Cost and Performance Analysis on Decentralized File Systems for Blockchain-Based Applications: State-of-the-Art Report

Aisyah Ismail, Mark Toohey
Aglive Lab
P.O. Box 196
Geelong, Victoria, Australia
aisyah@aglive.com

Young Choon Lee
School of Computing
Macquarie University
Sydney, Australia

Zhongli Dong, Albert Y. Zomaya
School of Computer Science
The University of Sydney
Sydney, Australia
zhongli.dong@sydney.edu.au

Abstract—**Blockchain technology is an immutable append-only decentralized ledger that theoretically makes it an ideal data storage system. The nature of blockchain design ensures stored data is reliable, trustworthy, and transparent. However, the blockchain's full-replication feature makes it ineffective to store a large volume of data directly on-chain. This paper provides a systematic review and analysis of existing decentralized file systems.**

Our focus was on systems that can support large, high-frequency data writing while still providing swift and easy data retrieval for blockchain-based applications. The challenge, in our view, is to find ways to achieve those efficiency outcomes while still retaining the key decentralized features of blockchain design. With that in mind, we assessed the costs involved in using nine state-of-the-art decentralized file systems and we also considered their latency performance.

Index Terms—**Blockchain, Decentralized File Systems, Decentralized Storage Network, Decentralized Cloud Storage, InterPlanetary File System (IPFS), P2P System.**

1. Introduction

Most modern applications heavily rely on cloud storage services such as Amazon Web Service (AWS), Microsoft Azure, and Google Cloud Platform (GCP) for their near-unlimited storage capacity. While vast cloud storage capacity clearly has advantages, this storage method is still centralized.

Centralized systems have architectural constraints due to their single point of failure risk. Such designs are also vulnerable to Denial of Service (DoS) attacks, data breaches, and ex-filtration attacks [1]. Centralized systems are also prone to data misuse, political censorship, and potential disputes over data ownership. Depending on the legal framework set out in the terms of use, private data stored on such systems may not only be owned by the individual user, but also by the cloud storage provider [2]. To date, AWS dominates the worldwide cloud storage market with 33% of the market share, followed by Azure with 21%, and Google at 8% [3].

Because data can be replicated on numerous physical locations, decentralized systems can eliminate single point failure issues. Decentralized systems also offer greater security and privacy if uploaded files are sharded and encrypted before being shared. Although blockchains provide an immutable and reliable option for data storage, the full-replication and append-only nature of that design can render it an ineffective data storage solution. Storing data directly on the blockchain (commonly known as 'on-chain'), is computationally expensive and time-consuming. Existing research tends to focus on the use of off-chain data storage or decentralized file systems (DFS) with references only being kept on-chain [4]–[12]. Of course, this method helps retain the decentralized characteristics of a blockchain.

Daniel and Tschorsch's recent study compared six different DFSs with IPFS [13]. However, to the best of our knowledge, there is no work that comprehensively reviews the cost and performance of a DFS for blockchain-based applications (decentralized applications or dApps). In this paper, we systematically review and analyse nine state-of-the-art DFSs for dApps. We considered them in terms of their cost effectiveness (monetary cost) and latency performance. We acknowledge that the cost minimization is of great importance in the adoption of a DFS. The benefits delivered by blockchain and dApps, such as reliability, trustworthiness, and transparency must remain cost effective or adoption will be hindered.

Section 2 of this paper assesses: the currently available DFSs; the common characteristics of a DFS; and outlines the existing research on the use of a DFS for off-chain storage. Section 3 discusses DFS usage costs and DFS performance latency. Section 4 is a summary of our findings.

2. DFS Use

Data stored in a centralized system can be susceptible to a single point of failure if the data is stored in a single location such as a data centre. A DFS eliminates the single point of failure by replicating uploaded files on multiple nodes in the network. This takes advantage of excess storage capacity around the world. Because files are encrypted before being shared to other nodes, a DFS is more resistant to censorship and manipulation and any data lost in a breach will be rendered useless to an attacker. Consequently, a DFS is more robust and offers greater redundancy protection. For example, if a region has a power or network outage, users can still access their data because the data is replicated on multiple nodes and is not confined to a single location or region.

In Section 2.2 we will set out the common characteristics of the nine state-of-the-art DFSs that we studied. We will also address the existing literature on DFS use for off-chain storage in (Section 2.3).

2.1. DFS Overview

The nine DFS systems that we considered, including IPFS [14], are listed in Table 1. They are the most mature DFSs at the time of writing.

IPFS can easily be integrated into current production systems and they have been successfully deployed in several production projects [15]. When data is spread across a network of nodes, IPFS uses distributed hash tables (DHT) to enable access and lookup between nodes [16].

Protocol Labs, the creator of IPFS, created Filecoin [17] as a complementary protocol for IPFS. IPFS is open-source and free for download and use. Node participation is voluntary. However, this means there are no guaranteed nodes where an IPFS will store and keep specific data. Filecoin was created to solve this problem by incentivising its nodes to maintain a consistent level of redundancy and availability. Filecoin launched its “Filecoin” cryptocurrency token in August 2017 as its incentive methods [18]. At the time of writing, Filecoin mainnet has a storage capacity of 16.481 EiB (exbibyte).

