

Performance Study of IPFS over Http(S) Using the Multi-Cloud Platform

Ajil A^{1*}, Shubham kashyap², Rohith SS³, Kumari Nisha Rani⁴

^{1,2,3,4}School of Computing and Information Technology, Reva University, Bangalore, India

Corresponding Author: ajil.a@reva.edu.in

DOI: <https://doi.org/10.26438/ijcse/v7si14.409415> | Available online at: www.ijcseonline.org

Abstract- Recent researches have proved that existing internet protocols like HTTP have been used widely and extensively in our day to day life which has led to poor and inefficient performance of the web. This study focuses on a new network protocol that is Interplanetary File System abbreviated as IPFS. IPFS is distributed and peer to peer file system which stores files in a decentralized way unlike http as well as keeps a track of versions pretty much like Git. IPFS uses Distributed Hash Table for the purpose of routing, Merkle DAG Data Structure for establishing links between two nodes in the form of cryptographic hashes and Bit Torrent for the exchange of data in bits from different peers. These functionalities make it possible for IPFS to transfer data with high speed and reliability. The unique hash values ensure data immutability and prevents violation of data security. In other words IPFS is the replacement for HTTP which uses centralized server to download files leading to all time crashing of the server, slow internet and unreliability in all terms.

Keywords- Interplanetary File System, Hypertext Transfer Protocol, Merkle Dag, BitSwap.

I. INTRODUCTION

The web is the backbone of our civilization. The wide and extensive usage of internet demands high speed and reliability in this modern era of technology where we demand survival for the fittest and fastest. We all are familiar with HTTP, the current internet protocol which has been in picture for more than 20 years and has resulted in inefficient performance, expensive bandwidth costs, and redundancy in file and so on. Hence in order to establish a reliable protocol, IPFS has been introduced which aims in fixing all these issues globally[5].

1.1 InterPlanetary File System

InterPlanetary File System is an open -source distributed file System that uses peer -to -peer networking architecture rather than the client-server architecture that is used by HTTP. IPFS moves web towards a decentralized server with faster transmission of data and low cost. It moves us to the permanent web[fig2]. A web where link never dies. While our HTTP web is IP addressed and IPFS is content addressed. When we are requesting data from content addressed we receive it faster since it get route from whoever owns the copy of the data closes to us. This makes will lose that knowledge forever. The major problem with HTTP is the most common error 404, which signifies either the data has been deleted or moved. HTTP links gets broken all the time. If data isn't manually copied by someone that data is gone forever. Last problem is Centralization that is data is located in a single location which leads to high internet traffic.

bandwidth usage much more efficient. When you upload a file to IPFS you get back its immutable hash. Each file in the network is identified by its content using this hash. The hash is cryptographically guaranteed to represent only that file. Anyone can provide storage for IPFS data. The storage providers are incentivized with a crypto token and data is replicated and chunked across all of them by a network to maintain data permanence. To achieve this IPFS combines several ideas from Computer Science in the past few years which will be discussed later in this project[5].

1.2 Analytical Problem

The Problem with the existing web is that it uses a protocol called HTTP that is Hypertext Transfer protocol developed by Tim BernersLee in 1989. It delivers all types of files and is considered a client-server protocol. The first problem with http is Bandwidth. More and more of the world is coming online fast and that means more people are demanding data. Websites crash all the time because too many people try to access the data at the same time from a single server. Hence it leads to higher bandwidth costs. Second problem is resiliency, data is not permanent. If Wikipedia was hacked and all its data deleted we will

1.3 Motivation

IPFS should be introduced to upgrade the web and possibly replace HTTP because in this era of technology even a single point of failure could lead to loss of huge amount of useful data. IPFS is different from other file systems. Once we save file in this system it can be accessed by anyone in the world in a very easy manner. Inspired by GIT it keeps tracks of all the versions of the file stored at any point of

time, which makes it flexible in all terms. It is considered as singleton storage of data with security as well. It overcomes the problem of centralization as well since the data here is distributed in bits to many peers which increases the availability of data all the time. And we are sure it will be able to overcome all the challenges that the current internet is facing. IPFS will reduce the cost of data transmission which will prove to be a remarkable upliftment in the growth of a company hence leading to the world-wide development.

