This work is licensed under a Creative Commons “Attribution-NonCommercial-ShareAlike 4.0 International” license.

Decentralizing the Web: A Comparative Study of IPFS and Traditional Client-Server Models

Matthew D. Wanner
wanne051@morris.umn.edu
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA

Abstract

The internet, as commonly experienced by most users, operates on a traditional client-server infrastructure. This paper examines the performance of an alternative internet architecture based on a peer-to-peer system, with a primary focus on the InterPlanetary File System (IPFS). It explores IPFS's functionality, performance, applications, and the advantages and disadvantages of employing IPFS as a model for a decentralized Internet. Through a series of tests, this study assesses IPFS's efficiency in data storage and retrieval, its resilience against common internet challenges, and its potential as a foundational technology for a distributed web. The findings underscore IPFS's notable benefits in reducing reliance on centralized servers and bolstering resistance to censorship. Nonetheless, the paper also addresses challenges, including initial content retrieval times and the complexity of network management. The analysis offers valuable insights into the feasibility of adopting decentralized systems for future internet infrastructure, outlining broader implications for users, developers, and policymakers as they navigate the transition towards a more distributed web.

Keywords: InterPlanetary File System (IPSF), peer-to-peer (P2P), Decentralization, Latency, Data Storage and Retrieval, Content Addressable Storage, Network Gateway/Gateways, Non-Fungible Tokens (NFTs).

1 Introduction

The client-server model underpins the majority of today's internet interactions, serving as the backbone of digital communication and content distribution. While this centralized architecture has been instrumental in the development and scalability of the internet, it is not without its drawbacks. Issues such as privacy concerns, data security vulnerabilities, and the potential for censorship are inherent challenges of relying on centralized web services. These limitations have sparked interest in alternative models that promise to address these concerns. Among them, peer-to-peer (P2P) web architectures stand out for their potential to enhance user privacy, data integrity, and resistance to censorship. This paper delves into the exploration of such decentralized systems, with a particular focus on the InterPlanetary File System (IPFS). IPFS emerges as a leading example of how a decentralized web could operate, offering insights into a future internet infrastructure that is not only more resilient but also places greater power in the hands of its users while still being reliable and efficient.

In this analysis, I argue that IPFS, as a leading model of peer-to-peer web architecture, offers significant improvements over traditional client-server infrastructures in terms of enhancing user privacy, ensuring data integrity, and mitigating censorship. By looking at a series of performance tests and functional evaluations [4] [3], this paper aims to demonstrate IPFS's potential to revolutionize internet infrastructure, while also acknowledging the challenges and complexities inherent in transitioning to a decentralized web. Despite certain operational hurdles, adopting a decentralized approach presents a viable and advantageous path forward for the future of internet architecture.

2 IPFS Overview

Recent research [2, 4] provides insights into how the IPFS network functions. IPFS is a revolutionary protocol designed for decentralized storage and sharing of files across a distributed network of computers. Unlike traditional web hosting services that rely on centralized servers, IPFS operates on a peer-to-peer system, where each participant stores a piece of the overall data (see Figure 1 for a visual representation).

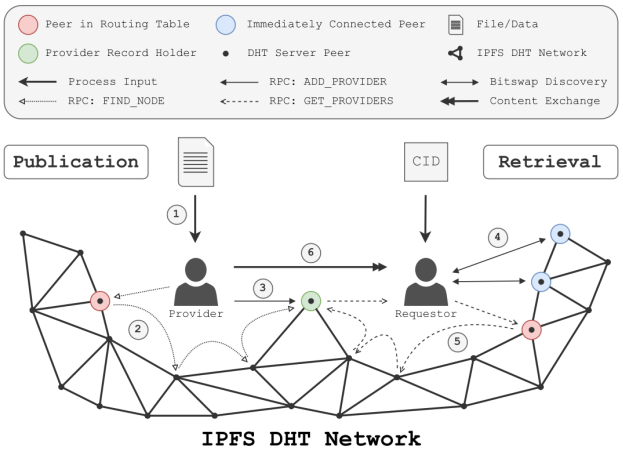

Figure 1. IPFS publication and retrieval (from [4])

