IPFS: Decentralized storage in a centralized world

Ionut-Viorel Gingu, Blockchain Developer at RebelDot Solutions
email: ionut.gingu@rebeldot.com
Daniel Hursan, Lead Blockchain Developer at RebelDot Solutions
email: daniel.hursan@rebeldot.com

Abstract

The current technical brief starts with a quick introduction to traditional file storage solutions, and their shortcomings. It then introduces IPFS (InterPlanetary File System), a decentralized peer-to-peer storage network and protocol that combines Distributed Hash Tables with successful concepts from technologies like BitTorrent and Git. IPFS has become increasingly popular with blockchain developers, as they favor its decentralization to store costly data like Non-Fungible Tokens or frontend code.

Index Terms – IPFS, InterPlanetary File System, Blockchain, Centralization, Decentralization, Web3, Storage, Peer-to-peer, public ledger

Introduction

21st century's cloud is mostly large organizations, providing services for storing data and hosting applications. These platforms are not only in control of all operating business rules but can also act as a single point of failure (cloud providers have excellent availability, but outages still happen occasionally).

The blockchain may address these kinds of situations, promising:

- decentralization and no single point of failure (a public ledger that everyone has, where a node can drop out of the network without any significant impact).
- transparency and data immutability (an entry on the blockchain should be traceable, visible to everyone and impossible to tamper with).

Even if the above applies and modern solutions like Ethereum are used, developers can not solely rely on the blockchain. Storing bulkier files is not only expensive but also leads to the chain becoming so big that no one can participate anymore, except for organizations with significant resources.

This can already be seen in practice, where running a full node requires hundreds of gigabytes of free disk space [1]. Since this is not easily met by most people, developers are incentivized to store only the most essential information on-chain and take everything else off-chain, on traditional storage solutions.

The current work briefly expands on how these solutions work, their advantages and disadvantages, and then introduces the InterPlanetary Filesystem: how it works and solves some of the shortcomings, as well as current uses in the IT industry.

Traditional storage solutions

Since managing onsite data centers is difficult and expensive to bootstrap, it is usually impractical, and a procedure reserved for large and resourceful companies.

To overcome these challenges, most small organizations choose to use a cloud storage provider, such as Azure, Google Cloud or AWS. These platforms offer UIs or programmatic access to upload, download or manipulate your data, which is stored on their hardware. Their services range from all kinds of databases to generic storage solutions, both for personal use (pictures, backups, etc.) and enterprise.

Cloud solutions come with a couple of advantages and disadvantages:

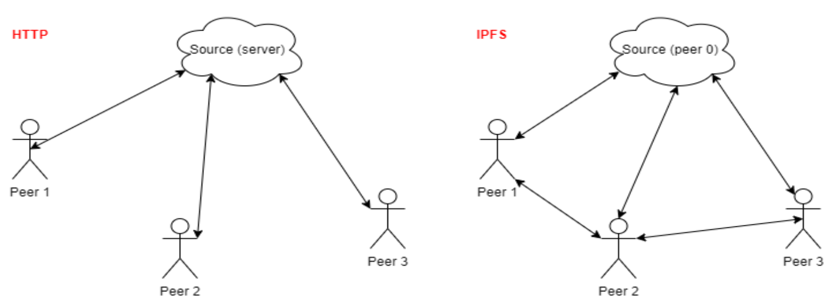
- Cheap, considering the alternative of setting up and maintaining everything onsite. Cloud platforms also provide advantageous payment plans for regular users, or free usage tiers.
- Easy to set up: a developer has an intuitive user interface to set up and scale infrastructure to their needs; a college student can just drag-and-drop their holiday photo album into the cloud.
- Increasingly stable by the year, as bugs get ironed out. Redundancy is increased by storing multiple copies of the data in different regions, so it is protected in case of hardware failures or natural disasters, while also reducing latency (signal needs to hop over less distance).
- Developers need to trust that platform owners have their best interest in mind, since they control the operational rules and pricing schemes.
- Heavily centralized, where a major outage takes down a considerable chunk of the internet, sometimes the very own tools a company would use to debug their software.

InterPlanetary File System (IPFS)

IPFS is a peer-to-peer distributed file system that seeks to connect all devices to the same pool of files. It aims to achieve “Lots of data, accessible everywhere” and functions very similarly to BitTorrent, where peers exchange file blocks with each other by following different incentivization strategies [2].

Decentralization

IPFS is a decentralized network, where every peer that downloads a file block, caches a copy of it for a while, then shares it with the others. The more a file is downloaded, the more copies there are, so the more available it becomes [3].



