Peer-to-Peer File Storage System using InterPlanetary File System

Manoj Athreya A[1], Ashwin A Kumar[1], Nagarajath S M[1], Gururaj H L[1],

Ravi Kumar V[1], Sachin D N[1] and Rakesh K[1]

[1] Computer Science & Engineering, Vidyavardhaka College of Engineering, Mysuru, India
{manojathreya.cs@vvce.ac.in, ashwinkumar.cs@vvce.ac.in, nagarajath.cs@vvce.ac.in,
gururaj1711@vvce.ac.in }

Abstract. In today's web, HTTP is the default protocol to transmit files. It works efficiently to move small files. But HTTP fails to implement other more efficient file distribution techniques. The InterPlanetary File System (IPFS) is a peer-to-peer globally mounted filesystem which addresses the challenges which can resolve data reliability, fault-tolerance, consistency and non-repudiation issues in the current system. In this paper, we have proposed an innovative and efficient way of storing and retrieving the files on web using IPFS, adding or uploading files is made more easier and assures high security to data. Many approaches are made where the files are stored in a distributed system but the approach using decentralized storage protocol is not used and it provides a new dimension to the application.

Keywords: Blockchain, Decentralized, Distributed, peer-to-peer network, IPFS, IPFS cluster.

1. Introduction

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. IPFS works similar to BitTorrent where the objects are exchanged in a single versioned controlled repository. In other words, IPFS is block storage model which provides high-throughput with content addressed hyperlinks, where the address of the link is based on the content it holds. This block storage model is formed by a generalized type of data structure called Merkle DAG, using which versioned file systems are built. [11] IPFS works by combining a distributed hash table (DHT), and a self-certifying nature of file system which is ensured but public key cryptography. There is no single point of failure in the IPFS, and nodes do not need to trust each other. [1]

As the world is moving towards decentralization revolution, a lot of dapps are developed every day. A true dapp needs a truly decentralized storage with a decentralized content distribution network. IPFS implies the rules on the movement of data and content in its network. This file system layer offers properties such as:

- the web pages that are truly distributed.

- the websites will have no original server to respond to the requests that can run entirely on client side browsers. [2]

Content Addressing

The method of content addressing is used in IPFS which identifies content in the network. In today's web the addressing is different where the same function is performed using IP addressing. IPFS objects are the entities present in the IPFS address which contains a list of IPFS Links and content data which is being addressed. The large files added to IPFS is broken into many smaller chunks of data and resolved to an array of IPFS links that points to the broken pieces of the original data. This type of addressing in the IPFS network ensures that the address specified will always results in the request to the same file. [12]

Another advantage of the content addressing system is that as long as the content hosted by a node is available, it can be retrieved from the IPFS network. This eradicates the problem of broken link which exists in today's internet. IPFS does not host duplicate links or files because the links point to the same hash value which is given as the result of the addressed contents. Thus, to retrieve a file from the network, one copy of the file is enough which can be hosted by any node across the network. [1][9]

IPFS Objects

The IPFS address is resolved through the data structure called IPFS objects which has two fields: Data and links.

Data- information or content which is converted into binary data of size less than or equal to 256 kB. Links- consists of link structures. which points to other IPFS objects present in the IPFS network. A Link in the IPFS object has data fields which involves fields such as name, hash and size. [10]

Name- the name given to the link.

Hash- the value of the hash generated for the IPFS object.

Size- The IPFS object size is specified using this field. [1]

Directory Structures

A directory in IPFS is an immutable globally mounted and is represented by IPFS object links which points to files or other directories.

Versioned File Systems

IPFS makes use of similar data structure that enables versioned file system used in git. The commit object in the versioned file system has one or more links with names which points to previous commits, and the link which contains name object which references the file system structure which is initiated by that commit.