At its core, IPFS uses content-addressable storage, meaning files are identified by their content rather than their location, like in traditional client-server systems. In IPFS, data is accessed through a unique hash generated from its content. When a file is added to IPFS, it is hashed, creating a unique identifier based on its contents. Retrieval requires requesting its hash; IPFS locates any node in the network storing that file. This approach not only ensures that data is tamper-proof and distributed, enhancing web efficiency and reliability, regardless of a user's geographical location, but also significantly increases resilience against censorship and data loss [4]. By leveraging a network of nodes, IPFS aims to create a more open web, where information is widely accessible and not controlled by any single entity.

2.1 Uploading to IPFS

When uploading a file to the InterPlanetary File System (IPFS), a multi-step process ensures the file is accessible across this decentralized network [4] (see Figure 1, publication side). Initially, the file undergoes a cryptographic hashing process to generate a unique Content Identifier (CID), a digital fingerprint that guarantees the content's integrity and uniqueness. This CID allows for efficient deduplication across the network and ensures that any piece of content can be verified against its identifier to confirm its authenticity.

Then, the user's node creates a provider record, a declaration within the network that it possesses the file linked to the specific CID. This critical step in content sharing involves updating the Distributed Hash Table (DHT), a decentralized system that maps CIDs to the nodes storing the corresponding content [2]. Rather than storing the content itself, the DHT stores these mappings, facilitating content location without central coordination. The provider record is strategically distributed to the 20 "closest" nodes (see Figure 1 step 1), where closeness is determined by the DHT's unique algorithmic distance metric that calculates proximity based on hash values rather than physical location. This method enhances the resilience and efficiency of content retrieval, ensuring that the system can locate provider records even as nodes join, leave, or change status within the network.

Content stored on the uploader's node remains there until it is actively retrieved by other nodes. These nodes may then cache or pin the content, with pinning indicating a commitment to store and provide the content long-term. This process not only aids in spreading content across the network but also in maintaining its availability.

2.2 Retrieving a File From IPFS

When a user seeks to access a file on IPFS, the process begins with the user's node looking up the file's CID (for visual representation look at the Retrieval side of Figure 1). To locate the file, the user's node queries the DHT. The DHT directs the query to the nodes that maintain the provider records for the desired file. (see Figure 1 step 5).

Upon receiving the query's results, the user's node obtains information about which nodes hold the file. It then initiates a connection to one of these nodes to request the file. This connection is facilitated by IPFS's network protocols, which navigate common hurdles such as Network Address Translation (NAT) barriers, ensuring nodes can communicate even in complex network environments. (see figure 1 step 6).

Once the connection is established, the querying node proceeds to download the file. After the download, the node verifies the file's integrity by recomputing its hash and comparing the result with the original CID. This step ensures that the content received is authentic and unaltered.

By successfully retrieving and verifying the file, the querying node not only accesses the desired content but also becomes a potential provider of the file. It can now serve the file to other nodes, enhancing the file's availability and redundancy across the IPFS network.

This retrieval process exemplifies IPFS's decentralized, peer-to-peer architecture, designed to enhance data availability, security, and integrity. By distributing the responsibility for storing and serving content across numerous nodes worldwide, IPFS creates a robust, resilient platform for information exchange [4].

2.3 Distributed Hash Table (DHT)

The DHT is collectively maintained across all nodes in the IPFS network, with each node holding only a small segment of the DHT (for visual representation look at the bottom portion of Figure 1). This segment helps the node guide requests either closer to the target content or directly to the node that stores it (see Figure 1 step 5). Continuous updates to the DHT are necessary as nodes join or leave the network and as content is added or removed ([2]).

3 Analysis

The authors of "Design and evaluation of IPFS", Trautwein et al. [4] employed three approaches to collect and evaluate data from IPFS. The first dataset they collected was information about peers acting as DHT Servers. To gather information about the number of users and the distribution of users around the world. The second dataset they collected was GET requests from a public IPFS gateway to understand how used gateways are on the IPFS network. The third and final dataset they collected was performance data about publication and retrieval of multiple IPFS nodes. To see how well individual nodes perform in different parts of the world.