TABLE 1. OVERVIEW OF AVAILABLE DFS.

Name	Incentive	Cryptocurrency	S3-Compatibility
IPFS	No	–	Yes, via 3rd party
Filecoin	Yes	Filecoin (FIL)	Yes, via 3rd party
Swarm	Yes	Ethereum (ETH)	No
Sia	Yes	Siacoin (SC)	Yes, via 3rd party
Storj DCS	Yes	STORJ	Yes
Arweave	Yes	Arweave (AR)	Yes, via 3rd party
Internxt	Yes	INXT	No
0Chain	Yes	ZCN	Yes, via 3rd party
Opacity	Yes	OPCT	Yes

Similarly to IPFS, Swarm is also an open-source globally shared storage provider that taps into unused storage among its nodes [19]. Swarm can be described as a development platform that operates natively in the Ethereum service layer (web3 stack). The Swarm incentive structure is based on the Swarm Accounting Protocol (SWAP), Registered nodes and Ensured ARchival (SWEAR), and Litigation on loss of content (SWINDLE) [13]. Swarm employs chequebook smart contracts to incentivise its network. This chequebook method uses the Ethereum cryptocurrency, Ether [20].

Sia [21], one of the oldest DFSs, was conceived in 2013. It competes with centralized cloud storage giants like AWS S3, Microsoft Azure, and Google Cloud Storage. Sia also owns and runs its own blockchain and uses it own cryptocurrency, Siacoin, to incentivise its contributing nodes [22]. According to SiaStat [23], the Sia network currently has a network capacity of 7.77 PB (petabyte), stores over 2.62 PB of data, and has over 730 active nodes distributed around the world.

Storj DCS, on the other hand, is powered by Storj [24]. Storj deals with the sourcing of decentralized storage while Storj DCS handles the network demand. Storj DCS uses Reed-Solomon erasure coding by breaking a file into 256 MB (megabyte) encrypted pieces [25]. Files are split into 80 pieces that each go to different nodes on the network, and any 30 parts can be used to retrieve the entire file. According to Storj, DCS has a network capacity of 15.16 PB with 14,400 vetted storage nodes in more than 90 countries [26]. Because Storj handles the procurement of storage, it uses the STORJ Token to incentivise its nodes.

Arweave’s [27] selling point is its permanent decentralized storage where user only pay once and data will be stored on Arweave network forever. At the time of writing, Arweave has a storage capacity of 74.09 TB (terabyte). Arweave has also introduced Permaweb, a permanent and decentralized web that is built on top of Arweave’s protocol.

Internxt’s [28] objective is to replace Dropbox and Google Drive by establishing a decentralized storage solution. Currently, Internext targets two different user groups: individuals and teams that lack API support for developers. Hence, the use of Internxt for dApps is currently not supported by Internxt.

0chain [29] is positioned as a different class of DFS with its dStorage platform. It implements n-dimensional architecture with multiple chains based on different forkable solutions. There are also three different types of miners for each chain. 0chain’s focus is on data storage for IoT applications that require high transaction volumes. 0chain uses its own ZCN token to reward miners.

Opacity [30] users are anonymous as no personal information is required, and the service can be purchased using its cryptocurrency OPCT. Similar to other DFSs, Opacity uses client-side encryption and shards each file into 5-10 MB chunks before distributing it across multiple storage locations. However, Opacity is essentially a centralized cloud storage system. It uses various cloud storage systems, instead of tapping into unused storage as is done by other DFSs [31].

2.2. DFS Characteristics

Based on Casino et al. [2], there are eight common DFS characteristics, that are: content immutability, equality, decentralization, fault tolerance and attack resilience, availability and censorship-resistance, unlimited resources, scalability, and marketplace monetization. Similarly, based on our observation, we concluded eight common DFS characteristics:

1. Content Immutability: The most common aspect of the DFS is their content immutability. Data stored in these systems can't be changed, or deleted, once it is uploaded. Content immutability for a DFS lies in the use of content addressing instead of the otherwise commonly used location addressing [16]. Content Addressing: Data or files are found in a network by using a cryptographic-hash matching. A minor edit will always result in a new address or a new hash of that file. Content addressing's advantage is that instead of using a location address it uses self-certification and checks file integrity. The hash proves authenticity and that dispenses with the need for third party verification of the content.

2. Permanency: Permanent data erasure across all hosting nodes is not guaranteed if some nodes are sharing the specific content. Some content could remain available to other network peers. DFSs such as Swarm, Sia, Arweave, and Storj DCS also inherit content immutability through the nature of a blockchain. Because all DFSs are decentralized, all peers or nodes in such a network usually have similar or equal permissions and possibilities. However, some DFSs enable users to ensure their data is given high priority, as in the case with IPFS and its pinning function.

3. Decentralization: Centralized storage systems like AWS, Microsoft Azure, and GCP use content distribution networks (CDNs) and data is replicated on multiple locations in different regions. This guarantees fault tolerance and high availability. However, such systems are still susceptible to DDoS attack due to their single point of failure architecture. DFS networks usually have a higher number of peers or nodes with no centralized entity compared to a centralized based storage system. Hence, a DFS is more fault tolerant and resilient to DDoS attacks. A DFS is more censorship-resistant compared to a centralized storage system as a DFS depends on multiple peers or nodes and does not rely on a single entity's storage redundancy capabilities.