Distributed Hash Tables

Distributed Hash Tables (DHT) are considered to be a database which is distributed in a peer to peer network where with the help of keys associated with the values can be stored and retrieved. Without any central coordinating force the nodes in the network balance and store the data. Distributed Hash Tables are fault tolerant that enables a system to continue operating properly even if there is a failure and resilient when key/value pairs are replicated. [7]

Blockchains

Nowadays, Bitcoin's technology is seeking more attention towards its second part and how the underlying technology of blockchain can be used for just more than money and transaction. There are other applications that has incorporated blockchain. Name coin is a service which represent decentralized name registration database which provides mechanism of identification of accounts and allowing users to create their own cryptocurrency. Another service called colored coins which is a more advanced application providing decentralized exchange, where a token can be traded for other tokens and provides blockchain identity. Thus, the current trends in blockchain involves building private side chain of a network or implementing a protocol on top of blockchain. [2][8]

2. Related Work

The potential solution that the blockchain companies are researching is in the field of decentralized storage. Decentralized storage networks make use of the cryptography and encryption algorithms which ensures the integrity of the data stored in the network. The projects that are trying to overcome the decentralized storage problems are listed below.

Swarm

Swarm is a decentralized protocol developed and maintained by Ethereum. It works alongside Whisper for Messaging, which is a decentralized messaging protocol. Swarm protocol provides storage for Ethereum’s public record which is stored in a decentralized manner, which mainly consists of the distributed dapp code and data which also includes blockchain data. [6]

BigchainDB

Improving the speed and efficiency of transactions, is the main objective of the most blockchain solutions. But, BigchainDB does not attempt to improve the blockchain protocol and takes a different route which utilizes existing consensus engine called tendermint to handle the P2P communication, and

stores data in the nosql database MongoDB, through which powerful queries on the data can be made. Multiple ownership and native support on the digital assets are provided. [3]

Storj

This is another decentralised storage platform powered by the Ethereum network. Storj is a platform as well as a cryptocurrency, and consists of dapps or decentralized apps. which are the applications runs on a distributed system. The technology in storj is similar to bit-torrent which uses bitswap protocol. When a user wants the file, they request is sent to Storj which uses DHT to locate all the links and clubs them together. Before sharing the files, the files are encrypted and the ownership is verified using the uploader's private key. [5]

Siacoin

Sia is a project which was started in 2013 at HackMIT and launched in 2015.They aim to create an efficient storage solution around the globe by leveraging the free space of the hard capacity which enables a marketplace for the data, where free hard disk space can be rented through the other peers in the network. Siacoin uses a unique type of smart contract called "file contract" where the hosts get paid for proving that they have stored a file. All data is encrypted before being uploaded to the network, and only the user can decrypt them again. [4]

3. Proposed Scheme

The main module of any software is the storage of data in the database. The control over the database is the major priority among many other factors and becomes the master if the control given to them. Blockchain technology involves blocks where the data is stored inside blockchain network. The copy of the data in the blocks will be received when a node joins the network. So there is no particular master of the data in this technology.

A. IPFS Primer

Normal Uploading:
To create an uploading feature, a developer needs to receive data from the browser and then store it somewhere. It could be a cloud service or other file hosting service which allows a developer to write server code to modify the image for example. There also could be multiple storage solutions. All these solutions increase the amount of bandwidth used by the application. A 1 mb upload becomes 2 mb because the server needs to upload it onto the storage solution.

Fig 3.1

IPFS uploading:
For uploading to IPFS, the standalone browser can be used where data is pushed onto IPFS. Through this the previous 2 MB upload in the client/server architecture becomes 1 mb. This ensures or saves the network costs. For the successful upload of the data to IPFS, the file reader and buffer class are used and the recent implementation of the IPFS API comes pre-packaged with these classes.