3.1 Peer Data

To gather data about peers in the IPFS network acting as DHT servers, the authors used a crawler since there's no central list available. A crawler is a software tool to traverse and gather data about networks. This crawler, running from a server in Germany every 30 minutes, systematically asked

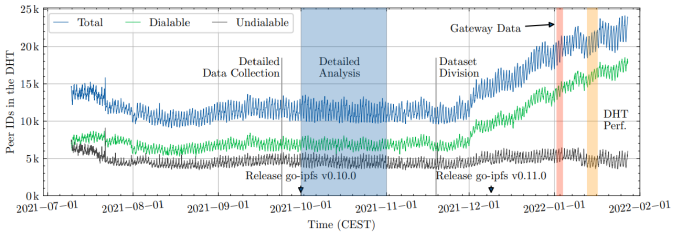

Figure 2. Total number of crawled peers over time and their fraction of dialable and undialable peers (one-day periodicity).

peers for their contact lists, starting from six known IPFS nodes and expanding outwards until it found no new peers. This method helped compile a detailed list of peers, including their locations and technical details, by running the crawler over 9,500 times.

3.2 Peer Data Results

The analysis of peer data revealed the presence of 198,964 peers spanning 152 countries. Notably, 54.5% of these peers were reachable at least once, while 45.5% remained inaccessible. The United States emerged as the country with the highest concentration of users, accounting for 28.5%, followed by China at 24.2%. France, Taiwan, and South Korea also featured prominently. This global distribution ensures that IPFS remains decentralized, safeguarding against dominance or disruption by any single country.

Merely 1.4% (2,747) of the peers exhibited an uptime exceeding 90%, thus deemed reliable by the authors. Conversely, approximately one-third of peers were never accessible. While seemingly concerning, this observation underscores the network's resilience. Despite a significant portion of peers being unreachable, the IPFS network continues to function effectively [4]. Refer to Figure 2 for a graphical representation of the crawler's findings. 'Dialable' denotes peers that were reachable, and 'undialable' indicates those that were not.

3.3 IPFS Gateway Data

For the second dataset the authors analyzed GET requests from a public IPFS gateway operated by Protocol Labs to understand how people use IPFS on a large scale. A gateway is a different way to interact with the IPFS network. Gateways provide a way to access the IPFS network without the need to run your own node by having users access it through an HTTP interface. This analysis focused on traffic from one day in January 2022 at a gateway located in the US, part of a network that distributes incoming traffic across several instances [4]. They examined 7.1 million requests, looking at details like when the request was made, what kind of device was used, where the request came from, how much data was sent back, and whether the data was already stored in the gateway's cache.

Table 1. Gateway Performance Data

	nginx cache	IPFS node store	Non-Cached
Latency (Median)	0 s	8 ms	4.04s
Traffic Served	46.4%	38.0%	15.6%
Requests Served	46.0%	40.2%	13.8%

Table 2. Number of Publication and Retrieval Operations

AWS Region	Publications	Retrievals
af_south_1	547	2,047
ap_southeast_2	547	2,630
eu_central_1	547	2,708
me_south_1	547	2,112
sa_east_1	546	2,363
us_west_1	547	2,704
Total	3,281	14,564

3.4 IPFS Gateway Data Results

Analysis of the data revealed that on the specified day, the authors identified 101,000 unique users accessing 274,000 distinct content identifiers (CIDs) . The average size of requests was 664.59 KB, with 79.1% of requests exceeding 100 KB in size. Notably, there was no discernible correlation between the size of objects and latency, suggesting that factors other than object size influence delay.

Approximately 46% of requests resulted in instantaneous retrieval, indicating a cache hit. This caching mechanism, inherent to gateways utilizing HTTP requests, significantly contributes to expediting response times. Moreover, the majority of remaining requests were serviced within 24ms. Interestingly, over half of the traffic (51.8%) originated from third-party sites, predominantly streaming platforms and NFT (Non-Fungible Token) platforms (see Table 1).