4. Sharding: Most DFSs (e.g. IPFS, Swarm, Sia, Storj) use sharding techniques where uploaded files are broken into smaller chunks and encrypted [32]. The encrypted chunks are then propagated across the distributed network's nodes to ensure high availability. This sharding technique makes a network more robust. An attacker is likely to only obtain a file fragment of indecipherable data.

5. Scalability: Both centralized and decentralized storage systems send requests to the closest peer or server to avoid bandwidth bottlenecks. However, a DFS has a higher number of peers or nodes compared to a centralized based storage system. This enables a DFS to require less bandwidth for each node compared to a centralized server that has more users per node.

6. Monetization: Most DFSs also give monetary incentives for node storage and use cryptocurrency for convenience, privacy, and anonymity.

7. Usability: Despite the many advantages of a DFS, such systems are awkward to use. Currently, the task of setting up a cryptocurrency wallet is far too complex. Even tech-savvy users can be left bewildered and frustrated when they are forced to navigate through a maze of conceptual complexity and clunky design. Improved and simplified storage services are required. There is a clear need for systems that deliver all the benefits of a DFS, but spare users from the complexity. In other words, user-friendly 'minimal-knowledge' services are required. When deployed, such systems would enable consumers to reap the DFS benefits. Filebase [33], GooBox [34], Pinata [35], Infura [36], and RTrade's Temporal and TemporalX [37] are decentralized cloud storage examples. Filebase and GooBox have Sia as their back-end architecture while TemporalX and Pinata use IPFS.

8. Compatibility: S3-compatibility has become the gold standard in object storage services. The new breed of decentralized cloud storage services tend to offer S3-compatibility to ensure a smooth migration from a centralized storage system to a DFS. IPFS users can migrate from centralized cloud service by using RTrade's TemporalX [38]. Similarly, Sia users can use Filebase for S3-compatibility object storage. Arweave also recently announced integration with 4EVERLAND buckets for S3-compatibility [39]. Similarly, One Step Cloud announced an S3 gateway for 0Chain user [40]. Other DFSs like Storj DCS, and Opacity support object-based storage interfaces and are compatible with the AWS S3 interface natively as listed in Table 1.

2.3. DFS as Off-Chain Storage

Most current research focuses on use of a DFS as an off-chain solution in order to solve the computationally expensive problems of on-chain data storage. Our analysis shows that almost all systems use IPFS, and Ethereum with Solidity smart contracts as their off-chain storage architecture, as shown in Table 2. Two papers mentioned the use of a DFS private network [4], [8] for IPFS. Grabis et al. [11] used third-party IPFS API Infura. We noted a trend towards the use of a private network via virtual blockchain, Ganache, or a test network like Rinkeby or Ropsten, for Ethereum blockchain implementation. In contrast, Zhou et al. [10] used Hyperledger blockchain and Hyperledger Composer Playground for smart contracts.

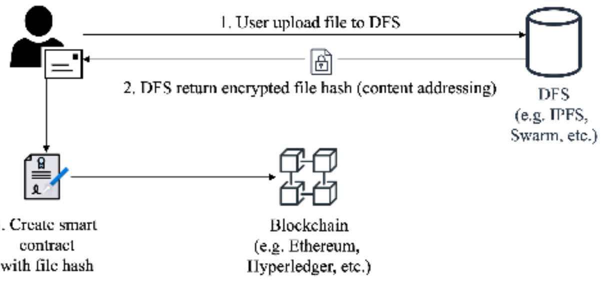

Figure 1. Common off-chain storage architecture.

TABLE 2. CONCEPT MATRIX OF THE REVIEWED LITERATURE RELATED TO OFF-CHAIN STORAGE

	DFS				Blockchain						Smart Contract		Domain			
Author(s)	IPFS	Swarm	Private Network	IPFS API	Ethereum	Private Network	Ganache	Rinkeby Network	Ropsten Network	Hyperledger	Solidity	Hyperledger	Data Sharing	Store Data	Proof-of-Concept	Measurement
Eisenring [4]	✓		✓		✓			✓			✓			✓		✓
Norvill et al. [5]	✓				✓						✓			✓		✓
Özyilmaz et al. [6]		✓			✓	✓					✓			✓		
Bhosale et al. [7]	✓				✓		✓				✓				✓	
Ramesh [8]	✓	✓	✓		✓	✓					✓			✓		✓
Naz et al. [9]	✓				✓	✓					✓		✓			✓
Zhou et al. [10]	✓									✓		✓		✓		
Grabis et al. [11]	✓			✓	✓				✓		✓			✓		✓
Javed et al. [12]	✓				✓		✓				✓			✓		✓
Nevpurkar et al. [41]	✓				✓								✓			
Kostamis et al. [42]	✓	✓			✓				✓		✓			✓		

There is a tendency to use off-chain storage to reduce the on-chain data volumes. Most of the implementation or use cases in the selected literature focused on stored data or data sharing applications. Out of the eleven papers that were reviewed, only six papers included metrics on the proposed solution [4], [5], [8], [9], [11], [12]. Section 3.1 will discuss the findings of those six measurements. We referred to those eleven papers, in our attempt to depict the similarity between all off-chain storage implementation architecture as illustrated in Fig. 1.