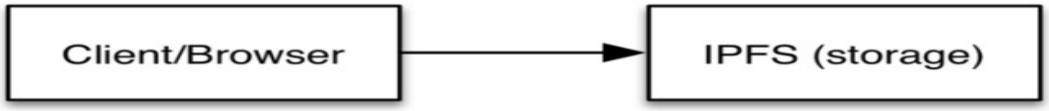

Fig 3.2

Connecting to IPFS:
To connect to IPFS either local or remote node, the IPFS package must be installed and the following commands enables the connection to the IPFS network,
ipfs init: The command will initialize the ipfs repository and node will be created along with key-pair and a repository containing ipfs objects.

ipfs daemon: The IPFS daemon spins up local IPFS network through the browser.The daemon also provides a Web UI for the IPFS node created locally. The default URL of the web UI is 127.0.0.1:5001/webui.

B. IPFS Cluster:

Collective pinning and composition for IPFS. IPFS cluster is a piece of software that enables coordination between IPFS daemons on different hosts. IPFS Cluster run on raft consenus algorithm. IPFS cluster consist of 2 main applications namely:

ipfs-cluster-service: for starting your cluster peer.

ipfs-cluster-ctl: for interacting with the cluster peer through various inbuilt APIs.

There are two ways to connect peers in cluster, one setting a predefined peerset or bootstrapping nodes.

ipfs-cluster-service init: IPFS Cluster peers are run with the ipfs-cluster-service command, more precisely service.json, which manages the cluster behavior, along with the peerstore file which stores the peers multi addresses. Those files can be found inside ~/.ipfs-cluster. This command will create cluster configuration $userprofile/.ipfs-cluster/service.json. This file will be used later to start a new cluster or join existing one.

ipfs-cluster-service daemon: This IPFS cluster command will start the cluster peer where raft will be initialized with the init_peerset. Peers will elect a Raft Leader and then become ready to receive data from other peers.

C. IPFS API

In this paper, we are using the python IPFS API which is a HTTP client library for IPFS.
To be able to allow cross-origin resource sharing (CORS) configuring ipfs is necessary to return headers for CORS to work. This can be done through the following commands:

1.ipfs config --json API.HTTPHeaders.Access-Control-Allow-Methods '["PUT", "GET", "POST", "OPTIONS"]'

2.ipfs config --json API.HTTPHeaders.Access-Control-Allow-Origin '["*"]'

To be able to upload the files to the IPFS and successfully retrieve the data back, the file reader and the buffer at client side is used.
File Reader allows you to read files in different formats such as readAsArrayBuffer, readAsBinaryString, readAsDataUrl, readAsText.

Buffer at the browser side is used for reading or manipulating streams of binary data. Buffer class instances are fixed-sized and has raw memory allocations which resembles to the arrays of integers. The buffer size is initialized at the time of creation and cannot be changed.
To create the application the python micro web framework,
flask is used through which different routes are created for the events of uploading and viewing the files that are uploaded to the IPFS network. Python implementation of IPFS HTTP client library provides useful functions to manipulate the streams of data in the IPFS network.
The connection to the IPFS is made through the function connect(),which takes five arguments:

ipfsApi.**connect**(*host*, *port number*, *base='api/v0'*, *chunk_size=4096*, ***defaults*)

The client or host is a TCP client for interacting with IPFS deamon. A Client instance will not actually establish a connection to the daemon until at least one of it's methods is called which is specified in the connect function. The connect function may throw the following exceptions if the connection to the IPFS client is not possible: Version Mismatch ,Error Response, Connection Error, Protocol Error, Status Error or Timeout Error

To add a file or directory to the IPFS the function add() is used which takes care of file reader and buffer.
add(*files*, *recursive=False*, *pattern='**'*, **args*, ***kwargs*)

Parameters:
files (str) -- A file path to either a file or directory.
recursive (bool) -- Controls if files in subdirectories are added or not
pattern (str | list) -- Single *glob* pattern or list of glob patterns and compiled regular expressions to match the names of the file paths to keep.

After adding a file the hash value of the file is returned which can be accessed through the get() function which downloads a file, or directory of files from IPFS and files are placed in the current working directory.