3.5 Performance Data

The third dataset focuses on testing how efficiently IPFS can publish and find content across different locations. The authors set up six virtual machines in various global regions using AWS, each running an IPFS instance to act as a DHT server node. These nodes were used to conduct experiments that measure how quickly a new piece of content (a 0.5 MB file) could be shared and then accessed across the network.

In each test, one node shared a new file, and the others tried to find and download it as illustrated in Table 2. This process tested the system's ability to distribute and locate content. To ensure accurate results, nodes disconnected after downloading the content to force the next test to start fresh, avoiding shortcuts through IPFS's content sharing mechanism, Bitswap, and relying instead on the DHT.

This setup aimed to mimic a real-world scenario within the controlled conditions of an experiment, acknowledging

	Publication Percentiles			Retrieval Percentiles		
AWS Region	50th	90th	95th	50th	90th	95th
af_south_1	28.93 s	107.14 s	127.22 s	3.75 s	4.88 s	5.31 s
ap_southeast_2	36.26 s	117.74 s	142.79 s	3.76 s	4.85 s	5.15 s
eu_central_1	27.70 s	106.91 s	133.27 s	1.81 s	2.28 s	2.50 s
me_south_1	29.32 s	105.45 s	130.48 s	2.59 s	3.24 s	3.48 s
sa_east_1	42.32 s	115.45 s	148.04 s	3.60 s	4.56 s	4.93 s
us_west_1	36.02 s	121.13 s	147.59 s	2.48 s	3.17 s	3.42 s

Figure 3. IPFS region publication and retrieval times.

the challenges of replicating the unpredictable behavior of network participants in a simulation. The paper notes the total counts of such publish and retrieve tests conducted from each location, mentioning that variations in these counts were due to the early termination of the experiment and minor coordination issues, which did not compromise the overall validity of the findings [4].

3.6 Performance Data Results

The median publication time across regions was recorded at 33.8 seconds, with the 90th and 95th percentile times standing at 112.3 and 138.1 seconds, respectively. These delays exhibit consistency across regions, as depicted in the Publication column of Figure 3. Notably, the primary contributor to publication delay is identified as the DHT walk, accounting for 87.9% of the total time [4]. In practical terms, this implies that out of a minute spent uploading a file to IPFS, approximately 87.9 seconds are attributed to the DHT walk. Enhancing the efficiency of the DHT walk emerges as a crucial area for future improvements.

On the other hand, retrieval performance achieved a 100% success rate [4], albeit with variability in retrieval times. On average, retrievals took longer than loading a typical web page but were faster than content publications on IPFS. The median retrieval speed was measured at 2.9 seconds, with the 90th and 95th percentile speeds at 4.34 and 4.74 seconds, respectively. Regional disparities were observed, with Central Europe boasting the fastest median retrieval time of 1.81 seconds, while South Africa exhibited the slowest median time at 3.75 seconds. The Retrieval column in Figure 3 illustrates the retrieval times across regions. The efficiency disparity between retrieval and publication times can be largely attributed to the nature of DHT walks. While publication DHT walks require locating 20 nodes to distribute the provider record, a retrieval walk concludes upon finding a single node.

4 Uses

4.1 Video on Demand

IPFS enhances peer-to-peer video and music streaming by leveraging its decentralized nature to distribute content across numerous nodes. This method significantly reduces bandwidth costs and enhances load times, particularly for popular content. Storage on multiple nodes allows users to access media from the nearest or most efficient sources, which minimizes latency and potentially increases download speeds. Furthermore, popular files are naturally replicated across more nodes, enhancing the network's capacity to handle large volumes of requests simultaneously without degrading performance. This decentralized approach to streaming not only improves user experience but also offers scalability and resilience in handling high demand for media content [4].

4.2 File sharing

IPFS significantly enhances file-sharing capabilities, particularly for large media files such as extensive datasets. By leveraging its decentralized network structure, IPFS distributes these files across numerous nodes globally. This distribution not only reduces reliance on centralized servers but also minimize bandwidth costs, a common challenge in traditional file-sharing systems [4]. Additionally, redundancy provided by IPFS enhances data durability, ensuring that files remain accessible even if some nodes hosting the data go offline. This redundancy helps preserve data integrity in a decentralized environment.