II. LITERATURE SURVEY AND RELATED WORK

- Idea about the IPFS protocol-<http://ipfs.io/>
After a few research works, we were able to conclude that the web is constantly evolving and it has become the backbone of the current civilization. Basic communication channels depend on the web to function. The web is just a collection of protocols and protocols are set of rules and ideas on how to build and transmit data that have own the hearts of developers. The process of writing protocols ended up being adopted by everybody else because they work really well. There is one protocol in particular that is in need for a serious upgrade that is HTTP protocol developed by Tim Berners Lee. We use HTTP to access most of the web. It connects all of our computing devices together such as phones, tablets, laptops which works very well, but we began to see issues of http like bandwidth, latency, resiliency, centralization and so on. IPFS has been designed to fix the problems and helps us more towards permanent web, where the link never dies and no entity controls our data. While HTTP is Location addressed, IPFS is content addressed. It uses Distributed Hash Table (DHT) to store data and a Merkle dag to give its structure and a BitTorrent mechanism to exchange data[5].
- Updates on Torrent and Bit Torrent sites-
<https://torrentfreak.com/bittorrent-surges-to-150-million-monthly-user-120109/> -
BitTorrent protocol is a file transfer protocol widely used for distributing data over the internet. Bit refers to the binary data which makes up files, torrent describes how the transmission is done. Since the number of BitTorrent users is more the protocol is particularly efficient at the transfer of large files and therefore it is commonly used to distribute audios and videos. For this reason, the protocol is recognised as a significant enabler of online piracy for the protocol to work, first a small torrent file is created which contains the information needed to locate and retrieve the data to be transferred. Those who have access to this torrent file are able to download the data of peers by using the information contained in the file to make a connection, these connections are facilitated by a torrent client which interprets the information contained in the torrent files. Once someone opens a torrent file for one of these clients they become a peer and enter the swarm of peers transferring data

between each other .This is why BitTorrent is referred to as a p to p protocol.

III. EXISTING AND PROPOSED SYSTEM

◆ Source System (System Currently in Use):

- HTTP protocol has been widely used since its development to protect transmitted data. It has been the default protocol for conducting financial transactions on the web[1].
- The HTTP protocol is used to provide a standard way for web browsers and servers to talk to each other.
- The usage of this protocol has been some limitations, mainly
 1. Centralization-There are lot of companies that own most of our data and there are only few companies which provide access to our data and this can cause lot of problems. If Facebook goes down lot of us can access any of our photos and if google goes down lot of us can access documents that were collaborating on. Centralization bring another problem with it and that is censorship, because content is hosted on just a few sources its easy for government to block access to them[fig2].
 2. Latency-HTTP provides poor performance due to network latency.
 3. Offline access-Many services in the internet do not have offline access. That is if google goes down we cannot collaborate on the documents we were working on.
 4. Location based addressing- Location based address are also an issue with the internet, if many people access to same data on a single server it may strain on a network and cause latency.
 5. HTTP is very inefficient and expensive.

◆ System in Design (Proposed system):

- The IPFS (Interplanetary File System) which is still under recognition for usage, is a Distributed file system.
- IPFS causes a Hypermedia distribution of protocol.
- The web completely distributed by running it on a peer to peer network that works similarly to how Bittorrent works. If we think about the p2p distributed network were data leaves on the edge of the network on your local machine. As your browsing the internet using an IPFS is actually downloading the data that you access, so you're having it locally on your machine and serving it simultaneously. So your both client and host[5].

• Benefits of using IPFS protocol:

1. Aims to create a permanent web where links are content addressed not IP address.
 2. It uses a Distributed hash table to store data, a merkle dag to give it structure and a Bittorrent mechanism to exchange data.
 3. IPFS is content based addressing.
 4. It is permanent and Decentralized system.
- IPFS is a new peer to peer hypermedia protocol that aims to replace the Hypertext transfer protocol that rules the we now.

The websites which store data in the internet is actually not a good system, you have to have a protocol like how data moves.

But that might about to change meet IPFS (Content based addressing).The web completely distributed by running it on a p2p network that works similarly to how Bittorrent works.

IV. METHODOLOGY

IPFS works by connecting all computing devices with the same system of files via nodes. IPFS uses a distributed hash table, incentivized block exchange, and self-certifying namespace.

In simpler terms, it acts similar to a torrent system, except that instead of sharing and exchanging media data, IPFS exchanges git objects. Any type of content can be added and it will give a key that can be used to retrieve the content again at any time. This is what IPFS allows for content addressing instead of location addressing. The key is completely independent and different and can be hosted anywhere [fig1].