3. DFS Cost-Effectiveness

Despite their commonalities, decentralized storage solutions comes in many different shapes and sizes. They often have unique priorities and target markets. This section narrows it down to two aspects of cost-effectiveness: DFS usage costs (Section 3.1); and performance latency (Section 3.2).

3.1. DFS Usage Costs

Centralized storage system pricing depends on a multitude of factors and the eventual cost is generally based on usage in the past month. The most common factor in centralized storage is the storage class.

Storage can be divided into three data access classes: Hot - highly accessed data; Cool - less frequently accessed data; and Cold - archived data.

On average, the cost to store 1 TB of data for one month in a centralized storage system is around $25, not including charges for data egress and API requests. In contrast, most DFS are cheaper than centralized storage systems, as shown in Table 3.

IPFS use is free, but there is no guarantee that the content will always be available - unless the owner keeps serving it from their own nodes or in their own private IPFS network. As mentioned in Section 2.1, Filecoin is created to incentivise nodes in IPFS as a motivation for storing content. Filecoin pricing depends on fluctuations in the Filecoin storage marketplace. According to Filecoin's website, the price to store 1TB of data for a year is less than $1 [43]. Apart from Filecoin, another option is to use a third-party IPFS API or file manager like Pinata, RTrade's Temporal, or TemporalX.

Pinata has different pricing tiers. Free starter accounts have storage capped at 1 GB (gigabyte). Heavier users can pay $20 per month for 50 GB storage, $100 per month for 250 GB storage, or $1000 per month for 2500 GB storage. On the other hand, RTrade has two IPFS API versions, Temporal and TemporalX. Temporal has 7070 a free tier with storage capped at 3 GB per month. For unlimited storage, Temporal costs $0.07 per GB or $0.05 per GB with a different IPFS key, IPNS records, and PubSub messages offering. Meanwhile, pricing for TemporalX starts from $299.

For Ethereum's Swarm storage, the cost depends on the Ethereum transaction cost and Swarm's own storage cost. Apart from Swarm's incentives using SWAP, SWEAR, and SWINDLE, the development partnership between Swarm and Rootstock Infrastructure Framework Open Standard (RIFOS). RIFOS is currently working on a storage application called RIF Storage [44]. The collaboration aims to implement Swarm incentives in the future.

The Sia Network uses its own Siacoin (SC), and the storage costs depends on the prices set in the Sia storage marketplace. Pricing is also based on Siacoin's market value

[45]. At the time of writing (June 2022), the price to store 1 TB per month cost 319.38 SC. This converts to around USD$1.23 per TB per month. On top of the storage price, users must also pay for upload and download bandwidth. That cost also depends on Sia's storage marketplace. For the past six months, the price for both upload and download bandwidth averaged below $1.50. Moreover, users also need to pay a one-time contract formation fee which cost less than $0.50 and 5-15% of the allowance formed for network fees.

Similar to the Sia Network, pricing for Arweave also depends on the Arweave marketplace or AR cryptocurrency pricing. Arweave via its ArDrive provides an easy to use calculator to estimate price of storing data on Arweave [46]. At the time of writing, the price to store 1 TB of data on Arweave or ArDrive is 334.87 AR or $3134.46. This is a one time payment compared to the monthly price incurred on other DFSs.

Storj DCS aims to be cheaper than centralized cloud storage its prices set at 80% less than big cloud prices. Listed pricing has two components: the free plan and the standard plan. Standard plans cost $4 per TB per month with a bandwidth cost of $7 per TB per month [47].

Internxt aims to replace Dropbox and Google Drive. Pricing for Internxt is similar to a centralized cloud with two categories: individual and business. For the individual, they offer a free account with a maximum of 10 GB storage. For a paid individual account, the pricing starts at 0.99 € per month for 20 GB storage, followed by 4.49 € and 9.99 € per month for 200 GB and 2 TB storage respectively. Business plan storage caters for collaboration, and pricing starts at 4.99 € per user per month for 200 GB. 2 TB storage cost 9.99 € per user per month while 20 TB storage priced at 95 € per user per month. Although Internxt has its token called INXT, it is only used to pay their hosting nodes [48].

0Chain position themselves as a zero-cost, super-fast, decentralized cloud for dApps to perform micro-transactions and store data using their n-dimensional architecture with multiple chains. However, they do not charge to use their architecture. Anyone who wants to use 0Chain needs to lock in a certain amount of ZCN token as a deposit. During contract creation, the locked token amount is determined based on the storage required in GB, read or write, and the period of tenure. At the end of the tenure, user will get back the locked amount of tokens [49]. As with any cryptocurrency, the price is volatile, and any savings for users may vary due to fluctuations.

Opacity, on the other hand, accepts the OPCT token and US Dollar payments.($). Interestingly, the price is fixed for both currency and free usage for 10 GB of storage with 2 GB file size limit. A basic account is priced at 2 OPCT per year or $19.99 per year for 128 GB of storage with 2 GB file size limit. For both professional and business accounts, there are no file size limits and professional user gets 1 TB of storage at 16 OPCT per year or $79.99 per year. Business accounts cost 32 OPCT per year or $99.99 per year with 2 TB of storage.