This results in:

- No single point of failure: a node disconnecting from the network should only come with a slight increase in latency unless it is the only node currently holding the requested file block.
- Censorship resistance: since a file is stored and shared by multiple peers, it is hard to fully block or remove it (think about how one would try to stop the whole internet: there is no OFF button). This helps with ensuring access to useful information that could have previously been unlawfully removed or hidden; on the other hand, it would be hard to deny access and limit the spread of hurtful and illegal content, like fake news or videos of bullying.

- Decreased latency: suppose watching an online video initially takes 10 seconds to load. A subsequent user should have it loaded faster, as they now benefit from the uploading bandwidth of previous peers as well. In the traditional system, a user having downloaded a file helps no other peers, as they will communicate with the same server setup as before.

Having to share a file with others also comes with a disadvantage: a node requires more bandwidth to operate, as participation in the network sometimes requires concurrent upload and download.

Content-based addressing

Most of the internet works using location-based addressing, where a file is available under a link chosen by the uploader. A user goes to that specific link and finds the piece of information.

IPFS takes a different approach, called content-based addressing, where a file's link is the hash of its contents, its fingerprint [4]. When in need of a file, a user asks for a specific fingerprint, and someone on the IPFS network can hopefully provide it for them.

One of the main advantages of this kind of addressing is that it provides link-content immutability: an uploader cannot move content to a different address, as the link is generated from the content itself; the content under a link cannot be updated, since it would result in a different hash.

Another advantage is the possibility of verifying that the requested file has not been tampered with. After receiving the file, its contents are hashed and compared with the requested fingerprint.

Lastly, content-based addressing is useful for deduplication: uploading the same file twice would result in only one copy being stored, as the second upload simply overwrites the first one.

Block storage mechanism

All file blocks are stored in one or multiple nodes' local storage. This can be disk space, if persistence after reconnection is required, or RAM memory. Data that is stored is treated like a cache: there is no guarantee it will continue to be stored indefinitely. For example, if a file has not been interacted with for some time, it will be overwritten by something else, which results in one less copy on the network.

To ensure survival of a file, a node can pin it, which does not allow overwriting the file in the node's local storage, ensuring at least one copy on the network.

Block exchange strategy

IPFS uses the BitSwap Protocol, where each peer advertises a want_list (blocks they need) and a have_list (blocks they have and can share). The protocol operates as a marketplace, where nodes barter and trade file blocks with each other. If a node has nothing that is currently wanted, it can look for pieces needed by its peers – this incentivizes exploration and dissemination of rare pieces of files.

In case a user has multiple blocks that others need, the protocol also tracks availability of each block and sends the rarest pieces first, thus prioritizing their spread through the network.

BitSwap also operates using a credit system, where each node keeps a ledger for each of the peers it has interacted with. This ledger records how much information has been exchanged between the two: if user A sent user B twice as many bytes as they have received, in the future B might be ignored when in need of file blocks, until the debt is repaid. This protects against leeches, nodes that never share information and use their whole bandwidth for downloading only.

Participation

The team behind IPFS describes it as a system based on possession and participation, where people have each other's files and take part in making them available. In contrast, today's World Wide Web is based on access and ownership, where one can get a file only if granted access by the owner.

IPFS's success is directly proportional to how many people get involved and run nodes, which is why it was designed to be quite easy to set up and use, even with the smallest available resources. Anyone can download the desktop application and start uploading and downloading files, at no cost.

IPFS use cases and adoption

IPFS can be used in a multitude of ways, ranging from a database to an encrypted CDN or a versioned package manager for software [5]. In 2021, the network had more than 230K unique weekly active nodes and 3.7M weekly ipfs.io Gateway users. It is currently leveraged by multiple companies, including Opera and Brave, a web browser designed for privacy and security [6].

As the blockchain on its own does not scale well as a storage solution, IPFS has also received a lot more attention and use by decentralized application developers in recent years: streaming music with Audius, blockchain-based DNS service with Unstoppable Domains, web wallet Metamask and others**.**

IPFS for storing decentralized applications' frontend

While a DApp's backend usually consists of smart contracts running on the blockchain, the frontend used to be on traditional storage solutions, centralized servers which did not match the blockchain paradigm very well. This has changed in recent years, increasingly more developers are choosing IPFS to store their HTML code. Both static single and multi-page websites are supported [7].

IPFS for storing NFTs

In 2021 alone, IPFS was used to store more than fifteen million Non-Fungible Tokens. One of the biggest adopters is OpenSea, currently the largest NFT marketplace on the decentralized web.

An NFT is a unique cryptographic token that exists on the blockchain. They are usually implemented as an on-chain immutable link, pointing to off-chain metadata or multimedia content. One of their use-cases is tracking ownership of digital assets: someone buys a music album, an NFT is minted, and the user has immutable proof that they own the album under that specific link. Since the off-chain content is mutable, the user might be confronted with their music album turning into a picture of a dog. IPFS's content-based addressing solves this, as it provides link-content immutability: not only cannot the link be changed since it is on the blockchain, but now the content under the link cannot be changed either, as it would hash to a different fingerprint and live at a different address [8].