Every file has a unique hash key which can be compared to a fingerprint, when you want to download a certain file you just ask the network who has the file with this hash and someone on the IPFS network will provide it to you (So security built in).What makes this possible? That is because of a data structure called “Merkle Dag”[1].

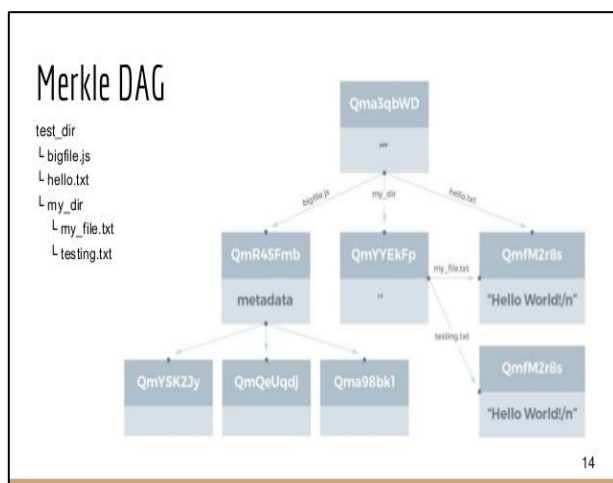


Fig1: Data Structure if IPFS

We have root directory that has pointing to three different resources hello.txt, my_dir and bigfile.js and we can already start to see some of the things happening with IPFS we see hello.txt and my_file.txt points to the same object because the content of the object is identical and the hash is identical so we are already getting some duplication on the network we are saving a little sore storage space also bigfile.js is getting chunked into three smaller pieces, IPFS does this

automatically it chunks big files to smaller pieces to make them easy to serve over the network, so actually when you access bigfile.js it concatenates the chunks together to serve one file and allows you to get to use the peer to peer network to be served from multiple people.Right when you access content IPFS it looks for the peers you are connected and so it can bring the content into your locally. Similarly how Bittorrent works.

Offline access in IPFS:

By opening some Bill English in Wikipedia on IPFS and off the data and reload it, I can still have the same page because its cached locally on my IPFS node, now for every page that I have been to on this website i have locally, so if Wikipedia goes out of business if jimmy wales can't get those donations, I'm serving Wikipedia now you all can still access, So we can think about how this is really addressing a lot of the problems that i talked about the beginning and the potential this kind of technology have.

Is this the future of the web? This is a really young technology there is not even a 1.0 version. Sometimes the imprints of the internet works to our benefit if our data gets leaked, published we kind of want that not be there anymore.IPFS is an open source project there's also making a javascript implementation. So I think this presents us with a really cool opportunity to rebuild in the way that we browse the web and the way we interact with the internet.

V. SYSTEM ANALYSIS AND DESIGN

HTTP is the most widely used protocol today but it seems to be getting older and unable to keep up with the fast-growing web.

Drawbacks of Hypertext Transfer Protocol

- The past decade has seen a constant increase in the number of internet users. And with this, the number of users requesting data is also increasing gradually. To provide these amount of data to the users, HTTP uses a huge amount of bandwidth available, even if the people who request the data are in proximity to each other.This is not just efficient enough considering the huge amount of data available.
- A transfer of data cannot surpass the speed of light. Hence, the reasonable thing to do would be to bring the servers closer. Thus, when certain data is required, it would reach soon. However, if we are request some data and coincidentally it is located in a system just a few meters away from us, it would be faster if the data is just taken from that system rather than going through the backbone of the web and the request is transferred to a closer city.
- Everyone has seen a **404 error**, which means the content is broken or deleted. The major problem we are facing with HTTP links is that it is unreliable and the

links get broken a lot of times. Also, if the content is deleted, it is lost forever unless someone has manually copied the data.

The biggest problem that HTTP lies in the fact that it is centralized[fig2]. This gives immense power to a few individuals who own the data and access to the data can be easily restricted by the governments and certain individuals if they wish to do so[5].