While most DFS have their cryptocurrency, only Sia, Arweave, and Opacity use them as the payment method. All other DFS uses fiat currency as their payment method with the US dollar ($) as their choice of currency - except for Internxt who use the Euro (€). Their tokens or cryptocurrency are also used to pay miners and hosting nodes as incentives. Of course, there can be a downside for cryptocurrency payments if the users find themselves on the wrong side of volatile pricing.

As mentioned above, the average price to store 1 TB on a centralized storage system is around $25 per month, not including charges for data egress and API requests. Based on the cost of using the nine DFSs, five out of nine DFSs have much lower prices compared to a centralized storage system 3. The five DFSs are Filecoin, Sia, Storj DCS, Internxt, and Opacity with an average of around $5 per TB per month compared to $25 for a centralized storage system.

TABLE 3. COST COMPARISON BETWEEN CENTRALIZED STORAGE SYSTEM AND DFS (AS OF JUNE 2022)

DFS	Cost for 1TB (Monthly)
Centralized	$25
Filecoin	<$0.1 * (<$1/year)
Pinata	$400
RTrade's Temporal	$50
Sia	<$1.25
Arweave	<$3500 §
Storj DCS	$4
Internxt	<€5
Opacity	<$7 * (<$80/year)

* Price shown is divided by 12 as users will be charged on a yearly basis
§ A one-time payment

Although an IPFS is free, third-party use of IPFS APIs or a file manager such as Pinata or RTrade's Temporal can be more expensive than the centralized storage system cost of $25. Similarly, 0Chain is free, but the user needs to put a ZCN token down as a deposit. At the time of writing, 0Chain is still in beta stage, and no pricing is currently available.

Arweave imposed a one-time payment at a higher price of more than $3000 per TB. Although Arweave is comparatively more expensive, users only need to pay once. There are no recurring payments, like there is in other DFSs or any centralized storage system.

In addition to the DFS costs, each transaction on an Ethereum blockchain incurs charges that are calculated using the Ethereum Gas unit. Each transaction that executes a smart contract has a fixed base cost. Any extra costs depend on the complexity of the smart contract code and the volume of data stored in the blockchain. Smart contract complexity directly impacts on amount of Gas used and the Gas price, which is calculated in Gwei, is market based and it fluctuates daily.

The Ethereum yellow paper states every transaction will require at least 21,000 gas [50]. Various gas price options can be selected to alter a transaction's processing priority. The actual cost depends on which gas price option the user chooses, and that decision directly affects transaction speeds. Naz et al. [9] deduced that the cost for contract creation on Ethereum is 0.00361647 ETH or $0.55 with 1 ETH equals to $150 at the time. While uploading a file to a server costs 0.000107896 ETH or $0.016. The price to delete a file from the system is 0.000058412 ETH or $0.0088. A similar experiment by Bhosale et al. [7] and Javed et al. [12] extrapolated that gas consumption for added data functions is higher compared to other functions. Other than that, Grabis et al. [11] concluded on-chain storage uses more storage and is more expensive than off-chain storage.

3.2. Latency Performance

Apart from the cost of subscribing to DFS or a cloud storage platform, latency is also a significant factor when choosing a storage provider. Low latency is the goal of any network-related system. The lower the latency, the faster the connection and faster connections reduce the time taken to download data from the cloud. Centralized storage systems claim high upload speeds, but the actual reading times are much slower, as in the case of AWS's S3. Recently, a user in AWS re:Post forum complained that it took up to 1 hour to upload around 30 MB files [51].

Performance can be affected by the geographical locations of the user and the data server. Theoretically, DFS data transfers are much faster because data is downloaded or uploaded asynchronously to multiple nodes in parallel, instead of to only one server in the traditional centralized cloud system.

Moreover, all major centralized cloud systems are also susceptible to human error and downtime. AWS experienced significant downtime in 2017. This was reportedly caused by human error [52]. The event took down many large internet sites for a few hours. Since then, AWS has changed its operational practices to reportedly ensure it will never happen again. Similarly, in 2019, both Microsoft Azure and Google Cloud Platform experienced a major global outage. Microsoft Azure's outage in May 2019 lasted more than one hour due to a DNS migration issue [53]. Google's Cloud Platform had a significant disruption in November 2019, when numerous services failed [54].

There are no official IPFS latency numbers as performance depends on the hosting node's bandwidth and also on the lookup operation via a public DHT. Huang et al. [55] reviewed previous research on IPFS and concluded that IPFS is still at an immature stage as it has high I/O latency and bottlenecks when reading and downloading remote objects. Henningsen et al. [56] came to a similar conclusion about IPFS with its high latency, low throughput, and high redundancy data retrieval. Henningsen et al. also mapped the IPFS overlay networks and found 59.19% of 44,474 nodes reside behind NAT. This indicated that most IPFS nodes are hosted by private individuals.

Table 4 presents the DFS performance results for eight papers selected in the literature review.

