads a file through the provided interfaces.
- PHP server receives the file and is stored in the server cache
- This is then encrypted by using SHA-256 encryption with the user's password.
- The server then uploads the file to IPFS.IPFS is accessed through IPFS Daemon running on localhost on the same server
- IPFS returns a hash for the uploaded file
- This Hash is stored in MySQL server for later reference to the files

2) *Downloads:* The following are the processes or modules that are currently working as intended: -

- The home page of the application also shows the uploaded files.
- When the user wants a file he just needs to click the download button with the particular file name.
- This triggers a "File Fetch" request to the IPFS Daemon with the stored hash in the MySQL server for the respective file
- This file is then downloaded on the PHP server cache and served to the user for download
- Once download is complete the file is deleted from the server cache.

IX. TECHNOLOGIES USED

The following are the technologies used in the development of the proposed system: -

1) *Database:* MySQL
2) *Networking Stack:* IPFS Daemon for uploading files to IPFS and XAMPP Server for running PHP and MySQL
3) *PHP:* Used in user registration, login, logouts, upload, download and session tracking.

X. CONCLUSIONS

The biggest benefit of the project is significant decrease in operating costs. For example, the cost of storing 1TB of data on AWS is approximately $20. Using the above technologies, it is possible to bring down the cost of storage approximately ten times!

Redundancy of data i.e. the data is safe and securely stored unless 51% of the network goes down or is attacked which is highly unlikely and extremely tough to do which means 100% safety and security.

Since data is encrypted before uploading, data stored cannot be accessed by *anyone* except only the person for whom it is meant for.

REFERENCES

[1] Protocol Labs. Technical Report: Proof-of-Replication. 2017.

[2] Gregory Maxwell, Proof of Storage to make distributed resource consumption costly. https://bitcointalk.org/index.php?topic=310323.0

[3] Giuseppe Ateniese, Randal Burns, Reza Curtmola, Joseph Herring, Lea Kissner, Zachary Peterson, and Dawn Song. Provable data possession at untrusted stores. In Proceedings of the 14th ACM conference on Computer and communications security, pages 598–609. Acm, 2007.

[4] B. Cohen. Incentives build robustness in bittorrent. In Workshop on Economics of Peer-to-Peer systems, volume 6, pages 68–72, 2003.

[5] medium.com

[6] coincentral.com

[7] cadasta.org

[8] cointelegraph.com

[9] cloudonlinestorage.net

[10] media.vectorzilla.io

[11] www.redmountainmakers.org