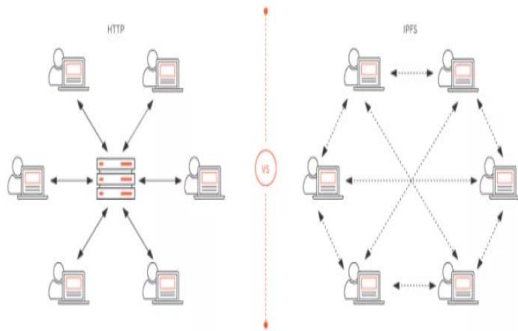


Fig2: Client server network architecture (http) versus peer to peer network (ipfs)

5.1 How does IPFS solve this issue?

The current generation of the Internet is not nearly as decentralized as it was idealistically and initially perceived to become. It is also predicated on some outdated protocols that have led to a lot of issues. The issues addressed by IPFS revolves around those associated with the current HTTP protocol of the Internet today[fig4].

5.1.1 Distributed Hash table

A Hash Table is a data structure that stores information as keys or value pairs. In Distributed Hash Tables, the data is distributed across so many computers, and efficiently coordinated to enable efficient access and lookup between the nodes. The main advantages of this system is decentralization, fault tolerance and scalability. The Nodes in this do not require central coordination, the system can function normally even when nodes fail or leave the network, and DHTs can scale to accommodate lots of nodes. However together these features result in a system that is generally more resilient than client-server structures[6].

5.1.2 Block exchanges

BitTorrent is a one of the popular file sharing system that is able to successfully coordinate the transfer of data between millions of node by relying on an innovative data exchange protocol. However, the system is limited to the torrent ecosystem. IPFS implements a generalized version of this protocol called a BitSwap, which can be operated as a marketplace for any type of data[fig3]. This marketplace is

the basis for the type Filecoin, a p2p storage marketplace built on IPFS[2].

5.1.3 Merkle Dag

A Merkle DAG is a combination of a Merkle Tree and a Directed Acyclic Graph (DAG). Merkle trees ensures that data blocks exchanged on p2p networks are correct, undamaged and unaltered. This verification is done by organizing data blocks by using cryptographic hash functions. This is a simple function that takes an input and calculates a unique alphanumeric string (hash) corresponding with that input. It is easy to check that whether the input will result in a given hash, but incredibly difficult to guess the input from a hash[fig3].

A Distributed hash table is a way to model topological sequences of information that have no cycles. A simple example of DAG is a family tree. A Merkle DAG is basically a data structure where hashes are used to reference data blocks and objects in a Distributed hash table. This creates several useful features, all contents on IPFS can be uniquely identified since each data block has a unique hash[1].

5.1.4 Version control systems

Another powerful feature of the Merkle DAG structure is it allows you to build a distributed Version Control System (VCS). The most popular examples of this is Github, which allows developer to easily collaborate on projects simultaneously. File on Github is stored and versioned using a Merkle DAG. It allows users to independently duplicate and edit multiple versions of a file, store these versions and later can be merged edits with the original file. Interplanetary file system uses a similar model for data objects as long as objects corresponding to the original data, and any new version are accessible, the entire file history can be retrieved accordingly. Given that data blocks are stored locally across the networks and can be cached indefinitely, this means that the IPFS objects can be stored permanently[fig3].

5.1.5 Self-certifying systems

The final component of IPFS is the Self-certifying File System (SFS). It is a distributed file system that doesn't require special permissions for data exchange. It is self-certifying because data served to clients is authenticated by the file name. Therefore, one can be able to securely access remote content with the transparency of local storage[fig3]. IPFS builds on this concept to create the InterPlanetary Name Space (IPNS). It is an SFS that use public-key cryptography to self-certify objects published by users of the network. All objects on IPFS can be uniquely identified, but this also extends to node. Each node on the network will have a set of public keys, private keys and a node ID which is the hash of its public key. Nodes can, therefore, use their private keys to 'sign' any data object they publish, and the

authenticity of this data can be verified by using the sender’s public key[3].