IPFS is an open-source project. Anyone can download and use the technology on their own private network. This is the case for many IPFS API providers (i.e. Pinata, Infura, TemporalX, etc.). Consequently, Confais et al. [57] reported low latency and high throughput when running IPFS in a private local network compared to IPFS on public networks. Confais et al. also reported it takes longer time to write or upload files than to read or retrieve a file from IPFS [57]. Similar experiments done by Javed et al. [12] also showed how the add data function consumed more gas compared to the functions delete and share file processes. On the other hand, Norvill et al. [5] experimented with different file sizes ranging from 66 bytes to 19,014 bytes for IPFS retrieval on Ethereum and found that the average retrieval time across all file sizes is 62 milliseconds. The work by Eisenring [4] measured how the transfer time for a 173 kilobytes payload varied between 0.2 to 2.6 seconds.

TABLE 4. LITERATURE REVIEW WITH MEASUREMENT OF DFS PERFORMANCE

Author(s)	DFS	File Size	Result
Confais et al. [57]	IPFS (Private)	256KB, 1MB, and 10MB	Low latency and high throughput in private IPFS. The time to write is higher than to read from IPFS
Eisenring [4]	IPFS (Private)	173kB	Transfer time varied between 0.2 to 2.6 seconds. The transfer time increases with the increase of distance of nodes
Norvill et al. [5]	IPFS	66-19,014 bytes	Reduces the size of data stored on blockchain by 93.86%. Average time for IPFS retrieval across all file size is 62 ms
Ramesh [8]	IPFS & Swarm (Private)	5,000 transactions of 10,000 IoT records	Swarm perform better than IPFS
Naz et al. [9]	IPFS	–	Actual cost for contract creation is $0.55 while upload a file to server operation costs $0.016
Grabis et al. [11]	IPFS API (Infura)	1,024-4,720 bytes	On-chain storage use more storage hence more expensive than off-chain storage.
Javed et al. [12]	IPFS	–	PoA uses less gas than PoW. Function add data consumed more gas compared to other functions (delete and share).
Kostamis et al. [42]	IPFS & Swarm (Rospten)	4KB–16MB	Algorithm and data structure used by IPFS and Swarm leads to performance gap.

Ramesh [8], compared the use of IPFS and Swarm private networks as off-chain storage for Ethereum. He found Swarm performs significantly better when compared to an IPFS. The experiments showed Swarm not only read and wrote faster than an IPFS, but it also consumed less CPU and RAM compared to an IPFS. The time taken to write data on an IPFS takes four times longer than on Swarm. While the time taken to read data does not differ much between the two DFSs. Work by Kostamis et al. [42] showed comparable results where Swarm performed faster than an IPFS when uploading data up to 256 KB. However, Swarm performance deteriorates significantly when uploading more than 256 KB. The performance gap between IPFS and Swarm lies in the data structure differences between the two systems. The IPFS default chunk size is 256 KB, while a Swarm chunk is 4 KB.

In March 2020, Storj officially announced the production launch of an enterprise-grade Tardigrade. It later became known as Storj DCS [58]. At launch, Storj referred to the thorough readiness testing done during their beta period. During that beta testing they achieved 99.995% availability, 99.9999999% file durability, 20% faster upload and download performance compared to AWS, 6 PB proven capacity, 5,000 active nodes, and 2% vetted node churn [59]. Storj DCS announced confidence in their rigorous testing and third-party security firm vetting. However, we were unable to find any official third-party report. Discussions between Storj DCS users in the Storj forum on 23 July 2020 showed Storj DCS didn't perform to expectations [60].

In a Sia blog post via Medium in July 2019, Sia promised 300 Mbps for upload and download of data [61]. However, the SpaceDuck.io blog discovered an average upload speed on Sia for 4.3 TiB (tebibyte) of data over 231.7 hours is 45.8 Mbps [62]. SpaceDuck showed bandwidth was reliable for the first 2.5 TiB with speeds around 75 to 100 Mbps. It should be noted that the experiment was halted before the full 4.3 TiB of data was uploaded because the bandwidth dropped below 2 Mbps. Based on the experiment log, Sia stops uploading data because it keeps duplicating existing data to achieve its three-time duplication standard. The SpaceDuck experiment was published in March 2018, which is more than a year before the Sia blog post. We were unable to find other Sia experiments at the time of writing.

4. Conclusion

Despite the many DFS advantages, adoption of these systems may take time as most current DFSs are in early development and still require improvements. This paper considered the background overview of nine state-of-the-art DFSs and their common characteristics compared to the traditional centralized cloud storage systems. Then, we discussed the use of DFS as off-chain storage solutions before we considered usage costs. The latency performance of the currently available DFSs was also considered.

We believe IPFS is the most stable DFS and it has a strong community and well written documentation. IPFS also has a good track record of successfully employed projects. Despite the latency issues in an IPFS, developers can opt to use a paid third-party IPFS API or host their own private network.

References

[1] M. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras, , and H. Janicke, "Blockchain technologies for the internet of things: Research issues and challenges," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2188–2204, 2019.

[2] F. Casino, E. Politou, E. Alepis, and C. Patsakis, "Immutability and decentralized storage: An analysis of emerging threats," *IEEE Access*, vol. 8, pp. 4737–4744, 2020.

[3] B. Murray and Y. Zhang, "Global cloud services spend hits us$55.9 billion in q1 2022," *Canalys*, 2022, [Online]. Available: https://www.canalys.com/newsroom/global-cloud-services-Q1-2022.