Our hands-on interaction with IPFS allowed us to solve the issue of maintaining NFT data for a customer, by transitioning from a traditional approach, in which data was stored on a major cloud storage solution, to now, a decentralized, distributed approach, based on IPFS.

We were easily able to adapt our code from sending files to the cloud storage provider, to pointing to one of the leading providers of IPFS storage for NFTs – Pinata - https://www.pinata.cloud/.

Technically, the solution implementation consisted of a node.js application that acted as a proxy between REST API calls from the outside layer (web application and backend), and was routed to a controller, after each input parameter was validated beforehand.

We made sure that each file on IPFS had a meaningful name, and that the files met our business requirements limit of size (this is partly because in general, NFTs can be not only images, but also other copyright protected digital art, such as videos or music).

A different layer in our application invoked the SDK from Pinata, which allowed us to perform three important operations

- Transform and hash IPFS metadata
- Upload the actual files
- Pin the files so they are not archived, making them future proof

More and more providers are now joining the IPFS space, and you can find similar services from providers such as Eternum, Estuary or Temporal, each with their own pros and cons and different price plans.

A straightforward way of deploying data to IPFS is by following the official IPFS.io guidelines [9], which, in short terms require installing a command line tool, set up an account with one of the main deployers listed above, call a mint command, and pin your newly created NFT so it does not get archived.

Conclusion

Over the last years, IPFS has become increasingly successful, not only in the traditional software community, but also in the Web3 space. DApp developers leverage this peer-to-peer network since it lets them shift a lot of storage load, while also fitting the decentralization paradigm of the blockchain.

In the future, IPFS is expected to gain more popularity, given its ease of participation: even non-technical users can run a node and upload their files.

There are currently multiple projects in development, including storing petabytes of genetic data, building a distributed OS or a social media platform.

References

[1] “Ethereum Chain Full Sync Data Size”, ycharts.com, Jun. 3, 2022. [Online]. Available: https://ycharts.com/indicators/ethereum_chain_full_sync_data_size. [Accessed: Jun. 3, 2022]

[2] J. Benet, “IPFS - Content Addressed, Versioned, P2P File System”, 2014. [Online]. Available: https://github.com/ipfs/papers/raw/master/ipfs-cap2pfs/ipfs-p2p-file-system.pdf. [Accessed: Jun. 3, 2022]

[3] J. Schilling, J. Matthews, C. Waring, T. Chadbourne, A. Dunster, D. Daroch, “What is IPFS? “, docs.ipfs.io, para. 10, Jun. 23, 2021. [Online]. Available: https://docs.ipfs.io/concepts/what-is-ipfs/. [Accessed: Jun. 3, 2022]

[4] J. Schilling, J. Matthews, C. Waring, T. Chadbourne, A. Dunster, D. Daroch, "Content addressing ", docs.ipfs.io, para. 5, Jun. 23, 2021. [Online]. Available: https://docs.ipfs.io/concepts/what-is-ipfs/#content-addressing [Accessed: Jun. 3, 2022]
[5] B. Falguiere, J. Matthews, J. Schilling, I. Campos, D. Justice, H. Dias, C, Waring, M. Brandly, "Usage ideas and examples", docs.ipfs.io, Oct. 25, 2021. [Online]. Available: https://docs.ipfs.io/concepts/usage-ideas-examples/. [Accessed: Jun. 3, 2022]
[6] "IPFS in 2021: The Backbone of Web3's Mainstream Momentum", blog.ipfs.io, Jan. 11, 2022. [Online]. Available: https://blog.ipfs.io/2022-01-11-IPFS-in-2021/. [Accessed: Jun. 3, 2022]
[7] J. Matthews, R. Ore, T. Merkle, B. Chatelain, M. Mack, J. Schilling, "Host a single-page website on IPFS", docs.ipfs.io, Mar. 17, 2022. [Online]. Available: https://docs.ipfs.io/how-to/websites-on-ipfs/single-page-website/. [Accessed: Jun. 3, 2022]
[8] Y. Napora, J. Matthews, S. Rika, N. Sales, J. Schilling, "Mint an NFT with IPFS", docs.ipfs.io, Jul. 28, 2021. [Online]. Available: https://docs.ipfs.io/how-to/mint-nfts-with-ipfs/. [Accessed: Jun. 3, 2022]
[9] Y. Napora, J. Matthews, S. Rika, N. Sales, J. Schilling, "Mint an NFT with IPFS", docs.ipfs.io, "Minty", Jul. 28, 2021. [Online]. Available: https://docs.ipfs.io/how-to/mint-nfts-with-ipfs/#minty. [Accessed: Jun. 3, 2022]