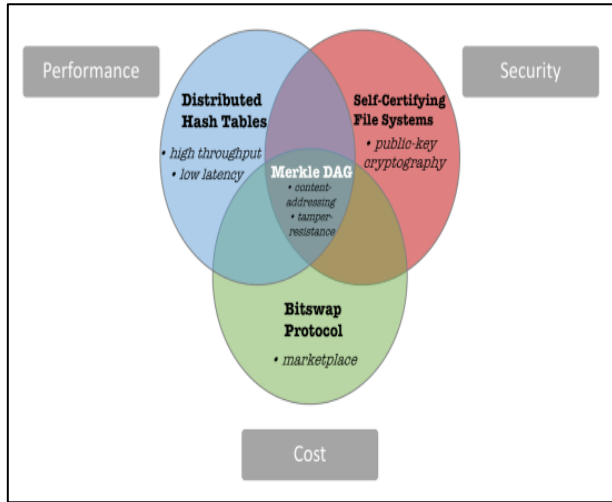


Fig3: Underlying architecture of ipfs protocol

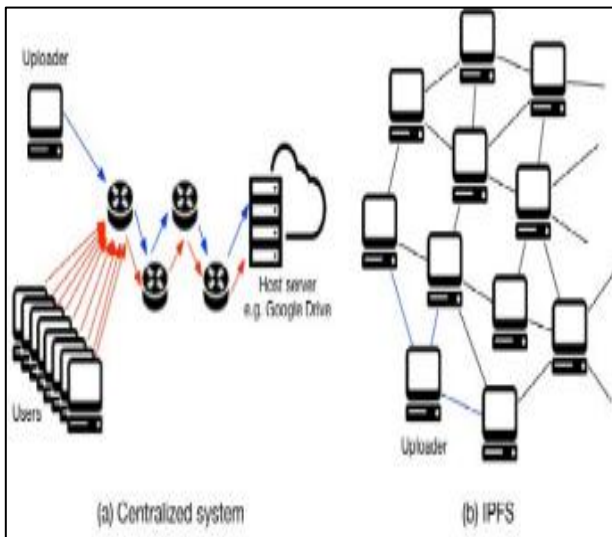


Fig4: A complete analysis of the distributed web

5.2 How IPFS stores data?

Every file has a unique hash key, which can be compared to a fingerprint, when you want to download a certain file you just ask the network who has this file with this hash key and someone on the IPFS network will provide it to you[fig5].

Files are stored inside the IPFS objects and these objects can store up to 256 kb worth of data and can contain links to each other IPFS objects. A simple “Hello World” text file, which is very small, can be stored in a single IPFS object. But larger files bigger than 256kb like image or videos well those are split up into multiple IPFS objects that are all each 256 kb in size and afterwards the system will create empty IPFS objects that links to all the other pieces of the file. IPFS uses content based addressing once something is

added, it cannot be changed or manipulated anymore it is an immutable data store much like a blockchain[4].

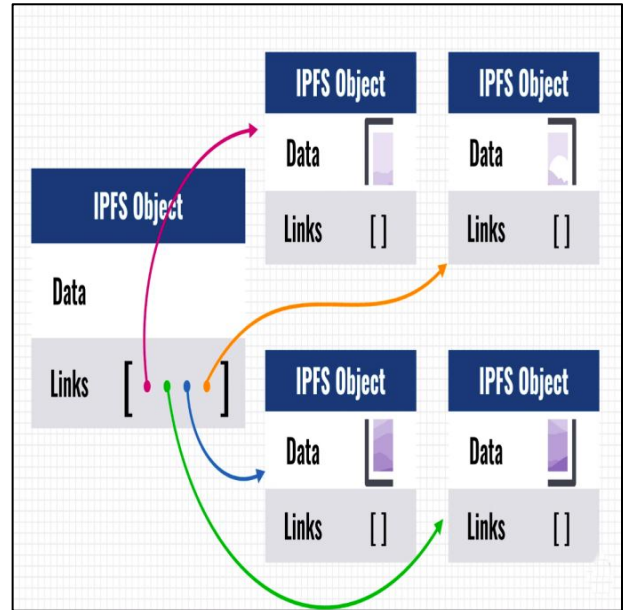


Fig5: Data storage in ipfs objects

VI. RESULT AND IMPLEMENTATION

Testing: We tried to upload and download files with various extensions into the IPFS server.

Results: We could download and upload files to the server.