[4] L. Eisenring, "Performance analysis of blockchain off-chain data storage tools," *University of Zurich*, 2018.

[5] R. Norvill, B. Pontiveros, R. State, and A. Cullen, "Ipfs for reduction of chain size in ethereum," *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2018.

[6] K. Özyılmaz and A. Yurdakul, "Designing a blockchain-based iot with ethereum, swarm, and lora: The software solution to create high availability with minimal security risks," *IEEE Consumer Electronics Magazine*, vol. 8, no. 2, pp. 28–34, 2019.

[7] K. Bhosale, K. Akbarabbas, J. Deepak, and A. Sankhe, "Blockchain based secure data storage," *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, no. 3, pp. 5058–5061, 2019, [Online]. Available: https://www.irjet.net/archives/V6/i3/IRJET-V6I31286.pdf.

[8] V. Ramesh, "Storing iot data securely in ethereum blockchain," *University of Nevada, Las Vegas*, 2019.

[9] M. Naz, F. Al-zahrani, R. Khalid, N. Javaid, A. Qamar, M. Afzal, and M. Shafiq, "A secure data sharing platform using blockchain and interplanetary file system," *Sustainability*, vol. 11, 2019, [Online]. Available: https://www.mdpi.com/2071-1050/11/24/7054.

[10] I. Zhou, I. Makhdoom, M. Abolhasan, J. Lipman, and N. Shariati, "A blockchain-based file-sharing system for academic paper review," *2019 13th International Conference on Signal Processing and Communication Systems (ICSPCS)*, 2019.

[11] J. Grabis, V. Stankovski, and R. Zariņš, "Blockchain enabled distributed storage and sharing of personal data assets," *2020 IEEE 36th International Conference on Data Engineering Workshops (ICDEW)*, 2020.

[12] M. Javed, M. Rehman, N. Javaid, A. Aldegheishem, N. Alrajeh, and M. Tahir, "Blockchain-based secure data storage for distributed vehicular networks," *Applied Sciences*, vol. 10, 2020, [Online]. Available: https://www.mdpi.com/2076-3417/10/6/2011.

[13] E. Daniel and F. Tschorsch, "Ipfs and friends: A qualitative comparison of next generation peer-to-peer data networks," *ArXiv*, vol. abs/2102.12737, 2021.

[14] "Ipfs powers the distributed web," *IPFS*, [Online]. Available: https://ipfs.io/.

[15] "Ipfs awesome," *GitHub*, [Online]. Available: https://github.com/ipfs/awesome-ipfs.

[16] J. Benet, "Ipfs - content addressed, versioned, p2p file system," *IPFS Whitepaper*, 2014, [Online]. Available: https://arxiv.org/abs/1407.3561.

[17] "A decentralized storage network for humanity's most important information," *Filecoin*, [Online]. Available: https://filecoin.io.

[18] J. Benet and N. Greco, "Filecoin: A decentralized storage network," *Filecoin Whitepaper*, 2017, [Online]. Available: https://filecoin.io/filecoin.pdf.

[19] "Storage and communication for a sovereign digital society," *Swarm*, [Online]. Available: https://ethersphere.github.io/swarm-home/.

[20] V. Trón, A. Fischer, D. Nagy, Z. Felföldi, and N. Johnson, “swap, swear and swindle: incentive system for swarm,” *Ethersphere Orange Papers 1*, 2016, [Online]. Available: https://swarm-gateways.net/bzz:/swarm.eth/ethersphere/orange-papers/1/sw%5E3.pdf.

[21] *Sia*, [Online]. Available: https://sia.tech.

[22] D. Vorick and L. Champine, “Sia: Simple decentralized storage,” *Sia Whitepaper*, 2014, [Online]. Available: https://whitepaper.io/document/17/siacoin-whitepaper.

[23] “Charts & metrics for the sia network,” *siastats.info*, [Online]. Available: https://siastats.info.

[24] “Storj - decentralized cloud storage,” *Storj*, [Online]. Available: https://www.storj.io/.

[25] “Storj: A decentralized cloud storage network framework,” *Storj Whitepaper*, 2018, [Online]. Available: https://storj.io/storjv3.pdf.

[26] J. Gleeson and V. Ihnatiuk, “Sharing space for fun and profit—part 2,” *Storj*, 2021, [Online] Available: https://www.storj.io/blog/sharing-space-for-fun-and-profit-part-2.

[27] “arweave - store data, permanently.” *arweave*, [Online]. Available: https://www.arweave.org/.

[28] “Be limitless,” *Internxt*, [Online]. Available: https://internxt.com.

[29] “Data privacy, protection & private sharing platform,” *0Chain*, [Online]. Available: https://0chain.net.

[30] “Cloud storage and file sharing, powered by crypto,” *Opacity*, [Online]. Available: https://www.opacity.io.

[31] “Opacity (opq) – blockchain storage solution,” *Altcoin Buzz*, 2019, [Online. Available: https://www.altcoinbuzz.io/reviews/crypto-education/opacity-opq-blockchain-storage-solution/.

[32] S. Wilkinson, “Storj: A peer-to-peer cloud storage network,” *Storj Whitepaper*, 2014, [Online]. Available: https://storj.io/whitepaper/.