Fig 3.3

Fig 3.4

The added files undergo the process of pinning where the nodes in the network pin the uploaded files or documents which ensures that the files do not vanish in the network. IPFS nodes treat the data stored like a cache, which means that there is no guarantee that the data will continue to be stored. Pinning tells IPFS network that the data is important and should not be garbage collected. Planning can be done using pin_add() which pins objects to local storage and returns a list of IPFS objects that have been pinned.

pin_add(*path*, **paths*, ***kwargs*)

Parameters:
path (str) -- Path to object(s) to be pinned.

recursive (bool) -- Recursively unpin the object linked to by the specified object(s)

Distributed private storage network can be setup using IPFS cluster which is a standalone application and a CLI used to allocate, replicate and track Pins across a cluster of IPFS daemons.

IPFS Cluster is an orchestration tool or software to gain control of the IPFS daemons or the nodes present in the IPFS network. An IPFS Cluster is created among number of nodes and the nodes share a pinset which contains CIDs which are cluster-pinned and their properties.

Cluster peers communicate using libp2p. To be able to exchange messages using libp2p, that each node contains a private key and has its own Peer ID. To ensure that the nodes in the network only exchange messages with known parties, an additional secret key is shared among all nodes in the clustered IPFS network.

4. Conclusion

In this paper, as proposed we have used a decentralized storage, IPFS for reliable and user-friendly interactions with the ipfs network. Since the data is distributed among the ipfs nodes, assures more security to the data as it is secured by the implementation of the hashing algorithm used in cryptography and in turn the data is stored in nodes of different computers in a network. The focus is mainly on components that strengthen the functionality of web based distributed file storage and retrieval of stored files. The IPFS clustering service ensures a swarm of nodes which can be orchestrated for the collective pinning mechanism which ensures the validity of the files that are uploaded to the network. The deployment of specialized agents, such as search agents for specific applications, or the proposal of improved network topologies and protocols are some of the performance improvement opportunities to explore.

References

[1] Juan Benet. “IPFS - Content Addressed, Versioned, P2P File System”,2014

[2] Antonio Tenorio-Fornés Samer Hassan Juan Pavón, “Open Peer-to-Peer Systems over Blockchain and IPFS: an Agent

Oriented Framework”, CryBlock’18 ,2018, Munich, Germany.

[3] “BigchainDB 2.0 the Blockchain Database”, Whitepaper,BigchainDB GmbH, Berlin, Germany, May 2018

[4] David Vorik, Luke Champine, “Sia:Simple Decentralized storage”, November 2014

[5] Storj Labs, Inc., “Storj: A decentralized cloud storage network framework”, October 30, 2018.

[6] Phillip Pieper, Timo Lehes, “Swarm: The blockchain for private equity”,White Paper ,April, 2018

[7] Bastien Confais ,Adrien Lebre, Benoît Parrein, “An Object Store Service for a Fog/Edge Computing Infrastructure based on IPFS and Scale-out NAS”, 2017 IEEE/ACM 1st International conference on Fog and Computing Edge Computing (ICFEC)

[8] Shangping Wang, Yinglong Zhang and Yaling Zhang, “A blockchain based framework for data sharing with fine-grained access control in decentralized storage system ” ,IEEE ACCESS , June 29 2018.

[9] Qiuhong Zheng, Yi Li and Ping Chen, “An innovative IPFS- based storage model for blockchain”, 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI).

[10] Sawood Alam, Mat Kelly, and Michael L. Nelson, “InterPlanetary Wayback: The Permanent Web Archive”,Joint Conference on Digital Libraries(JCDL) 2016.

[11] Yongle Chen, Hui Li , Kejiao Li and Jiyang Zhang , “An improved P2P File System Scheme based on IPFS and Blockchain”, 2017 IEEE International Conference on Big Data (BIGDATA)

[12] Muhammad Salek Ali, Koustab Dolui and Fabio Antonelli , “IoT Data Privacy via Blockchains and IPFS”, IoT 2017: the Seventh International Conference on the Internet of Things.