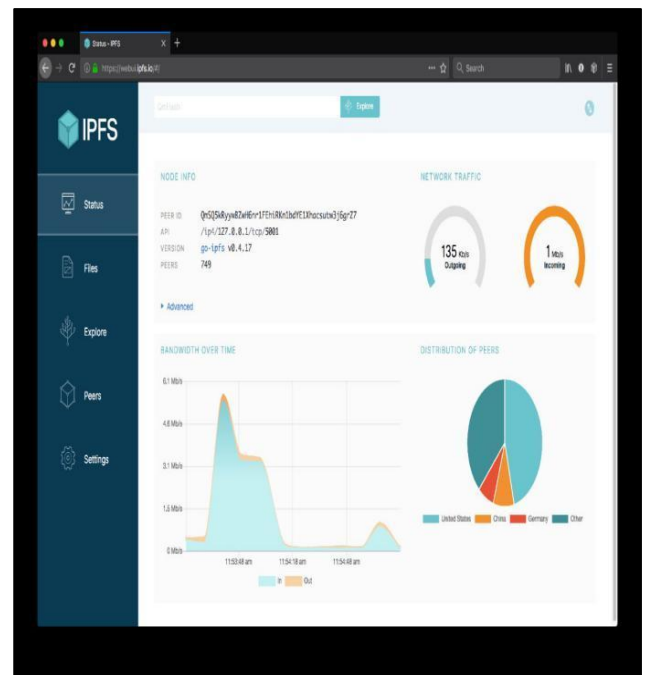


Fig6: Status of the IPFS

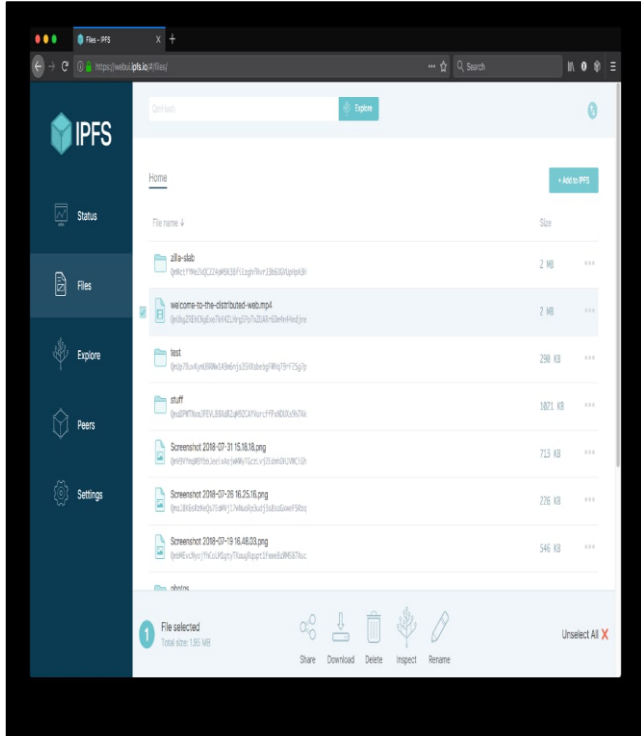


Fig:7 Files that are transferred through IPFS

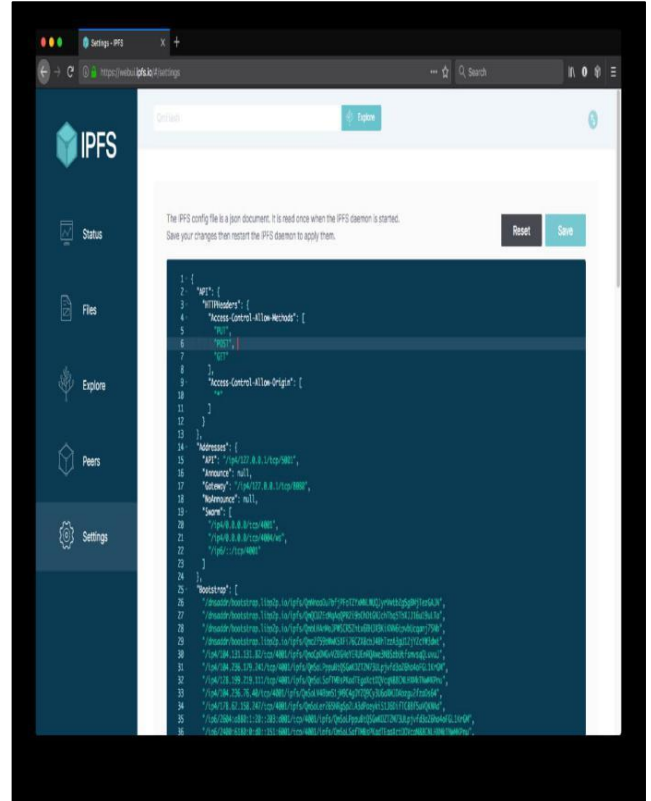


Fig9:Daemon connection established

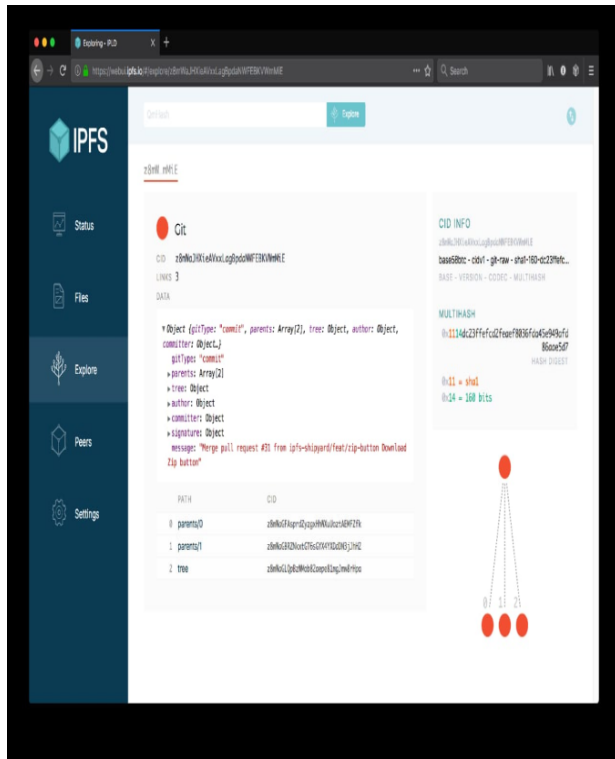


Fig8:File sent to the server is divided

After daemon connection is established, now it's been added in your node so when you started running that daemon you become a node. In these connections you become a peer on this network as the file travels through this network, it gets peered to of these different machines so even if you stop running your daemon it will still exist or if your daemon gets shut down. So if you go to the add files option in the IPFS web template which we created and add any file or mp3 or mp4 it will generate a unique hash key, copy and paste it in the explore box then data gets divided into chunks and gets spread across different peers and later copy and paste the hash in the browser URL so the file gets downloaded from IPFS server very fast compared to HTTP. The more it's played and downloaded the faster it gets because it has peered in more places[fig8].

VII.CONCLUSION

As technologies are growing faster and HTTP is a twenty years old technology so it needs replacement to keep up with the technology. IPFS is a new technology which provides faster internet speed, higher memory density and decentralization of virtual information. Moreover, IPFS provides increased security than HTTP. IPFS introduced us a technique self-certified file system. It has distributed data storage and it can be the standard data transfer protocol of the future with its ongoing development.

VIII. FUTURE WORK