[33] “Blockchain-based object storage,” *Filebase*, [Online]. Available: https://filebase.com.

[34] “Goobox,” *StoreWise*, [Online]. Available: https://github.com/GooBox.

[35] “Add files to ipfs effortlessly,” *Pinata*, [Online]. Available: https://pinata.cloud.

[36] “Ethereum api — ipfs api gateway — eth nodes as a service,” *Infura*, [Online]. Available: https://infura.io.

[37] “Temporal.cloud,” *Temporal*, [Online]. Available: https://temporal.cloud.

[38] “Usage ideas & examples,” *IPFS Docs*, [Online]. Available: https://docs.ipfs.io/concepts/usage-ideas-examples/#aws-s3-integration.

[39] “Integration announcement – 4everland bucket x arweave,” *Arweave*, 2022, [Online]. Available: https://arweave.news/integration-announcement-4everland-bucket-x-arweave/.

[40] “One step cloud implements tiering and s3 gateway into a blockchain-based storage platform,” *One Step Cloud*, [Online]. Available: https://www.onestepcloud.com/news/20200214.html.

[41] M. Nevpurkar, C. Bandgar, R. Deshmukh, T. Thombre, R. Sadafule, and S. Bhat, “Decentralized file storing and sharing system using blockchain and ipfs,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 7, no. 5, pp. 560–563, 2020, [Online]. Available: https://www.irjet.net/archives/V7/i5/IRJET-V7I5112.pdf.

[42] P. Kostamis, A. Sendrós, and P. Efraimidis, “Exploring ethereum’s data stores: A cost and performance comparison,” *ArXiv*, vol. abs/2105.10520, 2021.

[43] “Filecoin for large datasets,” *Filecoin*, [Online]. Available: https://largedata.filecoin.io/.

[44] “Rif storage: The first chucks,” *Rif*, [Online]. Available: https://www.rifos.org/blog/rif-storage-the-first-chunks.

[45] “Storage pricing,” *SiaStats.info*, [Online]. Available: https://siastats.info/storage_pricing.

[46] “Pricing - ardrive,” *ArDrive*, [Online]. Available: https://ardrive.io/pricing/.

[47] “Pricing — affordable object storage — storj dcs,” *Storj DCS*, [Online]. Available: https://www.storj.io/pricing.

[48] “Our tokenized asset,” *Internxt Token*, [Online]. Available: https://internxt.com/inxt.

[49] J. Rietz and S. Basu, “0chain economic protocol,” *0Chain Whitepaper*, 2018, [Online]. Available: https://0chain.net/page-whitepapers.html.

[50] G. Wood, “Ethereum: a secure decentralized generalised transaction ledger,” *Ethereum Yellow Paper*, [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf.

[51] Vincent, “Aws slow upload on tiny files,” *AWS Developer Forum*, 2022, [Online]. Available: https://repost.aws/questions/QU_6JJXJxGQj6pgaaRbRSaQw/aws-slow-upload-on-tiny-files.

[52] “Summary of the amazon s3 service disruption in the northern virginia (us-east-1) region,” *AWS*, [Online]. Available: https://aws.amazon.com/message/41926/.

[53] L. Tung, “Azure global outage: Our dns update mangled domain records, says microsoft,” *ZDNet*, 2019, [Online]. Available: https://www.zdnet.com/article/azure-global-outage-our-dns-update-mangled-domain-records-says-microsoft/.

[54] “Google cloud datastore incident #19006,” *Google Cloud*, [Online]. Available: https://status.cloud.google.com/incident/cloud-datastore/19006.

[55] H. Huang, J. Lin, B. Zheng, Z. Zheng, and J. Bian, “When blockchain meets distributed file systems: An overview, challenges, and open issues,” *IEEE Access*, vol. 8, 2020, [Online] Available: https://ieeexplore.ieee.org/document/9031420.

[56] S. Henningsen, M. Florian, S. Rust, and B. Scheuermann, “Mapping the interplanetary filesystem,” *2020 IFIP Networking Conference (Networking)*, 2020, [Online]. Available: https://arxiv.org/abs/2002.07747.

[57] B. Confais, A. Lebre, and B. Parrein, “Performance analysis of object store systems in a fog/edge computing infrastructures,” in *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2016, pp. 294–301.

[58] B. Golub, “General availability for tardigrade is here,” *Storj Blog*, 2020, [Online]. Available: https://storj.io/blog/2020/03/general-availability-for-tardigrade-is-here/.

[59] ——, “Measuring production-readiness using qualification gates,” *Storj Blog*, 2019, [Online]. Available: https://storj.io/blog/2019/11/measuring-production-readiness-using-qualification-gates/.

[60] GrolaG, “Tardigrade test results from may to august 2020,” *Storj Forum*, [Online]. Available: https://forum.storj.io/t/tardigrade-test-results-from-may-to-august-2020/8273.

[61] Z. Herbet, “Meet sia: the most viable non-financial application of blockchain technology,” *Medium*, 2019, [Online]. Available: https://blog.sia.tech/meet-sia-the-most-viable-non-financial-application-of-blockchain-technology-afe6e7412a25.

[62] “Sia load test result 2: Real data scenario,” *SpaceDuck*, [Online]. Available: https://blog.spaceduck.io/load-test-2/.