4.3 Social networking services

Because data on IPFS is distributed across multiple nodes, it is more difficult for governments or other entities to censor specific content or shut down the network. Users have more control over their data as it's not stored on centralized servers owned by a single company. This means they can decide who can access their information and under what conditions. For developers, utilizing IPFS reduces the need for costly server infrastructure, as data distribution and storage are handled by the network of nodes [4].

4.4 Non-Fungible Tokens

In "Dude, where's my NFT", Leonhard et. al [1] dive into how IPFS is used in the storage and distribution of Non-Fungible Token (NFTs). NFTs typically represent unique digital assets, including art, music, videos, and other forms of creative work. Traditionally, these tokens are stored on blockchains, another decentralized peer-to-peer network, also known for its robust security features. However, storing NFTs directly on blockchains presents several challenges, notably high costs and inefficiencies, prompting the adoption of alternative storage solutions [1].

IPFS offers a compelling solution by addressing the limitations associated with traditional blockchain storage:

- IPFS provides a **decentralized framework** for storing NFTs, mitigating the central points of failure associated with conventional cloud storage services.
- Unlike traditional URL-based addressing, IPFS uses **content-based addressing** to ensure data immutability and authenticity. Each CID facilitates verifiable authenticity, crucial for the integrity of NFTs.

- Files on IPFS are hosted by multiple nodes across the network, enhancing accessibility and reliability. This **redundant hosting** ensures that even if some nodes go offline, the data remains accessible, which is vital for the long-term preservation of digital assets.
- By leveraging IPFS for storage, the high costs associated with on-chain data storage are significantly reduced. IPFS allows for the **cost-efficient off-chain storage** of the actual digital assets, while the blockchain manages the ownership and transaction records using only the CIDs

While this paper does not focus on NFTs, the role of IPFS in NFT storage illustrates a practical application of how decentralized technologies can revolutionize digital content management and accessibility, transcending traditional boundaries and offering a scalable, cost-effective solution.

5 IPFS in Restricted Environments.

In "I'm InterPlanetary, Get Me Out of Here! Accessing IPFS From Restrictive Environments", Balduf et. al [3] describe their assessment of the functionality of the IPFS within restrictive network environments, particularly under the constraints imposed by China's Great Firewall (GFW). To do this, they set up a controlled experiment using four machines configured with different network settings. Two of these machines were non-NATed, serving as controls; one was located in Germany and the other in the United States. Non-NATed machines, which do not use Network Address Translation (NAT), provide a direct and unobstructed connection to the internet, thus serving as a baseline for optimal IPFS performance. In contrast, two NATed machines were established to simulate more restricted access environments; one was positioned in the United States and the other in China. The NATed machine in China was particularly critical for evaluating the impact of the GFW, an advanced censorship and surveillance system that blocks access to selected foreign websites and slows down cross-border internet traffic. By comparing the performance of these setups, the study aimed to explore both the data exchange capabilities between locally hosted IPFS nodes and the accessibility of IPFS gateways under varied network conditions. This experimental setup allowed the researchers to measure the differential impact of NAT and the GFW on IPFS, providing insights into how IPFS performs in environments where internet access is heavily regulated and restricted.

5.1 Gateway Testing in Restrictive Environments

In their investigation into IPFS accessibility, Balduf et. al [3] tested 81 public gateways, as listed by the IPFS community, to determine the extent to which the GFW affects IPFS gateway connectivity. Notably, all these gateways were hosted outside of China, requiring data requests from the Chinese node to pass through the GFW, thereby testing the firewall's impact on data transmission. The researchers conducted a systematic evaluation by attempting to retrieve a widely replicated text file from each of these gateways using nodes from different geographic locations. This approach was designed to assess the robustness and reliability of gateway access under varying network conditions. Additionally, they verified the integrity and authenticity of the retrieved files by checking their SHA256 hashes, a method that also helped determine if any gateways employed whitelisting or other forms of selective content delivery. This testing protocol not only provided insights into the accessibility of IPFS in restricted environments but also highlighted the potential variability in gateway performance and security measures across the network.