The idea behind IPFS are the decade of successful distributed system research in academics and open source. IPFS symphonizes the most successful system till date. It uses Bit Swap strategy for data exchange between two peers and this is a novel protocol. It has a new decentralized internet infrastructure upon which many applications can be built, which makes it different. It can be used as a versioned file system and namespace, or as the next generation file sharing system.

IPFS could take the web to new perception, where user can trust the information what they receive without trusting the peer from where they are receiving the data, because it provides more security. And with IPFS no old file or any data do not miss. IPFS introduced us a technique self-certified file system, this allows us to generate an address for remote file system to verify the validity of the address. In future IPFS will be responsible to bring us permanent web.

ACKNOWLEDGEMENT

The authors would like to thank the REVA University School of C & IT for allowing access and use of the resources. The authors also appreciate the assistance given by Prof. Ajil A, our project mentor.

REFERENCES

- [1]. Baumgart and S. Mies. S/kademlia: A practicable approach towards secure key-based routing. In Parallel and Distributed Systems, 2007 International Conference on, volume 2, pages 18. IEEE, 2007.
- [2]. Bit Torrent. Bittorrent and A torrent software surpass 150 million user milestone, Jan. 2012.
- [3]. J. Dean and S. Ghemawat. Level dB—a fast and lightweight key/value database library by google, 2011.
- [4]. P. Maymounkov and D. Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In Peer-to-Peer Systems, pages 53–65. Springer, 2002.
- [5]. Idea about the IPFS protocol-<http://ipfs.io/>
- [6]. J. H. Howard, M. L. Kazar, S. G. Menees, D. A. Nichols, M. Satyanarayanan, R. N. Sidebotham, and M. J. West. Scale and performance in a distributed file system. ACM Transactions on Computer Systems (TOCS), 6(1):51–81, 1988.
- [7]. J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, et al. Ocean store: An architecture for global-scale persistent storage. ACM Sigplan Notices, 35(11):190–201, 2000.