Table 3. Number of Working Gateways from each Machine

Machine	Tested	Working
Non-NATed German Client	81	14
Non-NATed US Client	81	13
NATed US Client	81	14
NATed China Client	81	5

5.2 Gateway Test Results

Balduf et. al [3] found that only 14 out of 81 tested gateways functioned correctly from non-NATed client machines located in Germany and the US. This observation raised concerns about the reliability of the publicly maintained gateway list, which may include outdated entries. Additionally, one of these 14 gateways exhibited inconsistent accessibility from a NATed node in the US, hinting at potential flakiness in its operation. The challenges were more pronounced from the node located in China, where only 5 out of 81 gateways were functional, underscoring the significant but not insurmountable barriers posed by the GFW. See Table 3 for a breakdown of gateway availability for each machine. These findings illustrate the variability in gateway performance and the impact of network restrictions on the accessibility of decentralized services like IPFS, highlighting the need for ongoing updates and maintenance of the gateway directories to ensure reliable access globally.

5.3 Client Node Testing in Restrictive Environments

In a subsequent phase of the study, Balduf et. al [3] utilized the same four vantage points, shifting their focus from public gateways to direct interactions using IPFS client software. To rigorously test the peer-to-peer functionality of IPFS, they generated random files at each node, ensuring that each node was the sole provider of its specific content. This setup allowed for a controlled examination of direct file transfers across the network. Over a period of seven days, each node was tasked with downloading content from another node in a pre-determined random sequence, effectively simulating

real-world data exchange scenarios and generating approximately 2000 unique data points per node. The integrity of each downloaded file was verified through SHA256 hash comparisons, confirming the authenticity and correctness of the data received. Notably, the researchers encountered some challenges while downloading and setting up the IPFS client software in China, primarily due to restrictions on accessing certain download sources. However, these hurdles were successfully overcome, demonstrating that while challenging, deploying IPFS in a restrictive environment like China is feasible. This phase of the testing underscored the robustness of IPFS's decentralized nature and its capacity to function effectively even when traditional download avenues are obstructed.

Table 4. Download Success Rate

Download Success Rate By Downloading Machine			
Stored On	*n*	Successful	Rate
Non-NATed German Client	2016	1160	0.58
Non-NATed US Client	2016	1323	0.66
NATed US Client	2016	1608	0.80
NATed China Client	2016	1621	0.80
Download Success Rate By Storing Machine			
Stored On	*n*	Successful	Rate
Non-NATed German Client	2016	1834	0.91
Non-NATed US Client	2016	1873	0.93
NATed US Client	2016	956	0.47
NATed China Client	2016	1049	0.52

5.4 Client Node Results

In a rigorous test of IPFS's functionality across diverse network settings, researchers conducted a total of 8,064 download attempts using four strategically positioned nodes, achieving an overall success rate of 71%. Analysis of the results revealed varying performance based on the network type: the German non-NATed client had a success rate of 58%, while the US non-NATed client showed slightly better performance at 66%. Remarkably, the NATed clients in both the US and China demonstrated higher success rates of 80% (see Table 4, top portion). This suggests that NATed environments do not necessarily impede IPFS's operational efficiency. On the uploading front, the non-NATed clients in Germany and the US achieved success rates exceeding 90%, indicating robust performance in unrestricted settings. Conversely, both the US and China NATed clients encountered more challenges, with success rates around 50%, highlighting some difficulties in more restricted network environments (see Table 4, lower portion). Despite these variances, the data confirms that IPFS client nodes maintain functional reliability even under restrictive conditions. The authors also noted that in practical applications, a node would typically source downloads from multiple other nodes, potentially increasing the likelihood of successful data retrieval [3]. This multi-node interaction inherent in IPFS's design enhances its resilience and efficacy, underlining its capability to operate effectively across a spectrum of global network conditions.

5.5 Nature of NATs

Uploading from NATed networks presented challenges, particularly when targeting non-NATed networks. This difficulty arises from the NAT's inherent function of masking IP addresses, which complicates the establishment of outbound connections. Interestingly, uploads between NATed networks were more successful, likely due to similar configurations which facilitate compatible connection protocols. Conversely, non-NATed networks displayed stronger performance in uploading due to their direct and unrestricted internet access, allowing them to initiate and maintain outbound connections more reliably. Downloading from NATed networks, however, proved more challenging for non-NATed networks, as the unpredictability of NAT configurations can hinder seamless data retrieval [3]. These findings highlight the intricate dynamics of IPFS's performance across varied network setups, emphasizing the need for adaptive strategies to optimize connectivity and data flow within decentralized frameworks.

5.6 Restricted Environments Conclusion

IPFS has demonstrated remarkable functionality, proving its efficacy even in restrictive environments and thereby overcoming significant barriers to information sharing. This robustness underscores IPFS's potential as a decentralized platform capable of maintaining operability despite network constraints or regulatory censorship. However, the system faces potential points of failure that could impact its performance and accessibility. One such vulnerability is the reliance on public gateways that could be limited or blocked in certain regions, affecting the ease of access to the IPFS network. Additionally, the distribution of IPFS software itself could be hindered by similar restrictions, posing challenges in environments where internet usage is heavily monitored or controlled. Despite these challenges, if these risks can be effectively mitigated—perhaps through more resilient network configurations and alternative software distribution methods—IPFS can be a powerful tool for the free exchange of information. Its ability to operate across restricted environments not only enhances its utility but also promotes a broader adoption of decentralized data solutions globally. [3]

6 Conclusion

While IPFS or other P2P network may never surpass the performance or replace the traditional client-server architecture, the various studies presented here show that IPFS is a reliable alternative way for storing, accessing, and sharing data around the world, making the internet a more open and user-empowered environment.

Acknowledgments

I would like to thank Professor Lamberty, my senior seminar advisor for her patience and guidance throughout this process. I would also like to thank Professor Guan for his work in leading the class each week. I would like to thank my alumni reviewer for their help. Finally, I would like to thank my friends and family for their continual support.

References

[1] Leonhard Balduf, Martin Florian, and Björn Scheuermann. 2022. Dude, where's my NFT: distributed infrastructures for digital art. In *Proceedings of the 3rd International Workshop on Distributed Infrastructure for the Common Good* (Quebec, Quebec City, Canada) *(DICG '22)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3565383.3566106

[2] Leonhard Balduf, Maciej Korczyński, Onur Ascigil, Navin V. Keizer, George Pavlou, Björn Scheuermann, and Michał Król. 2023. The Cloud Strikes Back: Investigating the Decentralization of IPFS. In *Proceedings of the 2023 ACM on Internet Measurement Conference* (<conf-loc>, <city>Montreal QC</city>, <country>Canada</country>, </conf-loc>) *(IMC '23)*. Association for Computing Machinery, New York, NY, USA, 391–405. https://doi.org/10.1145/3618257.3624797

[3] Leonhard Balduf, Sebastian Rust, and Björn Scheuermann. 2024. I'm InterPlanetary, Get Me Out of Here! Accessing IPFS From Restrictive Environments. In *Proceedings of the 4th International Workshop on Distributed Infrastructure for the Common Good* (<conf-loc>, <city>Bologna</city>, <country>Italy</country>, </conf-loc>) *(DICG '23)*. Association for Computing Machinery, New York, NY, USA, 13–18. https://doi.org/10.1145/3631310.3633487

[4] Dennis Trautwein, Aravindh Raman, Gareth Tyson, Ignacio Castro, Will Scott, Moritz Schubotz, Bela Gipp, and Yiannis Psaras. 2022. Design and evaluation of IPFS: a storage layer for the decentralized web. In *Proceedings of the ACM SIGCOMM 2022 Conference* (Amsterdam, Netherlands) *(SIGCOMM '22)*. Association for Computing Machinery, New York, NY, USA, 739–752. https://doi.org/10.1145/3544216.3544232